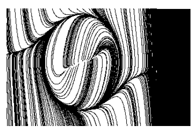
Reasonable Basic Calculus

Alain Schremmer

Reasonable Basic Calculus

According To . . . The Real World,
From . . . Mere Signed Decimal Numbers.

In other words, for people who believe calculus ought to make sense.



FreeMathTexts.org
Free as in free beer. Free as in free speech.

Version 1.7 — Monday 10th October, 2022

To Françoise.

Mathematician and pianist of my life!

Contents

What is important is the **real world**, that is physics, but it can be explained only in mathematical terms.

Dennis Serre[1]

real world

Preface You Don't *Need* To Read

Standard prefaces are never *for* you *but to convince* teachers *that the text is just what* they *want their students to* buy *for that class* they *are to* teach.
In contrast, this preface you don't need *to read* is *for* you.

Of course, every textbook is different from every other textbook—at least so claim their authors. And of course, so claims *this* author! But exactly *how* is this text different? First, though, how about the standard texts?

Why else would they ever have wanted to write it?

For Whom The Standard Texts Toll

Back in 1988, Underwood Dudley (https://en.wikipedia.org/wiki/Underwood_Dudley) published in the American Mathematical Monthly a wonderful article about calculus books—camouflaged as a Book Review![2]—which he said he wrote after having "*examined 85 separate and distinct calculus books*". (https://www.maa.org/sites/default/files/0002989051112.di991736.99p03667.pdf)

Dudley's first point was that "*Calculus books should be written for students*". As an example of one such, Dudley gives Elias Loomis' (https://en.wikipedia.org/wiki/Elias_Loomis) *Elements of the Differential and Integral Calculus* from 1851.[3] He points out that Loomis' "*proof of L'Hospital's*

[1] Bulletin of the American Mathematical Society, Vol 47 Number 1 Pages 139-144

[2] Here is *all* of the review: '*The book by Simmons is a fine one. It was written with care and intelligence. It has good problems, and the historical material is almost a course in the history of mathematics. It is nicely printed, well bound, and expensive. Future historians of mathematics will look back on it and say,* 'Yes, that is an excellent example of a late twentieth-century calculus book.'').

[3] Free from https://archive.org/details/elementsofdiffer00loom/page/n4/

That's the spirit!

*E.g. G. Strang in his Calculus (p151): "*I regard the discussion below as optional in a calculus course *(but required in a calculus book)."*

Which goes to show that universality can have a steep price.

Rule was short, simple, and clear, and also one which does not appear in modern texts because it fails for certain pathological examples." A bit later, Dudley continues: "*It is a still better idea to strive for clarity and let students see what is really going on, which is what Loomis did, rather than putting 'rigor' first. But nowadays, authors cannot do that. They must protect against some colleague snootily writing to the publisher "Evidently Professor Blank is unaware that his so-called proof of L'Hospital Rule is faulty, as the following well-known example shows. I could not possibly adopt a text with such a serious error.*"'

At less than $10!

Which, these days, would be an unspeakable horror!

As another example of a book written for students, Dudley gives Silvanus Thompson's (https://en.wikipedia.org/wiki/Silvanus_P._Thompson) *Calculus Made Easy*[2] from 1910 which was very successful and is in fact still in print. Dudley is visibly enchanted to report that "*Chapter 1, whose title is* 'To Deliver You From The Preliminary Terrors*' forthrightly says that* dx *means 'a little bit of* x". (Significantly enough, Thompson was a professor of physics and an electrical engineer.)

Another point Dudley made was that "*First-semester calculus has no application.*" Of course there is no question about CALCULUS being about the Real World. Absolutely none. The only thing is, the Real World is in the eye of the beholder and the beholder is, here again, the teacher. And so, of course, Dudley riffes on "*Applications being so phony*".

Well, this *one sure wasn't!*

Dudley concluded that "*It is a shame, and probably inevitable that calculus books are written for calculus teachers.*"

And, indeed, as he predicted, nothing has changed to this day. For instance, and even though it is about "*school math*", see the American Mathematical Society's 2015 Response (http://www.ams.org/notices/201505/rnoti-p508.pdf) to Elizabeth Green's New York Times article *Why Do Americans Stink at Math?* (https://www.nytimes.com/2014/07/27/magazine/why-do-americans-stink-at-math.html). In that response, Hung-Hsi Wu (https://math.berkeley.edu/~wu/) wrote "*If Americans do "stink" at math, clearly it is because they find the math in school to be unlearnable.* [...] *For the past four decades or so the mathematics contained in standard textbooks has played havoc with the teaching and learning of school mathematics.*" Why should it be any different with CALCULUS?

mode/2up

[2]Free from https://archive.org/details/CalculusMadeEasy/page/n4/mode/2up.

For Whom *This* Text?

Da Vinci
language
paper world
word
meaning
entity

The short answer is that, while standard texts usually do a good job of showing how CALCULUS proceeds from the *mathematian*'s viewpoint, *this* text wants to do it from *your* point of view and since, as **Leonardo da Vinci** (1452-1619) once put it, "*Learning is the only thing the mind never exhausts, never fears, and never regrets.* (https://www.azquotes.com/author/15101-Leonardo_da_Vinci), *this* text was written for people who want to read, ponder, wonder, and ... *learn*.

In short, for people allergic to "just teach them how to do it".

EXAMPLE 0.1. In *this* text, CALCULUS starts with Functions Given By Data (Part I, Page 63) with Functions Given By Rules appearing only in Part II, Page 207 and (Laurent) Polynomial Functions in Part III, Page 337 .

On the other hand, should you prefer to skip all that and go see for yourself, if only for now, just click on Chapter 0 - Numbers For Calculating (Page 1) or Chapter 1 - The Name Of The Game (Page 63).

Unfortunately, if this short answer may look good, it doesn't really say very much and a rather long answer follows for those who, before deciding whether or not to get into something, want to know *precisely* what it is they would be getting into and *why* they would want to do that in the first place.

Before that, though, and just in case you missed the subtitle of the book: as long as you can compare, add/subtract, multiply/divide *signed decimal* numbers, you don't have to worry about being "prepared".

In any case, if and when you want, there will always be Appendix A - Dealing With Numbers (Page 501).

Calculus Language Vs Everyday Language

The long answer starts with the fact that, to communicate about the real world, we need to use a **language** which is something that belongs to what's sometimes called the **paper world** and, at least from a theoretical standpoint, one should indeed distinguish **words** in the paper world from their **meaning**, that is from the **entities** in the real world that the words refer to. (See https://en.wikipedia.org/wiki/Language, https://en.wikipedia.org/wiki/Semantics, https://en.wikipedia.org/wiki/Entity)

Veeeeery long answer!

EXAMPLE 0.2. The word tree in the English *language*—as weill as the words arbre in French, Baum in German, árbol in Spanish, albero in Italian, etc, refers to the *real world* entity whose picture is

model theory
precise
sentence
situation
define

This distinction between paper world and real world is at the core of a relatively new part of MATHEMATICS called **MODEL THEORY**. See https://en.wikipedia.org/wiki/Model_theory but the article does not mention the recent applications of MODEL THEORY to other branches of (advanced) mathematics. Readers of this text might want to look at the author's https://www.researchgate.net/publication/346528673_A_Model_Theoretic_Introduction_To_Mathematics_4th_edition.

Veeery advanced.

However, while being aware of the distinction between paper world and real world is fundamental, using two sets of words, one for the real world and another for the paper world would not serve any purpose in this text. So:

AGREEMENT 0.1 In *this* text, we will *not* distinguish words from their meaning.

EXAMPLE 0.3. We shall use the *word* tree to refer to the *entity* tree.

On the other hand, just like "*Law depends on the* **precise** *meaning of words*", as Chief Inspector Kan reminds Inspector Van der Valk in Nicolas Freeling's (https://en.wikipedia.org/wiki/Nicolas_Freeling) thriller, *Criminal Conversation*[3], so do **S**cience, **T**echnology, **E**ngineering and . . . **M**athematics.

In STEM, the only suitable response to "You know what I mean" is a flat "No!"

So the first way this text claims to be different is the extreme attention paid to words. (https://plainlanguagenetwork.org/plain-language/what-is-plain-language/) Because, like in court, to be able to agree on what **sentences** are saying about real world **situations**, we need to use words that have been **defined** precisely. Only then will we have a chance, ultimately, to *deal* intelligently with the real world.

[3]A legal term! (https://en.wikipedia.org/wiki/Criminal_conversation)

Concerning the relevance of MATHEMATICS to the real world, here are two articles very much to the point:

- A very famous, if somewhat dense, article on "*The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics*, https://www.maths.ed.ac.uk/~v1ranick/papers/wigner.pdf by Eugene Wigner (https://en.wikipedia.org/wiki/Eugene_Wigner),

which eventually started a lively discussion on *natural law* and *mathematics*:

- https://www.quantamagazine.org/puzzle-solution-natural-law-and-elegant-math-20200117/

situation
calculate
compute
change
calculus word
everyday word
formal
meaningless

Notice their use of "real-world situation"*, practically a mantra in* this *text.*

1. Calculus language. Contrary to what most people think, CALCULUS is, *before anything else*, a language. But CALCULUS is a language that is *systematic* and thus lends itself to **calculating**—aka **computing**—how the real world **changes**. (https://en.wikipedia.org/wiki/Calculation)

i. In order to communicate *precisely* we will need **Calculus words**. So, to help you get a precise idea of what a calculus word means, each and every calculus word will be introduced using: **(i) everyday words** together with *already defined* calculus words, and **(ii)** an **EXAMPLE** to *illustrate* what the calculus word refers to in the real world.

Most of the time, that will be enough for you to keep on trucking safely but, occasionally, a **formal**, that is a dictionary-like, definition of a calculus word in terms of only previously defined calculus words will be necessary and will then appear in a special format:

EXAMPLE 0.4. just to show the special format:

DEFINITION 0.1 Meaningless means the same as without meaning.

ii. However, a major obstacle to learning the language of CALCULUS is that many calculus words are just everyday words to which a very precise CALCULUS meaning has been assigned. The danger then is for the reader later to forget they are facing a calculus word and go by the everyday word. Which, unfortunately, is exactly when CALCULUS will stop making sense.

And therefore to understanding *CALCULUS,*

iii. And, to make things even worse, we will have to use these calculus words alongside everyday words because it is of course with everyday words that we will describe and discuss *what* we will be doing with the calculus words and explain *why* we are doing what we are doing.

But why pseudo philosophers brandish mathematics words *like* space, catastrophe, field, category, ...

symbol
iff
□

iv. However, we will *not* use words that *mathematicians* often use but never really define. (https://en.wikipedia.org/wiki/List_of_mathematical_jargon)

v. And, because it is extremely easy to overlook for which previous noun in a sentence a *pro*noun stands, this text tries never to use *pro*nouns even though it means repeating the noun itself.

Pace English teachers!

vi. Symbols are necessary to carry out computations.

EXAMPLE 0.5. Figuring in *everyday language* the difference between three thousand seventy nine Dollars and eight Cents and six hundred forty seven Dollars and twenty six Cents would be a lot harder than *computing* the difference in the BASE TEN *language*:

$$\begin{array}{r} \$3\,079.08 \\ -\$647.26 \\ \hline \end{array}$$

(https://en.wikipedia.org/wiki/Hindu%E2%80%93Arabic_numeral_system)

Not all symbols, though, are for computational purposes and a few are just like abbreviations. For instance, we will use the following two symbols which are completely standard, if relatively recent inventions, but with which you may not be acquainted:

Not to be confused with https://en.wikipedia.org/wiki/Identification_friend_or_foe

LANGUAGE NOTE 0.1 iff , read "if and only if", is the symbol that indicates of two sentences that neither one can be 'true' without the other one also being 'true'. (https://en.wikipedia.org/wiki/If_and_only_if)

But why is it that "Jack sits *to the right of Jill iff Jill* sits *to the left of Jack" is* false*?*

EXAMPLE 0.6. The sentence "Jack is to the right of Jill iff Jill is to the left of Jack" is true.

LANGUAGE NOTE 0.2 □ , read "Q.E.D.", is the symbol that indicates the end of a proof. (https://www.urbandictionary.com/define.php?term=QED)

2. Click to recall. Because you will have to concentrate on *what's going on*, the language has to be *transparent*. But. as pointed out above, it is not easy to keep in mind precisely what calculus words refer to. So:

i. Like any scientific book, this text will help you retrieve what calculus words and symbols precisely refer to by having every single one of these calculus words and symbols in the INDEX at the end of the book along with the page where the calculus word or symbol is defined—and appears in bold black characters in the text as well as in red characters in the margin of that page.

Onscreen, a click on the page number will get you there.

ii. Using the INDEX more than occasionally, though, even onscreen, is a huge pain which makes it extremely likely you will put off looking up what the *calculus* word refers to precisely and rely instead on the *everyday* word, ...and then be left facing text that makes no sense.

And so, calculus word will always appear in red-black characters to remind you that *clicking* on that calculus word will instantly get you back to the page where what the *calculus* word refers to was precisely explained. In fact, more generally,

But you may have to scroll a bit if you want to see the word in the margin. And what to click on to return to where you were will depend on your pdf reader.

> **AGREEMENT 0.2** Anything, anywhere, that appears in red-black characters is a click away from what that thing refers to:
> - *Titles* in all tables of contents,
> - *Page numbers* in all references,
> - *References* as in DEFINITION 0.1 or **??** or as in the *Blue Note* just to the right.

To take a break from this Preface You Don't Need *To Read, would you like to go see the actual beginning of* CALCULUS *in Chapter 1 - The Name Of The Game (Page 63)?*

3. Reading calculus. One thing has to be emphasized, though, which is that, no matter how much attention has been given to language difficulties, it is impossible to get from a *single* reading of a piece of text everything that's in that piece of text. This is because it is impossible for *any* piece of text to say it *all* so that any piece of text will have to rely on some things having been said *earlier* to prepare the ground and there will be some things that can only be said *later*, when everything has been made ready to make the point.

So, here are a couple of maneuvers used by *mathematicians* when they are reading a text in a subject they are not familiar with and, like you will too, run into something they don't get:

Like the CALCULUS *in this book needs Chapter 0 - Numbers For Calculating (Page 1).*

- If, even after you have made sure you know whar every single calculus word in the piece of text denotes you are having trouble with, you still don't really get the message or something still does not connect, then try going back to a place in the text with which you have made your peace and reread it anyway. You will probably discover things you had not thought of when reading it before. Now read forward till you reach that

Back & Forth *maneuver!*

decide
true
false
factual evidence
state
believe

place where you stalled and it may very well be that those new things you hadn't thought of before will now help you make it through.

Wait & See *maneuver!*

- If you do get what a piece of text is saying but just don't see what the point is, make a note of it and then try reading ahead anyway and eventually you might have the "Aha", that is you may now understand the point of that piece of text you had trouble with.

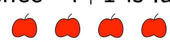
Like you might finally see the point of *Chapter 1 - The Name Of The Game (Page 63)*.

Proof Vs Belief.

Another way this text claims to be different has to do with the question: how do we **decide** if a sentence is **true** or **false**, that is whether what the sentence says about the real world is really the case? Or not? Or whether the sentence is undecidable?

A much debated issue—at least by some *people.*

With some sentences, we can decide on the basis of **factual evidence**, that is by checking what the sentence says directly against the real world. Unfortunately, most sentences cannot be checked against the real world.

EXAMPLE 0.7. We can decide that the sentence "4+1 is larger than 4" is true by trying to match  against .
But what collection should we look at in the real world to decide if the sentence "4 000 000 000 000 000 000 000 000 000 000+1 is larger than 4 000 000 000 000 000 000 000 000 000 000" is true?
And, even more to the point, what should we look at in the real world to decide if the sentence "*Any* number plus one is larger than the number itself" is true? Of course, we can check for any number(s) *we want* but that does not prove that the sentence "$x + 1$ is larger than x" will be true for *any* number we replace x with.

And, contrary to what many people seem to believe these days, just **stating** a sentence, no matter how many times and how forcefully, does *not* make the sentence true. And just invoking some other text doesn't work either: how do we know the author of that other text hadn't had some hidden agenda? Or didn't really know what they were writing about? Or made some honest mistake? So, in *everyday life* the matter often comes down to being a matter of **belief**.

Well, I don't know ... If you were to point a gun *at me ...*

On the other hand, in *mathematics*, we cannot just believe whatever we want.

EXAMPLE 0.8. What would happen, even in *everyday life*, if, say, the result of an addition was up to the beliefs of whoever does the addition?

theorem
theorem
postulate
axiom
deductive rule

We need to let others know on what *basis* we believe a sentence to be true and the standard way to do that in mathematics is to create a **theory**, that is a collection of sentences called **theorems**, by proceeding as follows:

i. **Postulate** (https://www.thefreedictionary.com/postulate) *a few* sentences *believed* to be true—to be referred to as **axioms** (https://en.wikipedia.org/wiki/Axiom)—to be theorems,

ii. Use **deductive rules** (https://en.wikipedia.org/wiki/Natural_deduction) to get new theorems from old theorems.

Then, because the deductive rules *preserve* truth, the use of deductive rules reduces the question of truth from that of a huge number of sentences to that of just a few axioms because the truth of the theorems will then derive from the truth of the theorems which were used with the deductive rules and so, ultimatley, from the truth of just the axioms. (https://en.wikipedia.org/wiki/G%C3%B6del%27s_completeness_theorem)

On what *basis* do we choose axioms is a totally separate issue. It could be on the basis of direct reference to the real world, or because the axioms are *conjectured* to be true on the basis of indirect or incomplete evidence (https://en.wikipedia.org/wiki/Conjecture) or maybe just out of curiosity, just to see what would ensue if we were to postulate *these* axioms instead of *those* axioms.

This [illegible] truth, called model-theoretic *or* semantic *(https://en.wikipedia.org/wiki/Semantic_theory_of_truth), is due to Alfred Tarski (https://en.wikipedia.org/wiki/Alfred_Tarski who, however, "[would not] claim [it was] the 'right' one [other than in* mathematics]*."*

However, we must always keep in mind that the deductive rules can spread falsehood like wildfire.

EXAMPLE 0.9. One of the deductive rules in ALGEBRA is that "*adding equals to equals yields equals*". Now:

- If we start from a *true* sentence like "$4+5$ and $6+3$ are equal", then the rule will force us to believe that, say, "$4+5+7$ and $6+3+7$ are equal" which is fine,

But:

- If we start from "9 and 8 are equal", then:
  - adding 9 and 8 to, say, 7 and 7 will force us to believe that 16 and 15 are equal.
  - adding 16 and 15 to 9 and 8 will force us to believe that 25 and 23 are equal,
  - adding 16 and 15 to 25 and 23 will force us to believe that 41 and 38 are equal,
  - Etc

Finally, just to clarify,

rigor

Cautionary note 0.1 A *scientific* theory is a much more complicated thing than a *mathematical* theory. See for instance https://en.wikipedia.org/wiki/Theory#Scientific, https://en.wikipedia.org/wiki/Theory#Philosophical_views, https://en.wikipedia.org/wiki/Scientific_theory#Theories_as_axioms, etc.

Reason Vs Rigor

So, since the foremost fear in MATHEMATICS is the fear of making a mistake in the use of the deductive rules after which every sentence risks to be false, the name of the game is to proceed as **rigorously** as possible, that is to provide as many steps in the use of the deductive rules—as possible while remaining "*readable*"—*and* always to be able and ready to provide missing steps on demand.

That's what referreing is all about.

Example 0.10. While'Delta functions' (https://en.wikipedia.org/wiki/Dirac_delta_function) had been used since the early eighteen hundreds, it was only in 1950 that Laurent Schwartz (https://en.wikipedia.org/wiki/Laurent_Schwartz) was awarded the Fields Medal[4] for having *defined* them rigorously.

The single quotes, ' ', say that, at this time, you are not supposed to know what the word means.

Indeed, CALCULUS has been extraordinarily difficult to develop *rigorously* (https://en.wikipedia.org/wiki/Nonstandard_analysis) and, as a result, the number-one worry for authors of CALCULUS texts is how much rigorous to be? A few texts, called REAL ANALYSIS (https://en.wikipedia.org/wiki/Real_analysis), are completely rigorous and the rest, just called CALCULUS, skip whatever the authors think will be too much for the prospective buyers of their book.

But the reason *this* text isn't rigorous is totally different: in contrast with textbooks that retain, however un-rigorously, the point of view of *mathematicians*, this text aims at the CALCULUS that *hard scientists* (https://en.wikipedia.org/wiki/Hard_and_soft_science) and *engineers* have been *using* forever—without worrying one bit about rigor. So, the mathematical conformists ought to be warned that, instead of being based on the use of 'limits' or 'infinitesimals' , as it seems all current CALCULUS textbooks are doing,

[2]Just about the very highest honors for a *mathematician*. (https://en.wikipedia.org/wiki/Fields_Medal)

CAUTIONARY NOTE 0.2 In this text, CALCULUS will be by way of 'polynomial approximations' which are the algebraic equivalent of the 'decimal approximations' used by *scientists* and *engineers* in **applications** of CALCULUS to the real world.
(As well as by *mathematicians* . . . in pure research!)

reason

As physicist David Hestenes of GEOMETRIC ALGEBRA fame said at the outset of his 2002 Oersted lecture: That "course content is taken [by many] as given [...] ignores the possibility of improving pedagogy by reconstructing course content."

Way To Learn?

Yet another way this text claims to be different has to do with the fact that "*Math Anxiety*" invariably originates with the standard textbooks, in the best cases because the textbook leaves so much "*going without saying*" that **reason** has become invisible, in the worst cases because the textbook has been gutted down to the disconnected "*facts and skills*" deemed necessary to pass some exam so that no reason remains.

In contrast, this text wants to do three things:

- As Einar Hille (https://en.wikipedia.org/wiki/Einar_Hille) wrote, "*Mathematics is neither accounting nor the theory of relativity. Mathematics is much more than the sum total of its applications no matter how important and diversified these may be. It is* a way of thinking."[3] (Emphasis added.)

  Of course, a way of thinking cannot be taught or even described and can only be *learned* from *experience*. Fortunately, as George Sarton (https://en.wikipedia.org/wiki/George_Sarton) wrote, "*It is only a matter of perseverance and of good will. Only thus will you acquire a* method *of thought. And if one cannot reproach anyone for being ignorant of this or that—for ignorance is not a sin—it is legitimate to reproach one with poor reasoning.* [. . .] [T]*his scientific sincerity is only achieved by the attentive study of a specific subject.*"[4]

  So, one thing this text wants to do is to facilitate your "*attentive study*" of CALCULUS by presenting and discussing matters in a way that will make *reasonable* sense to you.

  In other words, here, zero "show and tell".

- As John Holt (https://en.wikipedia.org/wiki/John_Holt_(educator)) wrote, "*Human beings are born intelligent. We are by nature question-asking, answer-making, problem-solving animals, and we are extremely good at it, above all when we are little. But under certain conditions,*

[3] Einar Hille, *Analysis*, 1964
[4] As quoted from his letters by his daughter, May Sarton, in her book *I Knew a Phoenix*

which may exist anywhere and certainly exist almost all of the time in almost all schools, we stop using our greatest intellectual powers, stop wanting to use them, even stop believing that we have them."[5]

*So, zero "*drill and test*" too.*

Which is why this text does *not* have **Exercises**: the important questions are those *you* will be asking yourself. All this text wants to do is to raise reasonable issues and deal with these issues up to a point where you will be equipped to look deeper into at least some of these issues.

The hope is you will.

Of course, you would be quite right asking how then will you know if you have learned CALCULUS but the answer still is: when *you* will have become able to answer most of *your* own questions. Better yet, though, is for two or three people to confront their understanding of this text.

That, *you surely won't like one bit! (At least for now.)*

- As Etienne Ghys (https://en.wikipedia.org/wiki/%C3%89tienne_Ghys) wrote, "*I have now learned that precision and details are frequently necessary in mathematics, but I am still very fond of promenades.* [...] *You have to be prepared to get lost from time to time, like in many promenades.* [...] *You will have to accept half-baked definitions.* [...] I am convinced that mathematical ideas and examples precede proofs and definitions."[6] (Emphasis added.)

That's because nobody is perfect. Not even authors.

And *that* really says it all.

*And this, Ladies and Gentlemen, is where (*Not Your Usual*) Preface comes to an end.*

∴

Or, if you can get hold of Simmons' book, the historical material Dudley praised.

Those curious about the history of CALCULUS might want to look up https://en.wikipedia.org/wiki/History_of_calculus or the shorter https://en.wikipedia.org/wiki/Calculus#History but, for those only *a little bit* curious, here is a very short version:

Veeery, very short!

> CALCULUS was invented in the late 1600s independently by Newton (initially by way of 'infinitesimals' but ultimately by way of 'limits') and Leibniz (solely by way of 'infinitesimals').
>
> The first of the many editions of the first calculus text, *Infinitesimal Calculus with Applications to Curved Lines*, by L'Hospital, is from 1696. (https://en.wikipedia.org/wiki/Guillaume_de_l%27H%C3%B4pital).
>
> Right away, *scientists*, *engineers* and *mathematicians* started using 'infinitesimals' routinely even though it was almost immediately realized that 'infinitesimals'—as well as 'limits'—were not rigorous.

If only because 'limits' can't be computed *but only* guessed *and then* checked *to see if they* are *the 'limit'.*

Guess what: 'infinitesimals' are still *mostly avoided like the plague!*

Arnold would have had the Fields Medal except his public opposition to the persecution of dissidents had led him into direct conflict with influential Soviet officials—and he suffered persecution himself, including not being

[5] John Holt *How Children Fail* A classic, first published in the 60s. Free download from https://archive.org/download/HowChildrenFail/HCF.pdf

[6] Etienne Ghys, *A singular mathematical promenade.* 2017. Free download from https://arxiv.org/abs/1612.06373

And when, over a century later, most *mathematicians* switched to 'limits' which had finally been made rigorous, *scientists*—and for a long time even *differential geometers*—stayed with 'infinitesimals'. (https://en.wikipedia.org/wiki/Calculus#Limits_and_infinitesimals).

Eventually, in 1961, Abraham Robinson (https://en.wikipedia.org/wiki/Abraham_Robinson), three years over the age limit for the Fields Medal, finally made "infinitesimals' rigorous .

Yet, as Vladimir Arnold (https://en.wikipedia.org/wiki/Vladimir_Arnold) wrote already in 1990: "*Nowadays, when teaching analysis, it is not very popular to talk about infinitesimal quantities. Consequently present-day students are not fully in command of this language. Nevertheless, it is still necessary to have command of it.*" (https://en.wikipedia.org/wiki/Infinitesimal).

In any case, a long time before all that, around 1800, Joseph-Louis Lagrange (https://en.wikipedia.org/wiki/Joseph-Louis_Lagrange), one of the greatest mathematicians ever, who explicitly wanted to free CALCULUS from "*any consideration of infinitesimals, vanishing quantities, limits and fluxions*", had developed the approach to CALCULUS by way of the 'polynomial approximations' to be used in this text. (https://en.wikipedia.org/wiki/Charles_Babbage#British_Lagrangian_School) Eventually, though, Lagrange realized that 'polynomial approximations' could not deal with certain *pathological* cases and reverted to 'infinitesimals'.

However, starting around 1880, yet another all time great mathematician, Henri Poincaré (https://en.wikipedia.org/wiki/Henri_Poincar%C3%A9), used 'polynomial approximations' to solve a very large number of problems so that Lagrange's 'polynomial approximations' are now known as 'Poincaré expansions' as well as 'asymptotic expansions' ...by mathematicians. (https://en.wikipedia.org/wiki/Asymptotic_expansion.)

But 'polynomial approximations*' are still often confused by* teachers *with 'Taylor* series*'. And* that *is* most *regrettable.*

When you have mastered numbers, you will in fact no longer be reading numbers, any more than you read words when reading books. You will be reading meanings.

Harold Geneen [1]

Chapter 0

Numbers For Calculating

Before considering CALCULUS according to the real world, it will be useful to consider a few aspects of ARITHMETIC according to the real world that are usually not granted much attention in ARITHMETIC textbooks.

No, no, this is not *going to be your usual* Review Of Basic Stuff *in disguise!*
You should read this Chapter 0 to have an idea of what's in there but don't panic: as you go on with the later chapters, there will alsways be clickable references and it will then make perfect sense.

To begin with, from the model theoretic standpoint,

CAUTIONARY NOTE 0.1 The word 'number' refers to the real world while a 'numeral' is the name of a 'number'. belongs to the paper world
See (https://en.wikipedia.org/wiki/Number) and https://en.wikipedia.org/wiki/Numeral

This being said, in accordance with Interpolation (AGREEMENT 3.1, Page 187),

[1] As discussed, very thoroughly, in https://history.stackexchange.com/questions/45470/source-of-quote-attributed-to-w-e-b-du-bois-when-you-have-mastered-numbers?rq=1, this famous quote is indeed from Geneen's book, *Managing*, Chapter Nine - The Numbers, p. 151, rather than from W.E.B. Dubois, as often asserted—with no reference.

qualifier
thing
qualifier

AGREEMENT 0.1 This text will *not* use the word 'numeral' but only the word 'number', leaving it to the reader which is actually intended.

1 The Numbers We Will Use

By itself, that is without **qualifier**, the word 'number' cannot be defined because *mathematicians*, *scientists*, and *engineers* all use many different kinds of 'numbers' for many different purposes. (https://en.wikipedia.org/wiki/Number)

1. Signed decimal numbers. In fact, even "the rest of us" use different kinds of 'numbers' depending on:

A. Whether the real world entity we want to deal with is:

- A collection of items

or

- An amount of stuff

and also on:

B. Whether the **information** we want about the real world entity is:

- The size of the real world entity *alone*,

or

- The size *together with* the orientation of the real world entity,

So, the word number should always be used with one of the following **qualifiers**

	Collection	Amount
Size *only*	plain whole number	plain decimal number
Size *and* orientation	signed whole number	signed decimal number

EXAMPLE 0.1.

- $783\,043$ is a plain whole numeral which may refer to a collection of *people*,
- 648.07 is a plain decimal numeral which may refer to an amount of *money*,
- $-547\,048\,308$ and $+956\,481$ are signed whole numerals,
- $-137.048\,8$ and $+0.048\,178$ are signed decimal numerals.

EXAMPLE 0.2.

- By "Numbers are beautiful", what will be intended is "*Signed decimal* numerals are beautiful",
- By "*Plain* numbers are weird", what will be intended is "*Plain* numerals, whether whole or decimal, are weird".

- By "*Decimal* numbers are handy", what will be intended is "*Decimal* numerals, whether plain or signed, are handy". number changeable set

However, in contradistinction with DISCRETE MATHEMATICS which deals exclusively with collections of items and therefore uses only whole numbers (https://en.wikipedia.org/wiki/Discrete_mathematics), CALCULUS deals only with amounts of stuff and we will use whole numbers only occasionally, mostly as explanatory backdrop for decimal numbers.

But since constantly having to use qualifiers would be unbearably burdensome,

> **AGREEMENT 0.2** In the absence of qualifier, **number** will be short for*signed decimal number.*

EXAMPLE 0.3. What will be intended by:

- "Numbers are beautiful" is "*Signed decimal* numbers are beautiful",
- "*Plain* numbers are weird" is "*Plain* numbers, whether whole or decimal, are weird".
- "*Decimal* numbers are handy" is "*Decimal* numbers, whether plain or signed, are handy".

2. Changeable number vs. set number. In the real world, numbers can occur in different ways:

- A number may be **changeable** in a situtation, that is in that situtation the number *can* be changed to any other number.

  EXAMPLE 0.4. The people of Jacksville are considering paving part of the parking lot next to Township Hall and since both the *length* and the *width* of the area to be paved are *changeable*, people are discussing the pro and con of 20ft long by 100 feet wide versus 60ft long by 60 feet wide versus 100ft long by 30 feet wide versus etc, etc.

- A number may be **set** in a situtation that is in that situtation the number *cannot* be changed to any other number.

  EXAMPLE 0.5. The people in Jillsburg are considering paving part of the road from City Hall down to the river but since the *width* is *set* by the County, only the *length* of the area to be paved is *changeable* and people are discussing the pro and con of 300 ft long versus 1000 ft long versus 500 ft long versus etc, etc.

0
syntactic
semantic
nothingness

Example 0.6. The circumference of a circle of diameter 702.36 is equal to 3.14159×702.36 (https://en.wikipedia.org/wiki/Circumference#Circle),
In that sentence, the number 702.36 is *changeable* to any other number to get the circumference of another circle but the number 3.14159 is *set*. (https://en.wikipedia.org/wiki/Pi)

2 Zero and Infinity

Zero and Infinity both play most important roles, different but reciprocal.

1. Zero. Already "*the ancient Greeks* [...] *seemed unsure about the status of zero as a number.*" (https://en.wikipedia.org/wiki/0#Classical_antiquity).

There are two difficulties with **0** that separate 0 from other numbers
i. The **syntactic** difficulty is that, at least to an extent, mathematicians use 0 only for convenience.

Example 0.7. When dealing with addition of signed numbers, not having 0 would create a difficulty when explaining what happens with adding *opposite* sign numbers.

On the other hand, as we will discuss in **??** **??** - **??** (**??**), the presence of 0 causes serious difficulties when dealing with *division*.

ii. The **semantic** difficulty, actually the more important one, is that **nothingness** does not exist in the real world in that there is no such thing as a zero amount of stuff.

Example 0.8.
- There is no such thing as a perfect vacuum. (https://en.wikipedia.org/wiki/Vacuum).
- There is no such thing as an absolute zero temperature. (https://en.wikipedia.org/wiki/Absolute_zero)

Of course, to say that the size of 0 is 0 merely moves the issue to plain numbers.
After months of waffling!

So the second difficulty is that since 0 does not denote *any* amount in the real world, 0 has no *size* and no *sign*.
Since it is standard practice, though, we will have to accept that

non-zero number
infinity
end of the line

CAUTIONARY NOTE 0.2 **0** is considered to be a *number* in spite of the fact that 0 does *not* denote *anything* in the real world.

So, what we will do is to distinguish **non-zero numbers**, that is all numbers except 0, from just numbers which include 0. So, all *non-zero* numbers have both a *size* and a *sign*.

Isn't playing it both ways convenient?

2. Infinity. Like zero, infinity is an ancient issue: "*Since the time of the ancient Greeks, the philosophical nature of infinity was the subject of many discussions among philosophers.*" (https://en.wikipedia.org/wiki/Infinity)

If only as "And so on."

i. There are indeed at least two difficulties with infinity:

- The semantic difficulty with infinity is that there is no such thing in the real world as an infinite collection of items or an infinite amount of stuff

  EXAMPLE 0.9. Neither the number of stars in the universe, nor the amount of energy, is infinite. Hugely huge, yes, beyond our ability even to imagine, yes, infinite, no.

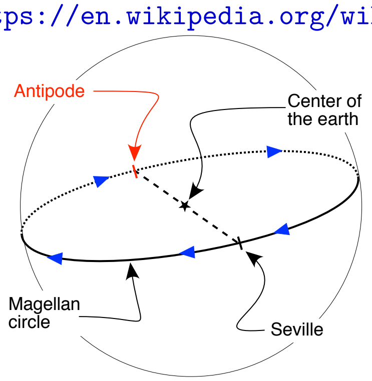
- Another difficulty with infinity is that, while numbers are endless, in the real world, sooner or later, we alway get to the **end of the line**. (https://en.wikipedia.org/wiki/End_(topology)) If we try to go farther and farther away, we have the feeling that the longer we go, the farther away we will get and that there is nothing to keep us from getting as far away as we want. But this is not really the case, nothing is endless .

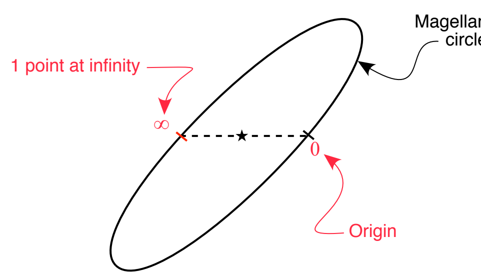
  EXAMPLE 0.10. Even though Magellan died in 1521 while trying to go as far away from Seville as he could, his ships kept on going west. And one of them eventually reached . . . home, bearing witness that there was no going around the fact that the earth is round. (https://en.wikipedia.org/wiki/Ferdinand_Magellan#Voyage)

compact
one-point compactification
Magellan
circle
∞
origin
two-points compactification
$+\infty$
$-\infty$

ii. As with zero, mathematicians tend to ignore the first difficulty but do acknowledge the second difficulty by occasionally **compactifying** numbers in either one of two ways:

- **One-point compactification**: Since what looks to us like a *straight line* is in reality just a piece of a **Magellan circle**, (https://www.quantamagazine.org/what-shape-is-the-universe-individuald-or-flat-20191104/), we can compactify a *straight line* into a Magellan circle by adding ∞, that is the point "down under" from the **origin**, as end of the line. (https://en.wikipedia.org/wiki/Projectively_extended_real_line)



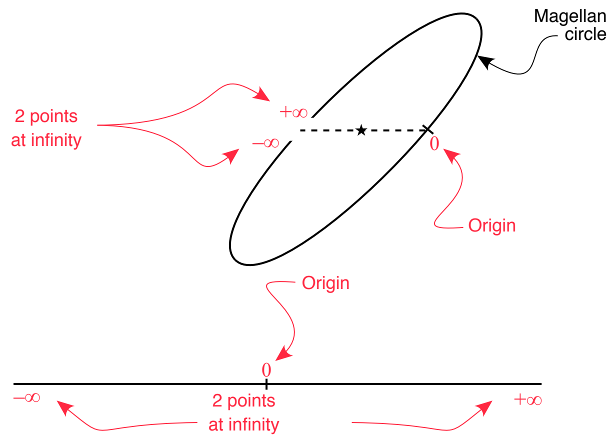


However,

CAUTIONARY NOTE 0.3 ∞ is *not* a *number*

- **Two-points compactification**: The other way mathematicians use to prevent numbers from being endless is to *end* the positive numbers and the negative numbers *separately*: the positive numbers with $+\infty$ and the negative numbers with $-\infty$. (https://en.wikipedia.org/wiki/Extended_real_number_line)

extended number
point
point



However,

> **CAUTIONARY NOTE 0.4** $+\infty$ and $-\infty$ are *not numbers* but what are often called **extended numbers**.

3 Numbers In General

If only for the sake of economy, other than in **EXAMPLES**, we won't deal with *given* numbers but with what applies to *any* number one could give.

1. Points. However, and in spite of CAUTIONARY NOTE 0.2 - 0 is a *dangerous* number (Page 5) and CAUTIONARY NOTE 0.3 - ∞ is *not a number* (Page 6), and because, for all their differences, we will be *using* 0, ∞, and non-zero numbers pretty much in the same way, it will be extremely convenient to use

> **DEFINITION 0.1** By the word **point**, we will mean any of the following:
> - Any non-zero number,
> - 0, (Even though 0 has no sign.)
> - ∞. (Even though ∞ is *not* a number.)

In particular, it will be extremely convenient to see the *points* ∞ and 0 as points that are reciprocal of each other.

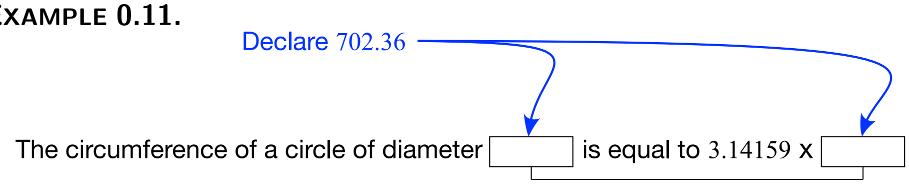
Nevertheless:

variable
formula
declare
global variable
x
y
z

Cautionary note 0.5 One cannot *compute* with points because the rules for computing with *non-zero* numbers and with 0 are different and we cannot compute with ∞ very much at all.

2. Global variables. A **variable** is a symbol that does *not* denote any particular number but which works like an *empty box* in a **formula** awaiting whatever number we will **declare** for the formula to become a sentence. (https://en.wikipedia.org/wiki/Variable_(mathematics))

Example 0.11.



yields the sentence at the beginning of Example A.9 (Page 506)

Because numbers can occur in different ways, we will be using different *kinds* of variables:

i. When we want to deal with *all* numbers (Inclunding 0) we will use

Definition 0.2 $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are (symbols for) **global variables** which can thus be replaced by *any* number we declare.

People against paving can vote for declaring 0.

Example 0.12. In Example A.7 (Page 505), until the people in Jacksville *declare* what they want, we can denote the length and the width of the area to be paved with the global variables x and y.

Language note 0.1 We are using the qualifier "global" to distinguish from another kind of standard variable which we will introduce in Section 12* - Real Numbers (Page 55) for which we will use the qualifier "local".

Cautionary note 0.6 Global variables can be replaced by 0 in view of Cautionary Note 0.2 - 0 is a *dangerous* number (Page 5) but *not* by ∞ in view of Cautionary Note 0.3 - ∞ is *not* a *number* (Page 6)

ii. Non-zero global variables. We mentioned in Subsection 3.1 - Points (Page 7) that there are situations in which we cannot use 0 and so, in these situatuions, we cannot use global variables either and we will use

non-zero global variable
$x_{\neq 0}$
$y_{\neq 0}$
$z_{\neq 0}$

> **DEFINITION 0.3** $\boldsymbol{x_{\neq 0}}$, $\boldsymbol{y_{\neq 0}}$, $\boldsymbol{z_{\neq 0}}$ are (symbols for) **non-zero global variables** which can thus be replaced by any *non-zero* number we declare.

> **LANGUAGE NOTE 0.2** Non-zero variables are *not* standard symbols and, instead, symbols for global variables are used together with a separate stipulation that only non-zero numbers can be declared. But we will find non-zero variables convenient because there is then no need to interrupt the flow.

EXAMPLE 0.13. One cannot write $\frac{1}{x}$ without writing somewhere not too far something like "$(x \neq 0)$" to prevent replacement by 0. So, *we* will write $\frac{1}{x_{\neq 0}}$.

iii. Set variable. There are situations where there is a particular number playing a particular role.

set variable
x_0
y_0

EXAMPLE 0.14. In a mystery novel, there usually is someone "who did it" and the detective has to use words like "the perpetrator" to say things like "the perpetrator wasn't looking for money", "the perpetrator's weapon", "Someone must have seen the perpetrator because ...". etc

So we will use

> **DEFINITION 0.4** $\boldsymbol{x_0}$, $\boldsymbol{y_0}$, $\boldsymbol{z_0}$ are (symbols for) **set variables** which can thus be replaced by any *set* numbers.

EXAMPLE 0.15. In EXAMPLE A.8 (Page 506), until the peope in Jillsburg declare what *length* they want to be paved, we can can symbolize the length *length* to be paved by y but we have to symbolize the width by x_0 because it *set* by the County.

> **LANGUAGE NOTE 0.3** There is no standard name for the symbols x_0, y_0, z_0 even though the symbols themselves are completely standard.
>
> We use "set" because the set numbers by which these variables can be replaced will be reset in each new situation. (https://en.

expression
generic expression
evaluate
at
declare
replace
individual expression
execute

wiktionary.org/wiki/set#Adjective)

EXAMPLE 0.16. In EXAMPLE A.8 (Page 506), while the County in which Jacksville and Jillsburg are set the width of paved roads at, say, 20 Ft, another County could set it at, say, 24 Ft.

3. Generic expression. In MATHEMATICS, an **expression** is formed according to rules with symbols that can be numbers, variables, operations, and 'functions', together with symbols to determine the order of operations. (Rephrased from https://en.wikipedia.org/wiki/Expression_(mathematics))

You may also want to look at an old, classic game: https://en.wikipedia.org/wiki/WFF_%27N_PROOF or https://americanhistory.si.edu/collections/search/object/nmah_694594/

For our purpose, though, it will be enough to define a **generic expression** as an expression involving at least one global variable.

EXAMPLE 0.17. In EXAMPLE 0.49 (Page 28), the cost to Jackstown for paving an x Ft. by y Ft. rectangular area at z Dollars per Sq.Ft. would be given by the generic expression Cost$(x, y, z) = x \times y \times z$.

EXAMPLE 0.18.

- To express that one times *any* number is equal to that number, we write the generic expression:
$$1 \times x = x$$
- To express that the order in which we add any two *numbers* does not matter, we write the generic expression:
$$x + y = y + x$$
- To express that the order in which we *add* any three wholenumbers does not matter, we write the generic expression:
$$(x + y) + z = x + (y + z)$$

4. Evaluation *at* a given number. We can usually **evaluate** a generic expression **at** a given number:

PROCEDURE 0.1 To evaluate a given generic expression in terms of x *at* a given number x_0:

i. Declare the given number x_0 by writing the declaration $\Big|_{x \leftarrow x_0}$, read "x *to be* replaced by x_0", to the right of the generic expression:

$$\text{generic expression in terms of } x \Big|_{x \leftarrow x_0}$$

ii. Replace every occurence of x in the generic expression in terms of x by the declared number x_0 to get the **individual expression** in terms of the given number x_0.

In standard CALCULUS *texts, step* ***i.*** *is usually omitted.*

$$\text{individual expression in terms of } x_0$$

iii. Execute the individual expression in terms of x_0, that is carry out the operations according to the rules.

DEMO 0.1a To evaluate the generic expression $\dfrac{x^{+2} \ominus +7}{x \oplus +3}$ *at* $+5$

i. We declare the given $+5$ by writing the declaration $\Big|_{x \leftarrow +5}$, read "x *to be* replaced by $+5$", to the right of the generic expression:

$$\frac{x^{+2} \ominus +7}{x \oplus +3}\Bigg|_{x \leftarrow +5}$$

ii. We get the individual expression for the given $+5$ by replacing every occurence of x in the generic expression by the given $+5$:

$$\frac{+5^{+2} \ominus +7}{+5 \oplus +3}$$

iii. We execute the individual expression in terms of $+5$, that is we carry out the operations according to the rules:

$$\frac{+25 \ominus +7}{+5 \oplus +3}$$

$$\frac{+18}{+8}$$

$+4.5$

Unfortunatley, there are three issues:

A. Generic expressions cannot be evaluated *at* ∞ in view of CAUTIONARY NOTE 0.3 - ∞ is *not a number* (Page 6),

B. Generic expressions cannot always be necessarily evaluated *at* a number x_0,

DEMO 0.1b To evaluate the generic expression $\dfrac{x^{+2} \ominus +7}{x \oplus +3}$ *at* -3

i. We declare the given -3 by writing the declaration $\Big|_{x \leftarrow -3}$, read "x *to be* replaced by -3", to the right of the generic expression:

$$\dfrac{x^{+2} \ominus +7}{x \oplus +3}\Bigg|_{x \leftarrow -3}$$

ii. We get the individual expression for the given -3 by replacing every occurence of x in the generic expression by the given -3:

$$\dfrac{-3^{+2} \ominus +7}{-3 \oplus +3}$$

iii. We try to execute the individual expression in terms of $+5$, that is we carry out the operations according to the rules but the execution comes immediately to a halt

$$\dfrac{+9 \ominus +7}{0}$$

$$\dfrac{+2}{0}$$

because when dividing, say \$2 among 0 people we can promise any share.

And things can easily turn out even more complicated. For instance:

DEMO 0.1c To evaluate the generic expression $\dfrac{x^{+2} \ominus +9}{x \oplus -3}$ *at* $+3$

digit
specify

i. We declare the given $+3$ by writing the declaration $\Big|_{x \leftarrow +3}$, read "x *to be* replaced by $+3$", to the right of the generic expression:

$$\left.\frac{x^{+2} \ominus +9}{x \oplus -3}\right|_{x \leftarrow +3}$$

ii. We get the individual expression for the given $+3$ by replacing every occurence of x in the general expression by the given $+3$:

$$\frac{+3^{+2} \ominus +9}{+3 \oplus -3}$$

iii. We try to execute the individual expression in terms of $+3$, that is we carry out the operations according to the rules but the execution comes immediately to a halt:

$$\frac{+9 \ominus +9}{0}$$

$$\frac{0}{0}$$

because dividing, say \$0 among 0 people, allows for *any* share.

C. Last but not least and as we will see in Section 1 - Global Input-Output Rules (Page 207), inasmuch as CALCULUS is intended to deal with '*change*', evaluating a generic expression *at* a number x_0 will *not* provide enough information for us to figure out changes *near* x_0.

EXAMPLE 0.19. We cannot tell the speed of a car from just a *still* picture.

We *will* be able to overcome all these dificulties but only in Section 11 - Neighborhoods (Page 40) because we will first have to look at several things and then build the necessary machinery.

4 Real-world Numbers

An important feature of a decimal number is how many **digits** the number has. (https://en.wikipedia.org/wiki/Numerical_digit) *Other than the digit 0!*

And then, when we want to ***specify***, that is when we want to denote *in order to bring about a specific result* (https://www.thefreedictionary.

figure
significant

com/specify), there is a difference between what is necessary to specify a collection of items and what is necessary to specify an amount of stuff.

1. Non-zero digits The more non-zero digits (whether left or right or both sides of the decimal point) a number has, the less likely the number is to denote anything in the real world..

Example 0.20. None of the following numbers

+602 528 403 339 145 485 295 666 038 294 891 392 775 987 378 000 261 234 386 384 558 003 000 384 203 771 790 349 303 000 000 000 809 234 329 262 234 085 108 500 000 002 891 038 456 666.

−0.602 528 403 339 145 485 295 666 038 294 891 392 775 987 378 000 261 234 386 384 558 003 000 384 203 771 790 349 303 000 000 000 809 234 329 262 234 085 108 500 000 002 891 038 456 666

+602 528 403 339 145 485 295 666 038 294 891 392 775 987 378 000 261 234 386 384 558 003 000 384 203 771 790 349 303 000 000 000 809 234 329.262 234 085 108 500 000 002 891 038 456 666

is likely to denote anything in the real world.

> **Language note 0.4 Figure** is the name often used instead of digit but In this text we will stick to digit,

2. Significant digits. Indeed, only so many digits can be **significant**, that is can be describing anything in the real world. (https://en.wikipedia.org/wiki/Significant_figures

Example 0.21. To say that "*the estimated population of the US was "*328 285 992 *as of January 12, 2019*" (https://en.wikipedia.org/wiki/DEMOgraphy_of_the_United_States on 2019/02/06) is not reasonable because at least the rightmost digit, 2, is certainly *not* significant: on that day, some people died and some babies were born so the population could just as well been denoted as, say, 328 285 991 or 328 285 994.

Example 0.22. Thenumbers given in https://en.wikipedia.org/wiki/Iron_and_steel_industry_in_the_United_States are much more reasonable: '*In 2014, the United States [...] produced' 29 million metric tons of pig iron and 88 million tons of steel.*" Similarly, "*Employment as of 2014 was 149,000 people employed in iron and steel mills, and 69,000 in foundries. The value of iron and steel produced in 2014 was 113 billion.*"

measure
uncertainty
differ
error

Identifying significant digits, however, is not a simple matter (https://en.wikipedia.org/wiki/Significant_figures#Identifying_significant_figures) and neither is determining in the result of a calculation which digits will be significant (https://en.wikipedia.org/wiki/Significant_figures#Arithmetic).

Neither of which will be considered in this text!

3. Size of a collection of items. In the case of a collection of items, things are simple because:

- the whole number *counted* to *denote* how many items we *get*

will *never* differ from

- the whole number *given* to *denote* how many items we *wanted*,

so that, in order to specify a collection of items we need only specify how many items we want in the collection.

Which, of course, is not to say that people—deliberately or not— will never miscount.

4. Size of an amount of stuff. While, in order to denote the amount of stuff we *want* we need only give a number, in order to denote what we *get* is another matter because we have to **measure** this amount of stuff and there will always be some **uncertainty** in the measurement because of such things as the quality of the equipment used to measure the amount, the ability of the person doing the measurement, etc. Therefore, the measured number will always **differ** from the given number by some **error**.

EXAMPLE 0.23. We cannot really say "we *have* 2.3 quarts of milk" because what we really have depends on the care with which the milk was measured. The *uncertainty* may of course be too small to matter . . . but then may not.

As Gowers (Fields Medal 1998) put it (6^{th} paragraph of https://www.dpmms.cam.ac.uk/~wtg10/continuity.html), "[. . .] *a measurement of a physical quantity will not be an exactly accurate infinite decimal. Rather, it will usually be given in the form of a finite decimal together with some error estimate:* $x = 3.14 \pm 0.02$ *or something like that.*" [Where 3.14 ± 0.02 is to be read as $3.14\pm$ some number smaller-size than 0.02]

At this point, you ptobably won't be able to make much of the rest of Gowers' paper but even a cursory glance will show his concern with the real world. Rare for a mathematician!

5. Specifying an amount of stuff. In Section 1 - The Numbers We Will Use (Page 2), we pointed out that we use different 'numbers' to denote what we *have* or what we *want*. But while using numbers is *sufficient* to denote what we *have*, in the case of an amount of stuff using numbers is *not sufficient* to specify what we *want*

- the decimal number *measured* to *denote* how much stuff we *get*

tolerance
significant

will *always* differ by some error from

- the decimal number *given* to *denote* how much stuff we *wanted*,

> **CAUTIONARY NOTE 0.7** A number cannot specify an amount just by *itself*.

So, *scientists* and *engineers* use specifications that consist of *two* numbers:

- a number to denote how much stuff they *want*,
- a number to denote the **tolerance** that is the size of at most how much the measured number is allowed to differ from the given number (https://en.wikipedia.org/wiki/Engineering_tolerance).

EXAMPLE 0.24. We *can* order "6.4 quarts of milk" with an error of at most 0.02 quarts of milk.

But, rather unfortunately, it is standard to write, as Gowers did above:

$$x = x_0 \pm T$$

that is that the measured number is *equal* to the given number $\pm$ the tolerance!

EXAMPLE 0.25. Strictly speaking, it makes no sense to specify $+6.4\ \pm 0.02$ because that would specify $+6.42$ or $+6.38$. But what is meant by that is that we are specifying $+6.4 \oplus$ a signed error whose *size* is smaller than 0.02 where 0.02 is the given *tolerance*.

We can then restate CAUTIONARY NOTE 0.7 - A number cannot specify an amount just by *itself* (Page 16) in a more constructive manner:

> **CAUTIONARY NOTE 0.7 (Restated) A number cannot specify an amount just by *itself***

EXAMPLE 0.26. While we cannot *specify* an amount of 6.4 quarts of milk (?? (??), we *can* specify an amount of 6.4 quarts of milk with a *tolerance* of 0.02 quarts of milk: what *can* then be poured will be $6.4 \pm$ a plain decimal number of quarts of milk *smaller than* 0.02.

Of course, not all differences are **significant**, that is carry information that is relevant to the real world situation.

EXAMPLE 0.27. The difference between \$3. and \$8. is the same as the difference between \$1 000 000 003. and \$1 000 000 008., namely \$5.. However, while the difference between \$3. and \$8. is *significant* because \$5. is in the same range as \$3. and \$8., the difference between \$1 000 000 003. and \$1 000 000 008.

is ... *insignificant* because \$5. is *much smaller* than both \$1 000 000 003. and \$1 000 000 008..

6. Real world numbers. Like all *scientists* and *engineers*,

real world number
real world number
give
quantitative ruler
tickmark
scale

And real world numbers are not at all the same as 'Real Numbers' which will be discussed in **?? ?? - ??** *(**??**)*

DEFINITION 0.5 By **real world numbers**, we will mean (signed decimal) numbers all whose digits are significant.

And so, from now on,

AGREEMENT 3.1 (Restated) Interpolation will be short for real world number.

7. Giving a number. There are at least three ways to **give** a number:

- The number *itself* can be given

EXAMPLE 0.28. Consider the number -107.53

- The number can be given as the result of a calculation

EXAMPLE 0.29. Consider the number that results from multiplying $+41.06$ and $+0.0317$

- The number can be given as the solution of an equation.

EXAMPLE 0.30. Consider the number 3 copies of which multiply to $+27$.

So, given numbers, in particular the numbers we will use in **EXAMPLES** and **DEMOS**, will of course be real world numbers.

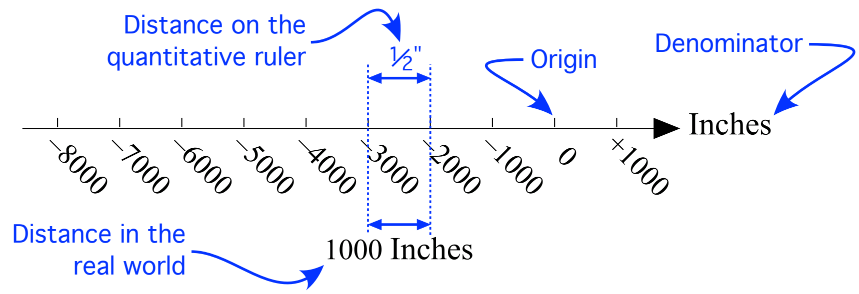
5 Picturing Real World Numbers

1. Quantitative rulers. So, to picture numbers, we will use **quantitative rulers** which are essentially just what goes by the name of "ruler" in the real world, that is an oriented straight line with *equidistant* **tickmarks** together with a denominator.

The **scale** of a quantitative ruler is the ratio of *any* distance on the quantitative ruler to the corresponding distance in the real world (https://en.wikipedia.org/wiki/Scale_(represent)

number line
axis
side

Example 0.31. The following :

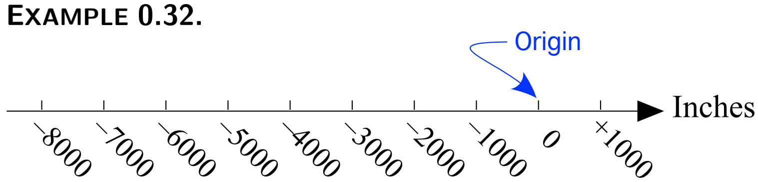


is a *quantitative* ruler whose scale is $\frac{\frac{1}{2}\text{inch}}{1\,000\,\text{inch}} = \frac{1}{2\,000}$

> **Language note 0.5 Number line** is the name most often used instead of quantitative ruler but another often used name is **axis** However, in this text we will stick to quantitative ruler.

2. Origin. Quantitative rulers must have a tickmark labeled 0 as an origin,

Example 0.32.



To know where the origin is is necessary because:

- The sign of a number says which **side** of the origin the number is—as seen when facing 0—and we will agree that

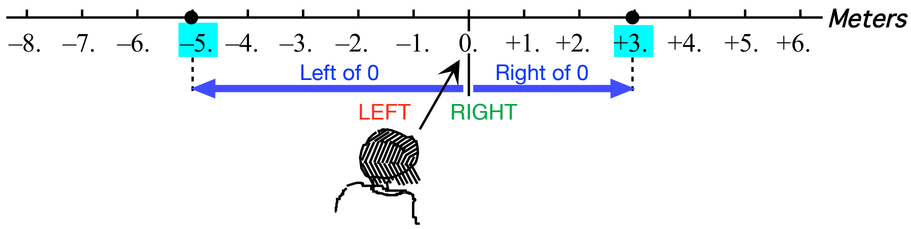
> **Agreement 0.3** When facing 0,
> - Positive numbers (+ sign) will be to the *right* of the origin 0,
> - Negative numbers (− sign) will be to the *left* of the origin 0.

Example 0.33. On a quantitative ruler,

Since Sign $-5 = -$, the number -5 is *left* of 0.

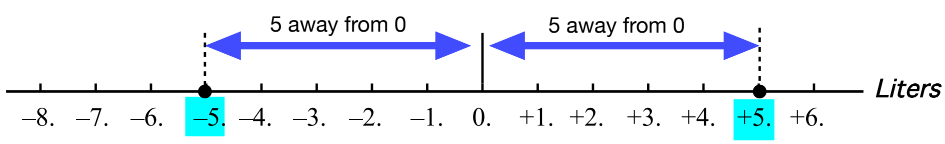
Since Sign $+3 = +$, the number $+3$ is *right* of 0.

symmetrical



- The size of a number says *how far* from 0 the number is on a quantitative ruler. Since opposite numbers have the same size, opposite numbers are **symmetrical** relative to the origin.

EXAMPLE 0.34. Thenumbers -5.0 and $+5.0$ have the same *size*, namely 5.0, so they are equally far from 0:



6 Computing with Real World Numbers

There are also several issues we need to bring up that all have to do with the fact that computing with *signed* numbers automatically involves computing with *plain* numbers, thereby creating a risk of confusion.

1. Comparing given numbers. The most important thing to keep in mind is that :

i. Comparing *signed* numbers (**??** **??** - **??** (**??**)) is quite different from comparing *plain* numbers—even though we use the same symbols, $<$, $>$, and $=$, with both plain numbers and signed numbers:

- *Positive* numbers compare the *same* way as their sizes,
- *Negative* numbers compare the *opposite* way from their sizes,

and:

- given numbers with opposite signs compare *regardless* of their sizes.

and

ii. The everyday use of *plain* numbers with *words* instead of *symbols* to indicate the orientations can make using the words *larger than*, *smaller than* and *equal to* quite confusing.

$\oplus$
$\ominus$
addition
subtraction

EXAMPLE 0.35. In everyday language, we say that

A \$700 *expense* is *larger* than a \$300 *expense*

because 700 is larger than 300 but the word *expense* cannot be seen as just meaning $-$ because

-700 is *smaller* than -300.

CAUTIONARY NOTE 0.8 Larger than, smaller than, equal to have different meanings depending on whether we are comparing *signed* numbers or comparing *plain* numbers.

.

2. Adding and subtracting given numbers. Notice that we have been using $+$ and $-$ not only as symbols for *addition* and *subtraction* of *plain* numbers, both *whole* and *decimal*, in spite of these being already quite different sets of numbers, but now also as symbols to distinguish positive numbers from negative numbers.

So, to avoid confusion as much as possible,

?? ?? - ?? (??) uses the symbols $\oplus$ and $\ominus$.

DEFINITION 0.6 **$\oplus$ and $\ominus$**, read "oplus" and "ominus", will be the symbols we will use for addition and subtraction of *signed* numbers.

Which is presumably why, say, $+13.73$ and -78.02 used to be written as ${}^{+}13.73$ and ${}^{-}78.02$ since ${}^{+}13.73 - {}^{-}78.02$ has the same advantages as $+13.73 \ominus -78.02$.

While the main point of the $\bigcirc$ around the symbol is to remind us to take care of the signs, another benefit is that using $\oplus$ and $\ominus$ lets us avoid having to use lots of parentheses.

EXAMPLE 0.36. Instead of the standard

$-23.87 + (-3.03)$, $-44,29 - (+22.78)$, $+12.04 - (-41.38)$

we will write:

$-23.87 \oplus -3.03$, $-44,29 \ominus +22.78$, $+12.04 \ominus -41.38$

which makes it clear without using parentheses which are symbols for calculations and which are symbols for signs.

THEOREM 0.1 Opposite numbers add to 0:

$$x = \text{Opposite } y \qquad \text{iff} \qquad x \oplus y = 0$$

3. Multiplying and dividing real world numbers.

i. While we could use the symbol $\otimes$ for the multiplication of *signed* given numbers, we will use the symbol $\odot$ because the symbol $\cdot$ is the usual practice in CALCULUS texts.

ii. Similarly, while we could use the symbol $\oplus$ for the division of *signed* given numbers, we will use the fraction bar $\_\_$ because it is the usual practice in CALCULUS texts.

$\odot$ $\_\_$

?? ?? - ?? (??) uses the symbols $\odot$ and $\_\_$.

For good reasons as you will see. And no circle around that one either!

EXAMPLE 0.37.

$+2 \odot +5 = +10$, $+2 \odot -5 = -10$, $-2 \odot +5 = -10$, $-2 \odot -5 = +10$

$\frac{+12}{+3} = +4$, $\frac{+12}{-3} = -4$, $\frac{-12}{+3} = -4$, $\frac{-12}{-3} = +4$,

THEOREM 0.2 Reciprocal *non-zero* numbers multiply to $+1$

$$x_{\neq 0} = \text{Reciprocal } y_{\neq 0} \quad \text{iff} \quad x_{\neq 0} \oplus y_{\neq 0} = +1$$

=======Begin WORK ZONE=======

4. Operating with 0. As warned in CAUTIONARY NOTE 0.2 - 0 is a *dangerous* number (Page 5), the behavior of 0 with multiplication and division causes difficulties.

i. Both 0 multiplied by any number and any number multiplied by 0 result in 0:

$$0 \otimes x = 0 \text{ and } x \otimes 0 = 0$$

The difficulty is that, while we are used to conclude from, say $x \otimes -7 = y \otimes -7$, that $x = y$, we cannot do so from $x \otimes 0 = y \otimes 0$

ii. We cannot divide a number, any number, by 0 because what would be the share in the real-world if we could divide 7 apples among 0 people?

And then, while we can say that $12 \div 4 = 3$ because when we share in the real-world 12 apples among 4 people each person gets 3 apples, to say that $12 \div 0 =$ some number would mean that $0 \times$ that number $= 12$ as menioned in **i.**

iii. Since:

- We cannot divide 1 by 0,

and

- ∞ is not even a number,

we cannot say, as much as we would want to, that 0 and ∞ are reciprocal of each other.

But wait, don't despair, we will, we will. Just be a bit patient.

========End WORK ZONE=======

5. Operating with more than two given numbers Given three numbers, let's call them Number One, Number Two, Number Three (which rule

may or may not be the same) and two operations, let's call them operation one and operation two (which may or may not be the same):

Number One operation one Number Two operation two Number Three

the overall result will usually depend on the order in which we perform the operations.

EXAMPLE 0.38. $-3 \ominus +5 \ominus -7$ **a.** $\underbrace{\underbrace{-3 \ominus +5}_{-8} \ominus -7}_{-1}$ **b.** $\underbrace{-3 \ominus \underbrace{+5 \ominus -7}_{+12}}_{-15}$

EXAMPLE 0.39. $-3 \odot +5 \oplus -7$ **a.** $\underbrace{\underbrace{-3 \odot +5}_{-15} \oplus -7}_{-22}$ **b.** $\underbrace{-3 \odot \underbrace{+5 \oplus -7}_{-2}}_{+6}$

So, another reason to use ⊕, etc as that keeps the number of parentheses down.

i. So, to indicate which operation(s) is/are intended to be performed *ahead of the other(s)*, one uses parentheses, ( ).
However, when one attempts to *minimize* the number of parentheses, stating "rules" to indicate the order in which operations are to be performed is actually a surprisingly complicated issue. (See `https://en.wikibooks.org/wiki/Basic_Algebra/Introduction_to_Basic_Algebra_Ideas/Order_of_Operations` and/or `https://en.wikipedia.org/wiki/Order_of_operations`)

Because we will want to focus on the CALCULUS rather than on the ALGEBRA.

ii. So, in order to keep matters as simple as possible, this text will always use however many parentheses will be necessary and we will just agree that the only

In other words, no PEMDAS, no BEDMAS, no BODMAS, no BIDMAS. (https://en.wikipedia.org/wiki/Order_of_operations)
Just WYSIWYG.

> **AGREEMENT 0.4** Computable expressions are expressions that, after parentheses surrounding a *single* number (if any) have been removed,
> **Rule A.** Do *not* include only *one* parenthesis (left or right),
> **Rule B.** May include two surrounding parentheses.

EXAMPLE 0.40. In EXAMPLE 0.16, using AGREEMENT 0.3 - Sides of the origin (Page 18),

a. With $(-3 \ominus +5) \ominus -7$,

- We cannot perform $\ominus$ as the expression $+5) \ominus -7$ breaks **Rule A**.
- We can perform $\ominus$ as the expression $(-3 \ominus +5)$ complies with **Rule B**.

The computation would thus be writen:

size-compare

$$(-3 \ominus +5) \ominus -7 \underbrace{= (-8) \ominus -7}_{\text{Step can be skipped}} = -8 \ominus -7 = -1$$

b. With $-3 \ominus (+5 \ominus -7)$,

- We cannot perform $\ominus$ as the expression $-3 \ominus (+5$ breaks **Rule A**.
- We can perform $\ominus$ as the expression $(+5 \ominus -7)$ complies with **Rule A** and **Rule B**. The computation would thus be written:

$$-3 \ominus (+5 \ominus -7) \underbrace{= -3 \ominus (+12)}_{\text{Step can be skipped}} = -3 \ominus +12 = -15$$

EXAMPLE 0.41. In EXAMPLE 0.17 (Page 10) 0.17, using AGREEMENT 0.3 - Sides of the origin (Page 18),

a. With $(-3 \odot +5) \oplus -7$:

- We cannot perform $\oplus$ as the expression $+5) \oplus -7$ breaks **Rule A**.
- We can performe $\odot$ as the expression $(-3 \odot +5)$ complies with **Rule B**.

The computation would thus be writen:

$$(-3 \odot +5) \oplus -7 \underbrace{= (-15)}_{\text{Step can be skipped}} \oplus -7 = -15 \oplus -7 = -22$$

b. With $-3 \odot (+5 \oplus -7)$:

- We cannot perform $\odot$ as the expression $-3 \odot (+5$ breaks **Rule A**.
- We can perform $\oplus$ as the expression $(+5 \oplus -7)$ complies with **Rule B**.

The computation would thus be written:

$$-3 \odot (+5 \oplus -7) \underbrace{= -3 \odot (-2)}_{\text{Step can be skipped}} = -3 \odot -2 = +6$$

7 Size-comparing Real World Numbers

Aside from comparing signednumbers as we did in Subsection 7.1 - Size-comparing vs. comparing sizes (Page 24), we can also **size-compare** (signed) given numbers, that is we can compare the (signed) numbers in terms of *only* their sizes and *regardless* of their signs:

DEFINITION 0.7 Given two (signed) numbers x and y,

- x is **smaller-size** than y iff Size x is *smaller* than Size y,
- x is **larger-size** than y iff Size x is *larger* than Size y,

smaller-size
larger-size
equal-in-size

- x is **equal-size** to y iff Size x is *equal* to Size y, (So, iff x and y are either equal or opposite.)

EXAMPLE 0.42.
- -234 is larger-size than $+32$ (Even though -234 is smaller than $+32$)
- $+71$ is smaller-size than -728 (Even though $+71$ is larger than -728)
- -35 is equal-size to $+35$. (Even though -35 and $+35$ are *opposite*.)

In particular:

THEOREM 0.3 Sizes of *reciprocal* numbers:
- The larger-size a non-zero number is, the smaller-size its reciprocal, and
- The smaller-size a non-zero number is, the larger-size its reciprocal.

Getting there, eh?

Proof. zzzzz □

1. Size-comparing vs. comparing sizes. There is a big difference between:
- size-comparing two signed given numbers

and
- comparing the *sizes* of two signed numbers

In the first case, *what* we are talking about are *signed* numbers—as it happens from the point of view of their sizes, while, in the second case, *what* we are talking about are *plain* numbers—which happen to be sizes of *signed* numbers.

EXAMPLE 0.43. There is a big difference between
- Age-comparing two people,

and
- Comparing the ages of two people.

In the first case, *what* we are talking about are *people* while, in the second case, *what* we are talking about are *numbers*.

EXAMPLE 0.44. Since:
i. The size of -7 is 7
ii. The size of -3 is 3
Then,

The *size* of -7 is larger than the *size* of $+3$

is a statement about the *sizes* of -7 and $+3$ but

-7 is larger-size than $+3$,

is a statement about thenumbers -7 and $+3$ *themselves*.

2. Procedure. But *size-comparing* is almost invariably confused with "*comparing sizes*. And, because we always want to know *what* we are talking about and to avoid any confusion in the matter, it will be convenient to use:

PROCEDURE 0.2 To size-compare two (signed) numbers

i. Get the *plain* numbers that are the sizes of the given numbers,
ii. Compare the *plain* numbers,
iii. Use DEFINITION 0.4 - $\boldsymbol{x_0}$, $\boldsymbol{y_0}$, $\boldsymbol{z_0}$ (Page 9).

DEMO 0.2
To *size-compare* the signed numbers -7.5 and $+3.2$

i. We get their *sizes*: the size of -7.5 is 7.5 and the size of $+3.2$ is 3.2
ii. We compare their *sizes*:

Since $7.5 > 3.2$

size -7.5 *is-larger-than* size $+3.2$

iii. Using DEFINITION 0.4 - $\boldsymbol{x_0}$, $\boldsymbol{y_0}$, $\boldsymbol{z_0}$ (Page 9), we conclude that

-7.5 *is-larger-size-than* $+3.2$

The trouble in most textbooks, though, is that the first step is the only one that is explicited while the rest is supposed to "*go without saying*", perhaps because, unfortunately,

CAUTIONARY NOTE 0.9 There are *no symbols* for size-comparisons of numbers.

so that, in view of CAUTIONARY NOTE 0.4 - $+\infty$ and $-\infty$ are *not numbers* (Page 7), we will have to say larger-size and smaller-size, in so many words, as in DEFINITION 0.3 - **Non-zero global variables** (Page 9).

3. Picturing size-comparisons of given numbers Given two numbers,

individualr
farther

- The smaller-size number is **individualr** to 0 than the larger-size number,
- The larger-size number is **farther** from 0 than the smaller-size number.

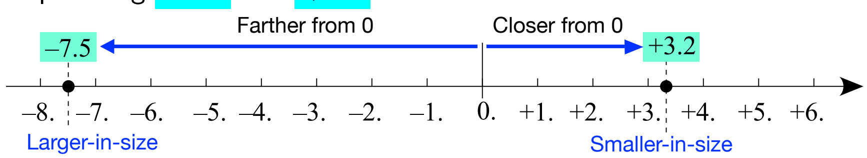
EXAMPLE 0.45. Given the numbers -7.5 and $+3.2$, we saw in EXAMPLE 1.27 (Page 93) that

- -7.5 is *larger-size-than* $+3.2$,

and therefore that

- $+3.2$ is *smaller-size-than* -7.5,

After picturing -7.5 and $+3.2$



we see that

- -7.5 is *farther from* 0 than $+3.2$,
- $+3.2$ is *individualr to* 0 than -7.5,

8 Qualitative Sizes

Mathematicians calculate in exactly the same way with *all* (signed decimal) numbers, regardless of their size.

EXAMPLE 0.46. $+0.3642$ and -105.71 are added, subtracted, multiplied and divided by exactly the same rules as $-41\,008\,333\,836\,092.017$ and -0.000001607.

And, of course, one a bit more complicated.

In theory, we can of course give *any* (signed decimal)numbers we want but, in the real world, things work in a different way.

1. Sizes beyond belief. To begin with, there are unbelievably many numbers that are unbelievably larger-size than any number you care to imagine as well as unbelievably many numbers that are unbelievably smaller-size than any number you care to imagine:

- We all went through a stage as children when we would count, say, "*one, two, three, twelve, seven, fourteen, ...*" but soon after that we were able to count properly and then we discovered that there was no largest number: we could *always* count one more. (Of course, counting backwards into the negative numbers has no end either so there is no largest-size number.) But that was only the tip of the iceberg.

EXAMPLE 0.47. Start with, say -73.8, and keep multiplying by 10 by moving the decimal point to the *right*, inserting **0**s *left* of the decimal point when it becomes necessary

-73.8
-738.
-7 38**0**.
-73 80**0**.
-738 00**0**.
-7 380 00**0**.
. . .
-738 000 000 000 000 000 000 000 000 000 00**0**.

This last number is probably already a lot larger-size than any number you are likely to have ever encountered.

*If not, just keep inserting **0**s until you get there!*

(See https://en.wikipedia.org/wiki/Large_numbers#Large_numbers_in_the_everyday_world)

- On the other hand, as children knowing only *plain* whole numbers, we thought there was a number smaller than all other numbers, namely 1 or perhaps 0. With decimal numbers, though, there is no smallest-size number.

EXAMPLE 0.48. Start with, say 41.6, and keep dividing by 10 by moving the decimal point to the *left*, inserting **0**s *right* of the decimal point when it becomes necessary.

41.6
4.16
0.416
0.**0**41 6
0.**0**04 16
0.**0**00 416
0.**0**00 041 6
0.**0**00 004 16
. . .
0.**0**00 000 000 000 000 000 000 000 000 000 000 000 004 16

This last number is probably already a lot smaller-size than any number you are likely to have ever encountered in a real world situation.

*If not, keep inserting **0**s until you get there!*

range
metric
out-of-range

2. Ranges of numbers. For *scientists* and *engineers*, numbers fall into size **ranges** that depend of course on what they are doing.

Example 0.49. The numbers that *astrophysicists* (https://en.wikipedia.org/wiki/Astrophysics) give and the numbers that *nanophysicists* (https://en.wikipedia.org/wiki/nanophysicist) give definitely fall into entirely different size ranges.

> In this regard, the following are well worth looking up:
> - The 9 minutes 1977 classic video at (the *bottom* of) http://www.eamesoffice.com/the-work/powers-of-ten/,
> - Terence Tao (Fields Medal 2006, http://terrytao.files.wordpress.com/2010/10/cosmic-distance-ladder.pdf.
>
> (Notice that it's all about *distances* which are *sizes*.)

Of course, units of the appropriate size allow the use of numbers in whatever size range is convenient—which is one reason why *scientists* and *engineers* use **metric** units: the conversion of metric units is easy because it involves only moving the decimal point without changing the digits.

Scientists *work hard enough not to bother with inconvenient units but many* engineers *in this country have to.*

Example 0.50. In the US Customary System,
- Instead of talking about 38016 inches, we usually say 0.6 miles,
- Instead of talking about 0.01725 tons, we usually say 34.5 pounds.

while, in the Metric System,
- Instead of talking about \$3 370 000., we usually say 3.37 MegaDollars.
- Instead of talking about 0.0000074 Meters, we usually say 7.4 microMeters.

=======Begin WORK ZONE=======

Since $+1$ unit and -1 unit are most likely to be in any range,

> **Agreement 0.5** $+1$ and -1 will *always* be within the range.

========End WORK ZONE=======

3. Out-of-range numbers By the same token, for *scientists* and *engineers*,, in any real world situation there will be numbers that will be **out-of-range**.

Example 0.51. Numbers like

a. $-7\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000$
$000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000$

000 000 000 000 000 000 000 000 000 000 000 000 000 000.

or

b. $-0\,.000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000\,000$
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 003
are *not* very likely to be within any range.

cutoff size
upper
lower

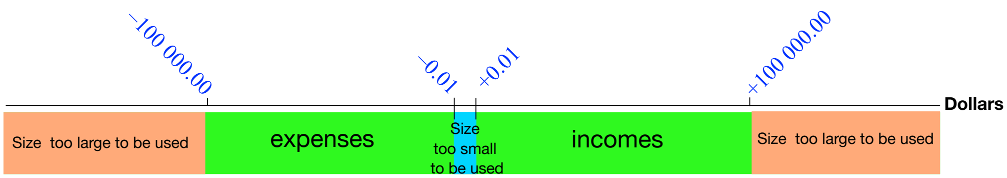
4. Cutoff sizes. So, in any real world situation, there will be two **cutoff sizes** that determine the range:

- An **upper** cutoff size *above* which numbers will surely *not* denote anything in the situation,
- A **lower** cutoff size *below* which numbers will surely *not* denote anything in the situation.

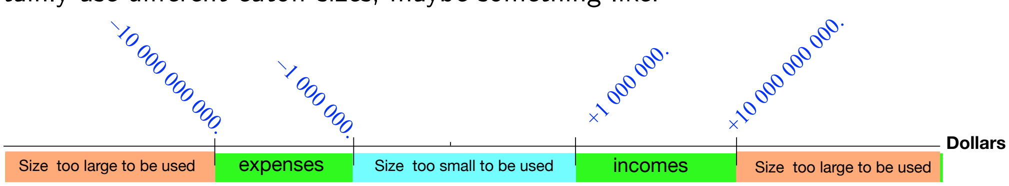
Unfortunately, often left to go without saying.

Of course, the upper cutoff size and the lower cutoff size will likely be different in different real world situations.

EXAMPLE 0.52. A small business could take 100 000.00 and 0.01 as cutoff *sizes* for their accounting system as it probably would never have to deal with amounts such as $\$-1\,058\,436.39$ or $\$+0.00072$.

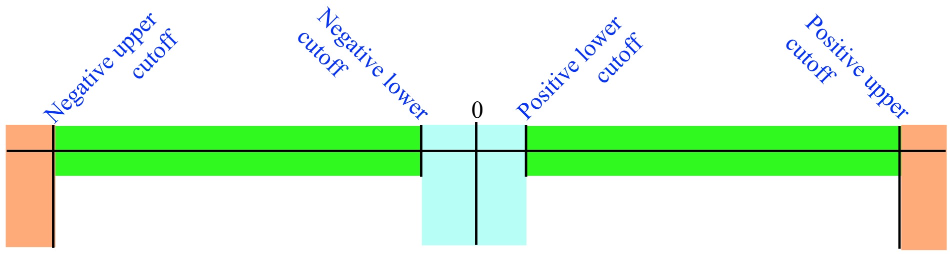


In contrast, the accounting system for a multinational corporation would certainly use different cutoff sizes, maybe something like:



5. Numbers we can or cannot give. Then, given cutoff sizes,

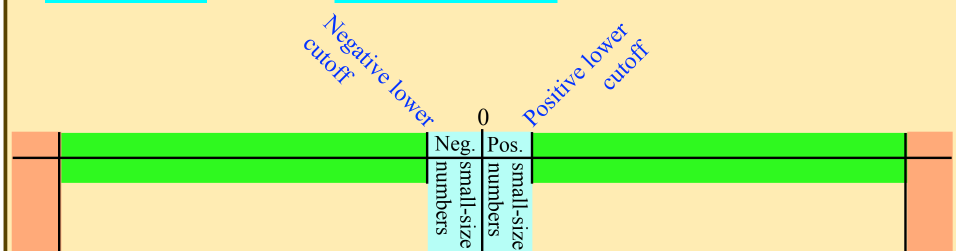
qualitative size
small-size
small variable
h
large-size



6. Qualitative sizes. We can define the following **qualitative sizes** for numbers:

i. The numbers whose size is too small for us to *give*:

DEFINITION 0.8 **Small-size** numbers are number that are smaller-size than the *lower* cutoff size

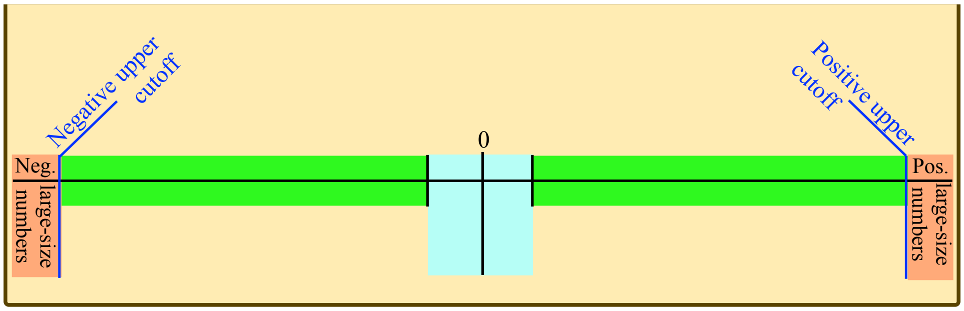


DEFINITION 0.9 The small variables $\boldsymbol{h}$, $\boldsymbol{k}$, ... will be the (standard) symbols for generic *small-size* numbers.

CAUTIONARY NOTE 0.10 because 0 has *no* size to begin with. (CAUTIONARY NOTE 0.2 - 0 is a *dangerous* number (Page 5))

ii. The numbers whose size is too large for us to *give*:

DEFINITION 0.10 **Large-size** numbers are numbers that are larger-size than the *upper* cutoff size ,



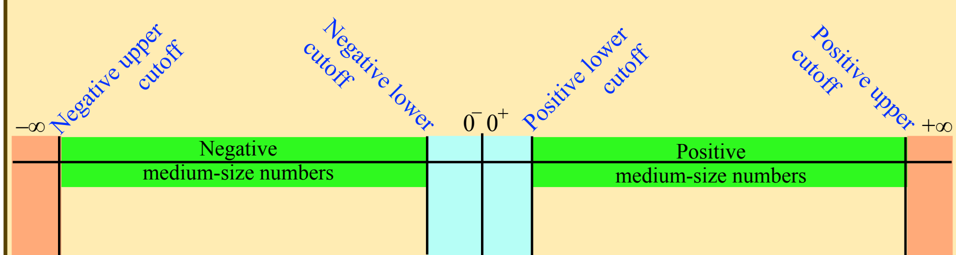
large variable
L
medium-size
appreciable number
sizable number

Definition 0.11 The large variables $\boldsymbol{L}$, $\boldsymbol{M}$, ... will be the (non-standard) symbols for generic *large-size* numbers.

Cautionary note 0.11 because ∞ is *not* a number to begin with. (Cautionary Note 0.3 - ∞ is *not* a *number* (Page 6))

iii. The numbers whose size is just right for us to *give*:

Definition 0.12 **Medium-size** numbers are numbers that are *both* smaller-size than the *upper* cutoff size *and* larger-size than the *lower* cutoff size



(https://www.macmillandictionary.com/thesaurus-category/british/not-big-or-small-size)

Appreciable numbers *or* **sizable numbers** *might have been better words.*

Theorem 0.4 Mid-size numbers are *non-zero* numbers. (But non-zero numbers are *not necessarily* medium-size numbers.)

Proof. Acording to **??** **??** - **??** (**??**) and as the represent illustrates, ordinary numbers

- The upper cutoff size keeps medium-size numbers away from $-\infty$ and $+\infty$.
- The lower cutoff size keeps medium-size numbers away from 0^- and 0^+.

□

=======Begin WORK ZONE=======

Ordinary numbers are medium-size real world numbers

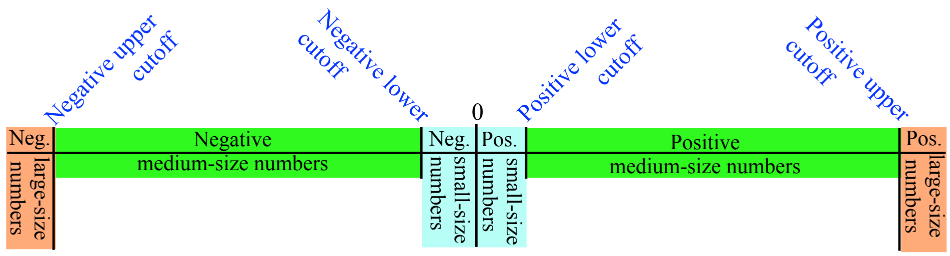
> **AGREEMENT 3.1 (Restated) Interpolation** will be short for ordinary number.

In view of **?? ??** - **??** (**??**), both $+1$ and -1 are medium-size.

========End WORK ZONE=======

While the variables x, y, z can stand for numbers of *any* qualitative sizes,

Altogether, then, these qualitative sizes are illustrated by:



7. About the language.

- We all have an *intuitive* idea of what the everyday words *large*, *small* and *medium* mean and these words have the same meaning for everybody even though *large*, *small* and *medium* are relative concepts.

Of course, in some countries, a dollar an hour is actually a large *amount of money.*

EXAMPLE 0.53. Nobody likes to work for a *small* amount of money: billionaires would no more dream of working for, say, a thousand dollars an hour than the rest of us would like to work for a dollar an hour.

Veery, veery carefully!

However, we needed to define *large*, *small* and *medium* as calculus words so we had to proceed carefully.

- Here are a few dictionary definitions of *large*:

> "*bigger than usual in size*". (https://www.macmillandictionary.com/dictionary/british/large_1)
> "*exceeding most other things of like kind especially in quantity or size*"

finite
infinite
infinitesimal

(https://www.merriam-webster.com/dictionary/large)
"*Of greater than average size*" (https://www.thefreedictionary.com/Large)
"*of more than average size*" (https://www.dictionary.com/browse/large)
"*greater in size than usual or average*" (https://www.collinsdictionary.com/dictionary/english/large)

Notice that *all* these dictionary definitions use, essentially, *larger-size* and that they also use "*than most other*", "*than average*", "*than usual*" as some sort of upper cutoff size.

- The words *large* and *small*—even though it is very tempting to use them, if only as shorts for large-size and small-size—are too close to the everyday words *larger* and *smaller* which are used to compare *plain* numbers in the everyday language while in *any* mathematical language *larger* and *smaller* are used to compare *signed* numbers.
- The meanings of the words medium-size, small-size, and large-size, are very close to the meanings of

Language note 0.6 The mathematical words

- **Finite** (For medium-size) (https://en.wikipedia.org/wiki/Finite_number),
- **Infinitesimal** (For small-size) (https://en.wikipedia.org/wiki/Infinitesimal),
- **Infinite** (For large-size) (https://www.dictionary.com/browse/infinite),

But since *mathematicians* understand the words finite, infinitesimal, and infinite much more strictly than we would, we will stay with the words medium-size, small-size, and large-size.

9 Computing with Qualitative Sizes

REWRITE ALL THIS SECTION USING h and L

=======Begin WORK ZONE=======

While 0 does not exist in the real world, small-size numbers do exist in the real world

h^n

So, while $5 \odiv 0$ does not exist in the real world so that we do not want to write $5 \oplus 0 = \infty$, small-size number does exist in the real world and there is no problem writing $5 \oplus h = L$ /Users/alainschremmer/Desktop/untitled folder small-size number $\oplus$ small-size number

=========End WORK ZONE========

For *calculating* purposes, qualitative sizes make up a rather crude system because qualitative sizes carry no information whatsoever about *where* the cutoffs are.

And if you're worried about rigor, you'll be glad to know qualitative sizes *lead straight to Bachmann-Landau's little o's and big O's (https://en.wikipedia.org/wiki/Big_O_notation).*

Nevertheless, as we will see, the calculations we *can* do with qualitative sizes will be plenty enough to help us simplify calculations by separating what is qualitatively the right size to be relevant to the point we are interested in from what is qualitatively the wrong size and therefore irrelevant to that point.

We will now discuss to what extent we can calculate with *numbers* of which all we know is their qualitative size: large-size, or small-size, or medium-size.

You don't need extreme *cutoff sizes but do pick your numbers* far *from the cutoffs.*

In each case, it is most important that you develop a good feeling for what is happening and so it is important for you to experiment by setting cutoff sizes and then picking numbers with the qualitative sizes you want. A good rule of thumb for picking:

- medium-size numbers is to try ± 1,
- large-size numbers is to try ± 10.0 or ± 100.0 or ± 1000.0 etc
- small-size numbers is to try ± 0.1 or ± 0.01 or ± 0.001 etc

1. Adding and subtracting qualitative sizes.

THEOREM 0.5 Oplussing qualitative sizes numbers

$\oplus$	large-size	medium-size	small-size
large-size	?	large-size	large-size
medium-size	large-size	?	medium-size
small-size	large-size	medium-size	small-size

Proof. **i.** The non-highlighted entries are as might be expected.

EXAMPLE 0.54.

$$-100\,000 \oplus +1\,000 = -99\,000$$
$$-100\,000 \oplus -0.001 = 100\,000.001$$

So, the reader is invited to decide on cutoff sizes, experiment a bit, and then prove the non-highlighted entries using these cutoff sizes.

ii. When the two large-size numbers have opposite signs, the addition is **undetermined** because the result could then be large-size, or small-size, or medium-size, depending on "how much" large-size the two large-size numbers are compared to each other. undetermined

EXAMPLE 0.55. Here are two additions of large-size numbers whose results are different in *qualitative sizes*:
$+1\,000\,000\,000\,000.7 \oplus -1\,000\,000\,000.4 = +999\,000\,000\,000.3$,
but
$-1\,000\,000\,000\,000.5 \oplus +1\,000\,000\,000\,000.2 = -0.3$.

□

=======Begin WORK ZONE=======

Since $\ominus = \oplus$ Opposite

=======End WORK ZONE========

2. Multiplying qualitative sizes.

THEOREM 0.6 Otiming qualitative sizes

$\odot$	large-size	medium-size	small-size
large-size	large-size	large-size	?
medium-size	large-size	medium-size	small-size
small-size	?	small-size	small-size

The generic symbols have different subscripts because, even when they have the same qualitative size, they stand for different numbers.

Proof. **i.** The non-highlighted entries are as might be expected.

EXAMPLE 0.56. $-10\,000 \odot -1\,000 = +10\,000\,000$
$+0.01 \odot -0.001 = -0.00001$

So, the reader is invited to decide on cutoff sizes, experiment a bit, and then prove the non-highlighted entries using these cutoff sizes.

ii. large-size $\odot$ small-size is undetermined because the result could be large-size, or small-size, or medium-size, depending on "how much large-size" large-size is compared to "how much small-size" small-size is.

EXAMPLE 0.57. Here are different instances of large-size $\odot$ small-size that result in different *qualitative sizes*:

$-1\,000 \odot -0.1 = +100$	$-100\,000\,000 \odot -0.00\,001 = +100$
$+1\,000 \odot -0.001 = -1$	$+1\,000\,000 \odot -0.00\,001 = -1$
$+1\,000 \odot +0.00\,001 = +0.01$	$+1\,000 \odot +0.00\,001 = +0.01$

Similarly for small-size $\odot$ large-size.

□

3. Dividing qualitative sizes.

The generic symbols have different subscripts because, even when they have the same qualitative size, they stand for different numbers.

THEOREM 0.7 ⊙dividing qualitative sizes

—	large-size	medium-size	small-size
large-size	?	large-size	large-size
medium-size	small-size	medium-size	large-size
small-size	small-size	small-size	?

Proof. **i.** The non-highlighted entries are as might be expected.

EXAMPLE 0.58. $\frac{-10\,000\,000}{+50} = -200\,000$
$\frac{+0.03}{+6\,000\,000} = +0.000\,000\,005$

So, the reader is invited to decide on cutoff sizes, experiment a bit, and then prove the non-highlighted entries using these cutoff sizes.

ii. $\frac{\text{large-size}}{\text{large-size}}$ is undetermined because the result could be large-size, or small-size, or medium-size, depending on "how much large-size" large-size and large-size are compared to each other..

EXAMPLE 0.59. Here are three instances of $\frac{\text{large-size}}{\text{large-size}}$ that result in different *qualitative sizes*:
$\frac{-1\,000\,000}{-1\,000} = +1\,000$, $\frac{-1\,000\,000}{-100\,000} = -10$, $\frac{-100\,000}{-1\,000\,000\,000} = +0.000\,1$.

And $\frac{\text{small-size}}{\text{small-size}}$ is similarly undetermined.

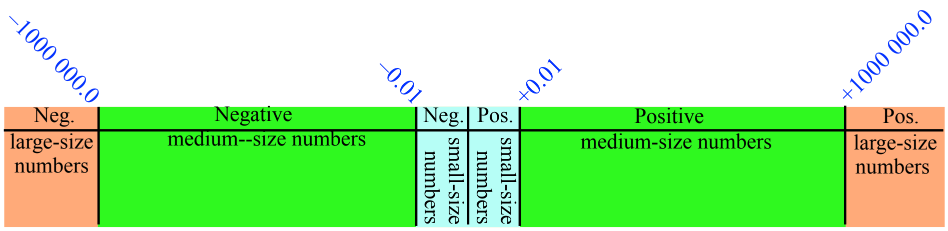
EXAMPLE 0.60. Here are three instances of small-size $\odot$ small-size that result in different *qualitative size*:
$-0.001 \odot +0.1 = -0.01$, $+0.001 \odot +0.001 = +1$, $-0.01 \odot -0.001 = +10$

□

4. Reciprocal of a qualitative size. We really would like the reciprocal of a small-size number to be a large-size number and, the other way round, the reciprocal of a large-size number to be a small-size number.

i. Unfortunately, because we defined qualitative sizes in terms of cutoff sizes which we set independently of each other, this is *not necessarily* the case and the reciprocal of a small-size number need *not* be a large-size number and, the other way round, the reciprocal of a large-size number need *not* be a small-size number because the upper cutoff size and the lower cutoff size are *not necessarily* reciprocal of each other.

EXAMPLE 0.61. The following cutoff sizes are probably suitable for the accounting system of a small business:



i. $+0.009$ is below the positive lower cutoff ($+0.009 < +0.01 = +0.010$) and is therefore a small-size number,
ii. The reciprocal of $+0.009$ is $+111.1$ (Use a calculator.)
iii. $+111.1$ is below the positive upper cutoff and is therefore *not* a large-size number.

ii. Fortunately, it is always possible to take the cutoff sizes so that

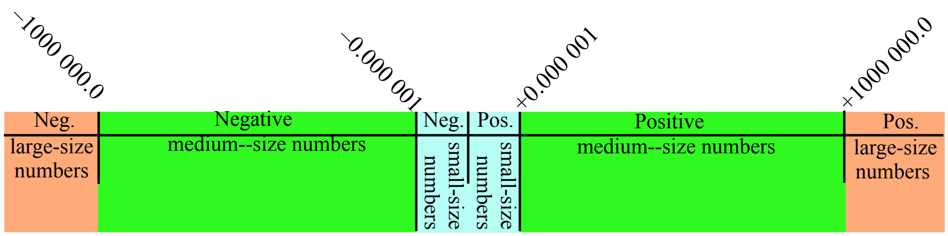
- the upper cutoff size *is* the reciprocal of the lower cutoff size

and, the other way round,

- the lower cutoff size *is* the reciprocal of the upper cutoff size

because all that will happen is that with the adjusted cutoff sizes there will now be more numbers that will be medium-size than is really needed.

EXAMPLE 0.62. We can change the lower cutoff size in **??** (**??**) to $0.000\,001$:



so that now the lower cutoffs and the upper cutoffs are reciprocal of each other:
i. $+0.000\,9$ is below the positive lower cutoff ($+0.0009 < +0.001 = +0.0010$) and is therefore a small-size number,
ii. The reciprocal of $+0.0009$ is $+1\,111.1$ (Use a calculator.)

iii. $+1\,111.1$ is above the positive upper cutoff and is therefore a large-size number.

The price is just thatnumbers whose size is between 0.01 and 0.000001 will now also be medium-size—but most probably will never be used.

iii. So then, from now on,

> **AGREEMENT 0.6** The lower cutoff size and the upper cutoff size will be reciprocal of each other.

iv. We then have:

> **THEOREM 0.8 Reciprocity of qualitative sizes**
>
> ▶ Reciprocal of large-size number $= \dfrac{+1}{\text{large-size number}}$
> $= \text{small-size number}$
>
> ▶ Reciprocal of small-size number $= \dfrac{+1}{\text{small-size number}}$
> $= \text{large-size number}$
>
> ▶ Reciprocal of medium-size number $= \dfrac{+1}{\text{medium-size number}}$
> $= \text{medium-size number}$

Proof.

- If a given number is large-size,
  - By DEFINITION 0.5 - Real world numbers (Page 17), the given number is larger-size than the upper cutoff size
  - By THEOREM 0.1 - Opposite numbers add to 0: (Page 20), the reciprocal of the given number is then smaller-size than the reciprocal of the upper cutoff size.
  - But by AGREEMENT 0.5 - (Page 28), the reciprocal of the upper cutoff size is the lower cutoff size.
  - So, the reciprocal of the given number is smaller-size than the lower cutoff size.
  - And so, by DEFINITION 0.5 - Real world numbers (Page 17), the reciprocal of the given large-size number is a small-size number
- The reader is invited to make the case for the reciprocal of a small-size given.
- The reader is invited to make the case for the reciprocal of a medium-sizegiven number that is medum-size

□ x_+ y_+ z_+ $x_{<0}$ $y_{<0}$ $z_{<0}$ restricted

A star next to a Section Number *is the standard way to say you can skip the section. But, eventually, ...*

10* Computing with Extended Numbers.

As it happens, we will *not compute* with extended numbers so this section can be safely skipped. On the other hand, it is interesting to see how it goes.

1. Positive and negative variables.

To denote the operations for extended numbers, we need a few more kinds of variables:

DEFINITION 0.13 are **resticted** variables:

- For *positive* variables, only *positive* numbers can be substituted, and
- For *negative* variables, only *negative* numbers can be substituted,

CAUTIONARY NOTE 0.12

▶ $x_{<0}$, $y_{<0}$, $z_{<0}$, $x_{>0}$, $y_{>0}$, $z_{>0}$ are restricted *variables*,

while

▶ $x<0$, $y<0$, $z<0$. $x>0$, $y>0$, $z>0$, are *inequations* involving the global variables x, y, z.

2. Operation tables.

$\oplus$	$-\infty$	$y_{<0}$	0^-	0^+	$y_{>0}$	$+\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	?
$x_{<0}$	$-\infty$	z	$x_{<0}$	$x_{<0}$	$z_?$	$+\infty$
0^-	$-\infty$	$z_{<0}$	0^-	$0^?$	$z_{>0}$	$+\infty$
0^+	$-\infty$	$z_{<0}$	$0^?$	0^+	0^+	$+\infty$
$x_{>0}$	$-\infty$	$z_?$	$x_{>0}$	$x_{>0}$	$z_{>0}$	$+\infty$
$+\infty$	?	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$

$\ominus$	$-\infty$	$y_{<0}$	0^-	0^+	$y_{>0}$	$+\infty$
$-\infty$	?	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$x_{<0}$	$+\infty$	$z_?$	$x_{<0}$	$x_{<0}$	$z_{<0}$	$-\infty$
0^-	$+\infty$	$y_{>0}$	$0^?$	0^-	$y_{<0}$	$-\infty$
0^+	$+\infty$	$y_{>0}$	0^+	$0^?$	$y_{<0}$	$-\infty$
$x_{>0}$	$+\infty$	$z_{>0}$	$x_{>0}$	$x_{>0}$	$z_?$	$-\infty$
$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$+\infty$	?

$\odot$	$-\infty$	$y_{<0}$	0^-	0^+	$y_{>0}$	$+\infty$
$-\infty$	$+\infty$	$+\infty$	$+?$	$-?$	$-\infty$	$-\infty$
$x_{<0}$	$+\infty$	$z_{>0}$	0^+	0^-	$z_{<0}$	$-\infty$
0^-	$+?$	0^+	0^+	0^-	0^-	$-?$
0^+	$-?$	0^-	0^-	0^+	0^+	$+?$
$x_{>0}$	$-\infty$	$z_{<0}$	0^-	0^+	$z_{>0}$	$+\infty$
$+\infty$	$-\infty$	$-\infty$	$-?$	$+?$	$+\infty$	$+\infty$

$\oplus\!\!\!\!\div$	$-\infty$	$y_{<0}$	0^-	0^+	$y_{>0}$	$+\infty$
$-\infty$	$+?$	$+\infty$	$+\infty$	$-\infty$	$-\infty$	$-?$
$x_{<0}$	$0_{>0}$	$z_{>0}$	$+\infty$	$-\infty$	$z_{<0}$	$0_{<0}$
0^-	0^+	0^+	$+?$	$-?$	0^-	0^-
0^+	0^-	0^-	$-?$	$+?$	0^+	0^+
$x_{>0}$	0^-	$z_{<0}$	$-\infty$	$+\infty$	$z_{>0}$	0^+
$+\infty$	$-?$	$-\infty$	$-\infty$	$+\infty$	$+\infty$	$+?$

One reason we will *not* compute with extended numbers is of course the yellow boxes in the above operation tables.

3. Are ∞ and 0 reciprocal? Another reason for *not* computing with extended numbers is that,

- From the *division* table, we get that $\dfrac{x_{>0}}{-\infty} = 0^-$ and therefore, in particular, that $\dfrac{+1}{-\infty} = 0^-$ so that, as would be expected, the reciprocal of $-\infty$ is 0^- and, similarly, we get that the reciprocal of $+\infty$ is 0^+,
- However, from the *multiplication* table we get only that $-\infty \odot 0^- = +?$ and that $+\infty \odot 0^+ = +?$

While not contradictory, this would be annoying and, as we will see in THEOREM 0.6 - Otiming qualitative sizes (Page 35), we will have a much more satisfying way to compute whether or not 0 and ∞ are reciprocal.

11 Neighborhoods

As we saw in Size of an amount of stuff (Subsection 4.4, Page 15), while

- ▸ We certainly cannot evaluate generic expressions *at* ∞ because ∞ is *not* a *number* (CAUTIONARY NOTE 0.3, Page 6),

more generally,

- ▸ We cannot even always evaluate generic expressions *at* a given number x_0 because of the difficulties with 0 and division: 0 is a *dangerous* number (CAUTIONARY NOTE 0.2, Page 5)

▶

nearby number
near 0
near ∞

1. Nearby numbers. Evaluating a generic expression *at* a point, though, is to ignore the real world and, in fact, since, as we will see in Subsection 3.4 - Sparseness of sets of plot dots (Page 80), CALCULUS deals with 'change', instead of wanting to investigate what happens *at* a given point, we will investigate what happens *at* **nearby numbers**.

In a crime novel, the victim is never the story. The story is always around *the victim. (Anonymous crime writer.)*

EXAMPLE 0.63. As opposed to EXAMPLE 0.19 (Page 13), we can tell a car is moving from a *movie*, that is from still pictures during a short time span.

More precisely:

i. As we saw in Section 2 - Zero and Infinity (Page 4), *nothingness* does not exist in the real world,

EXAMPLE 0.64. We use 0 quart of milk to denote the amount of milk that appears to be in an empty bottle but it might just be that the amount of milk in the bottle is just too small for us to see.

Just how clean is clean?

So, in accordance with the real world, we will use nearby numbers that is, in this case, numbers **near** 0, that is *small-size* numbers,

EXAMPLE 0.65. $-0.002.078$ and $+0.000.928$ are both near 0.

ii. As we saw in Subsection 0.3 - (Page xxii), *infinity* does not exist in the real world,

EXAMPLE 0.66. We may say that the number of molecules in a spoonful of milk is infinite, but of course it's just that the number of molecules is too large for us to count under a microscope.

So, in accordance with the real world, we will use nearby numbers, that is, in this case, numbers **near** ∞, that is *large-size* numbers,

EXAMPLE 0.67. $-12\,729\,000\,307$ and $+647\,809\,010\,374$ are both near ∞

iii. As we saw in **??** **??** - **??** (**??**), measured numbers will always differ from a given number x_0 by some error

EXAMPLE 0.68. I can give you 3 apples but I cannot give you a 3 foot long stick as it will always be a bit too long or a bit too short.

So, in accordance with the real world, we will use nearby numbers that is, in this case, numbers **near** x_0, that is numbers that *differ* from x_0 by only *small-size* numbers.

neighborhood
thicken
center
indeterminate number
circa variable

EXAMPLE 0.69. $-87.36 \oplus -0.000.032 = -87.360\,032$ and $-87.36 \oplus +0.000.164 = -87.359\,836$ are both near -87.36

Actually, it is completely standard to speak of a

> **DEFINITION 0.14 Neighborhood** of a point:
> - A neighborhood of 0 consists of the numbers *near* 0.
> - A neighborhood of ∞ consists of the numbers *near* ∞,
> - A neighborhood of x_0 consists of the numbers *near* x_0.
>
> (https://en.wikipedia.org/wiki/Neighbourhood_(mathematics))

And, in fact, we will often speak of **thickening** a given point, that is we will be looking at that point as just the **center** of a neighborhood of that point.

2. Evaluation *near* a given point. So, in order to evaluate a generic expression *near* a given point, we will evaluate the generic expression *at* an **indeterminate number** *near* the given point. In other words:

- Instead of declaring 0, we will declare the small variable h,
- Instead of declaring ∞, we will declare the large variable L,
- Instead of declaring x_0, we will declare:

> **DEFINITION 0.15** The **circa variables** $\boldsymbol{x_0 \oplus h}$, $\boldsymbol{x_0 \oplus k}$ are the (standard) symbols for numbers near x_0.
>
> "Circa" because the numbers are "around" h, "variable" because of h.

Why "circa"? Because nearby is already used.

In other words, we will use PROCEDURE 0.1 - Evaluate a generic expression *at* a given number (Page 10) but with an *indeterminate* number instead of a *given* number.

PROCEDURE 0.3 To evaluate a given generic expression in terms of x *near* a given point:

i. Declare an indeterminate numbers *near* the given point, that is:

- If the given point is 0, declare the small variable h by writing the declaration $\Big|_{x \leftarrow h}$, read "x *to be* replaced by h", to the right of the generic expression:

$$\text{generic expression in terms of } x \Big|_{x \leftarrow h}$$

- If the given point is ∞, declare the large variable L by writing the declaration $\Big|_{x \leftarrow L}$, read "x *to be* replaced by L", to the right of the generic expression:

$$\text{generic expression in terms of } x \Big|_{x \leftarrow L}$$

- If the given point is x_0, declare the local variable $x_0 \oplus h$ by writing the declaration $\Big|_{x \leftarrow x_0 \oplus h}$, read "$x$ *to be* replaced by $x_0 \oplus h$", to the right of the generic expression:

$$\text{generic expression in terms of } x \Big|_{x \leftarrow x_0 \oplus h}$$

ii. Replace every occurence of x in the generic expression in terms of x by the declared variable to get the generic expression for numbers near the given point :

- generic expression in terms of h for numbers near 0
- generic expression in terms of L for numbers near ∞
- generic expression in terms of $x_0 \oplus h$ for numbers near x_0

iii. Execute the general expression in terms of the declared variable according to the relevant rules in Section 9 - Computing with Qualitative Sizes (Page 33)

In contradistinction with Demo 0.1a - From x to $+5$ (Page 11), we have:

Demo 0.3a To evaluate the generic expression $\dfrac{x^{+2} \ominus +7}{x \oplus +3}$ *near* $+5$

i. We declare that the numbers are to be near $+5$ by writing the declaration $\Big|_{x \leftarrow +5 \oplus h}$, read "$x$ *to be* replaced by $+5 \oplus h$", to the right of the generic expression:

$$\left.\frac{x^{+2} \ominus +7}{x \oplus +3}\right|_{x \leftarrow +5 \oplus h}$$

ii. We replace every occurence of x in the generic expression in terms of x by the local variable $+5 \oplus h$ to get the generic expression for numbers near $+5$:

$$\frac{+5 \oplus h^{+2} \ominus +7}{+5 \oplus h \oplus +3}$$

iii. We execute the generic expression in terms of $+5 \oplus h$:

$$\frac{+25 \oplus +10h \oplus +h^2 \ominus +7}{+5 \oplus +h \oplus +3}$$

$$\frac{+18 \oplus +10h \oplus +h^2}{+8 \oplus +h}$$

Since the division probably won't stop by itself and since where *we* will stop the division will depend on the information we will want, the last expression just above is not an executed expression.

In contradistinction with DEMO 0.1b - From x to -3 (Page 12), we have:

DEMO 0.3b To evaluate the generic expression $\dfrac{x^{+2} \ominus +7}{x \oplus +3}$ *near* -3

i. We declare that the numbers are to be near -3 by writing the declaration $\Big|_{x \leftarrow -3 \oplus h}$, read "$x$ *to be* replaced by $-3 \oplus h$", to the right of the generic expression:

$$\left.\frac{x^{+2} \ominus +7}{x \oplus +3}\right|_{x \leftarrow -3 \oplus h}$$

ii. We replace every occurence of x in the generic expression in terms of x by the local variable $-3 \oplus h$ to get the generic expression for numbers near -3:

$$\frac{\left(-3 \oplus h\right)^{+2} \ominus +7}{-3 \oplus h \oplus +3}$$

iii. We execute the generic expression in terms of $-3 \oplus h$:

$$\frac{+9 \oplus -6h \oplus h^2 \ominus +7}{-3 \oplus +3 \oplus h}$$

$$\frac{+2 \oplus -6h \oplus h^2}{h}$$

$$+2h^{-1} \oplus -6 \oplus h$$

Since the division was by h, the last expression just above is an executed expression.

In contradistinction with DEMO 0.1c - From x to $+3$ (Page 12), we have:

DEMO 0.3c To evaluate the generic expression $\dfrac{x^{+2} \ominus +9}{x \oplus -3}$ *near* $+3$

i. We declare that the numbers are to be near $+3$ by writing the declaration $\Big|_{x \leftarrow +3 \oplus h}$, read "$x$ *to be* replaced by $+3 \oplus h$", to the right of the generic expression:

$$\left.\frac{x^{+2} \ominus +9}{x \oplus -3}\right|_{x \leftarrow +3 \oplus h}$$

ii. We replace every occurence of x in the generic expression in terms of x by the local variable $+3 \oplus h$ to get the generic expression for numbers near $+3$:

$$\frac{\left(+3 \oplus h\right)^{+2} \ominus +9}{+3 \oplus h \oplus -3}$$

iii. We execute the generic expression in terms of $+3 \oplus h$:

$$\frac{+9 \oplus +6h \oplus h^2 \ominus +9}{+3 \oplus -3 \oplus h}$$

$$\frac{+6h \oplus h^2}{h}$$

$$+6 \oplus h$$

Note that, here, the division being by h, we just did it and the expression just above is an executed expression.

And here is how it goes near ∞:

DEMO 0.3d To evaluate the generic expression $\dfrac{x^{+2} \ominus +9}{x \oplus -3}$ *near* ∞

i. We declare that the numbers are to be near ∞ by writing the declaration $\Big|_{x \leftarrow L}$, read "x *to be* replaced by L", to the right of the generic expression:

$$\left.\frac{x^{+2} \ominus +9}{x \oplus -3}\right|_{x \leftarrow L}$$

ii. We replace every occurence of x in the generic expression in terms of x by the local variable L to get the generic expression for numbers near ∞:

$$\frac{L^{+2} \ominus +9}{L \oplus -3}$$

iii. We execute the generic expression in terms of L:

$$\frac{L^2 \ominus +9}{L \ominus -3}$$

$$\frac{L^2 \ominus [...]}{L \ominus [...]}$$

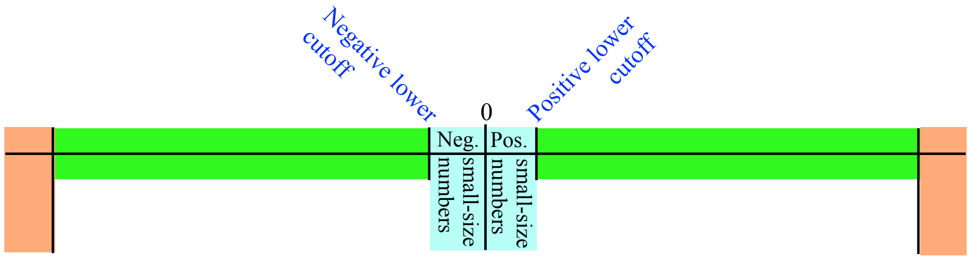
$$\frac{L^2}{L} \oplus [...]$$

magnifier

$L \oplus [...]$

The last expression just above is the executed expression.

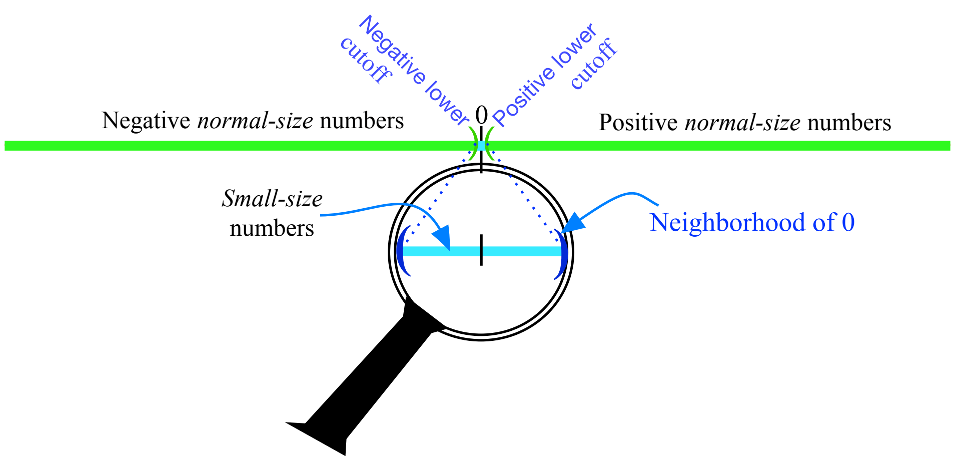
3. Picturing a neighborhood of 0. In DEFINITION 0.8 - Small-size numbers (Page 30), small-size numbers were pictured with



which is *not* really a representation because the three qualitative sizes are represented at different scales. (https://en.wikipedia.org/wiki/Scale_(represent)#Large_scale,_medium_scale,_small_scale).

i. On a quantitative ruler, at just about any scale (https://en.wikipedia.org/wiki/Scale_(represent)#Large_scale,_medium_scale,_small_scale), the negative lower cutoff for medium-size numbers and the positive lower cutoff for medium-size numbers will both be on top of 0 and we won't be able to see small-size numbers.

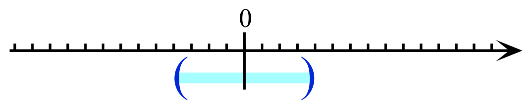
So, in order to see a neighborhood of 0, we would need some kind of **magnifier**:



The fact though, that, the neighborhood needs to be representd at a scale larger than the scale of the quantitative ruler creates a problem. One way

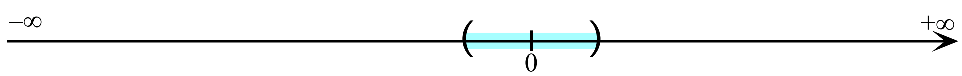
qualitative ruler
compactor

out, of course, would be to draw the neighborhood of 0 just *under* the quantitative ruler:

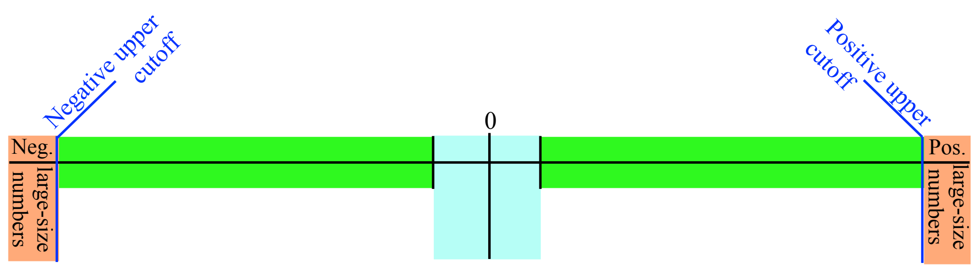


ii. But on a **qualitative ruler**, that is on a ruler *without* scale therefore without tickmarks—not even for 0— but with $-\infty$ and $+\infty$ as end of the line symbols in accordance with AGREEMENT 0.3 - Sides of the origin (Page 18):

we can draw a neighborhood of 0 as



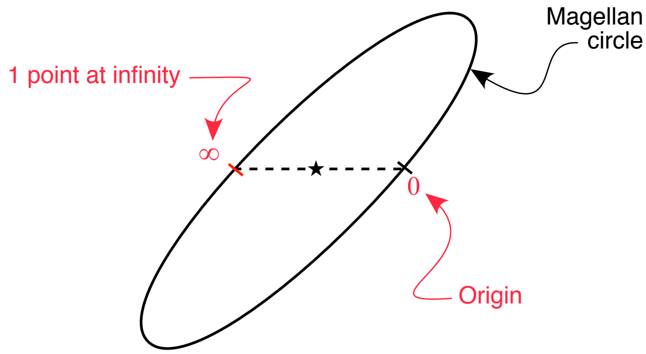
4. Picturing a neighborhood of ∞. In DEFINITION 0.10 - Large-size numbers (Page 30) large-size numbers were pictured with



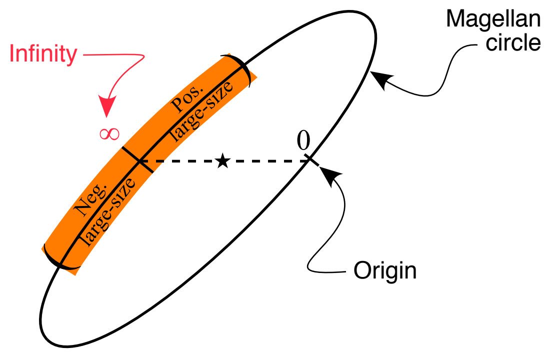
which, again, is *not* a representation because the three qualitative sizes are representd at different scales. (https://en.wikipedia.org/wiki/Scale_(represent)#Large_scale,_medium_scale,_small_scale)

i. On a *quantitative* ruler, at just about any scale, the negative *upper* cutoff for medium-size numbers and the positive *upper* cutoff for medium-size numbers will both be way off the represent so we would need some kind of **compactor**.

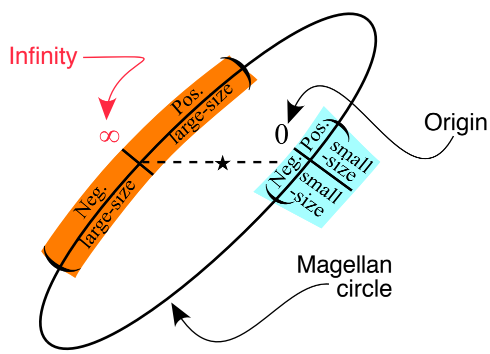
ii. In the spirit of one-point compactification, using a Magellan circle



on which large-size numbers are representped as



the advantage is that positive large-size numbers and negative large-size numbers are representped right next to each other the same way as positive small-size numbers and negative small-size numbers:



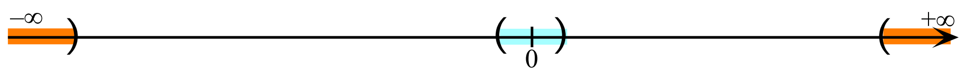
which represents large-size numbers as a neighborhood of ∞ just the way small-size numbers make up a neighborhood of 0. *Nicely!*

iii. In the spirit of two-points compactification, we can also represent a neighborhood of ∞, that is large-size numbers, on a qualitative ruler as:

Mercator

And, after all, 0 *is the center of our neighborhood.*

Here, the advantage is that we are still facing 0 but the disadvantage is, as opposed to the Magellan represent, that positive large-size numbers and negative large-size numbers are separated from each other, the opposed way of positive small-size numbers and negative small-size numbers which are right next to each other:

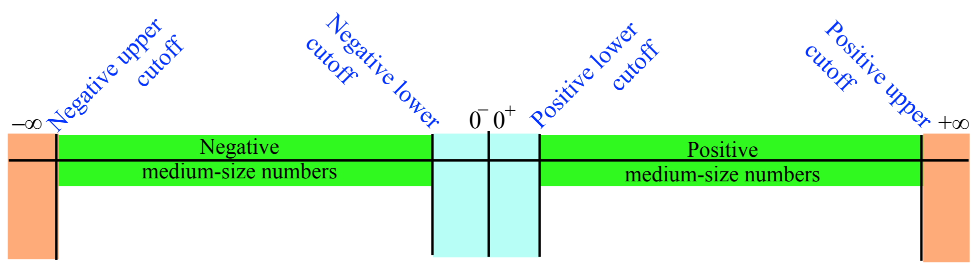


This is often referred to as a **Mercator** represent. (`https://en.wikipedia.org/wiki/Mercator_projection`)

iv.

========End WORK ZONE=======

5. Picturing a neighborhood of x_0. In DEFINITION 0.12 - Medium-size numbers (Page 31) medium-sized numbers were pictured wirh



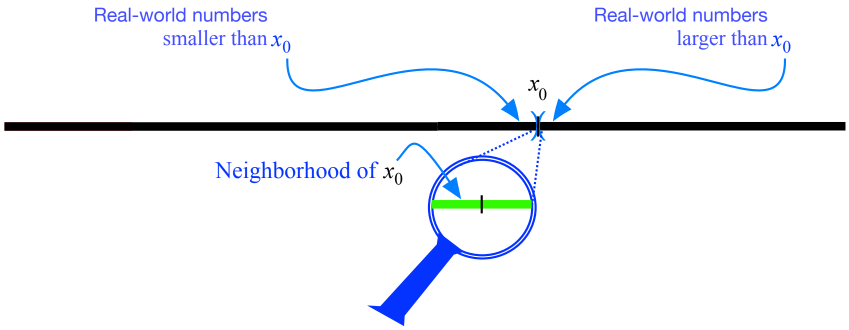
which, again, is *not* a represent because the three qualitative sizes are representd at different scales. (`https://en.wikipedia.org/wiki/Scale_(represent)#Large_scale,_medium_scale,_small_scale`)

The situation with a neighborhood of x_0 is similar to the situation with a neighborhood of 0:

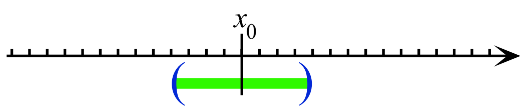
i. On a quantitative ruler, at just about any scale (`https://en.wikipedia.org/wiki/Scale_(represent)#Large_scale,_medium_scale,_small_scale`), the medium-size numbers smaller than x_0 and the medium-size numbers larger than x_0 leave no room between them and we won't be able to see the numbers near x_0

So, in order to see a neighborhood of x_0, that is numbers near x_0, that isnumbers that differ from x_0 by only small-size numbers, we would need to aim a magnifier at x_0, the center of the neighborhood.

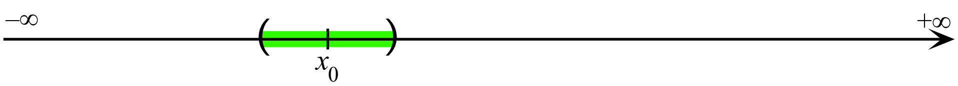
side-neighborhoods
left-neighborhood
right-neighborhood
0^+
right
0^-
left



Again, the fact that a neighborhood needs to be representd at a scale larger than the scale of the quantitative ruler creates a problem. And again, a way out would be to represent the neighborhood of x_0 just *under* the quantitative ruler:



ii. But on a *qualitative* ruler we can represent a neighborhood of x_0 as



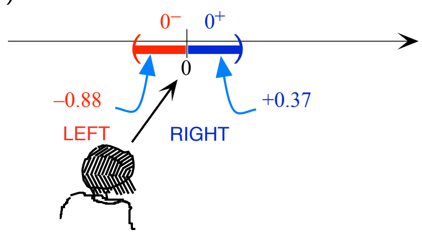
6. Side-neighborhoods. In order to deal *separately* with each side of a neighborhood we will often have to distinguish the **side-neighborhoods**. Pinning down the **left-neighborhood** from the **right-neighborhood**, though, depends on the nature of the point:

- – A left-neighborhood of 0 consists of the *negative* numbers *near* 0 (*negative* small-size numbers),
  – A right-neighborhood of 0 consists of the *positive* numbers *near* 0 (*positive* small-size numbers),

  In order to deal *separately* with each side of a neighborhood of 0, we will use the symbols
  - $\mathbf{0^+}$ (namely 0 with a little + up and to the right) which is *standard* expression for positive small-size numbers.
    Positive small-size numbers are **right of** of 0, that is they are to *our right* when *we* are facing 0, the center of the neighborhood.
  - $\mathbf{0^-}$ (namely 0 with a little – up and to the right) which is *standard* expression for negative small-size numbers.
    Negative small-size numbers are **left** of 0, that is they are to *our left* when *we* are facing 0, the center of the neighborhood.

$+\infty$
$-\infty$

EXAMPLE 0.70. 0^{+} refers tosmall-sizenumbers right of 0 (such as for instance $+0.37$) and 0^{-} refers tosmall-sizenumbers left of 0 (such as for instance -0.88):



So, never forget that

> **CAUTIONARY NOTE 0.13** $^{+}$ or $^{-}$ up to the right and *by itself* is *not* an 'exponent' but indicates which *side* of 0.

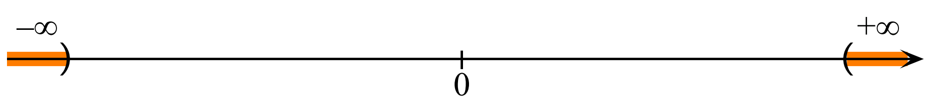
- – A left-neighborhood of ∞ consists of the *positive* numbers *near* ∞ (*positive*large-size numbers),
  – A right-neighborhood of ∞ consists of the *negative* numbers *near* ∞ (*negative* large-size numbers),

  Just as we will often have to refer separately to each side of a neighborhood of 0, we will often have to refer separately to each side of a neighborhood of ∞

  So we will use:
  - $+\infty$ as symbol for *positive* large-size numbers,
  - $-\infty$ as symbol for *negative* large-size numbers,

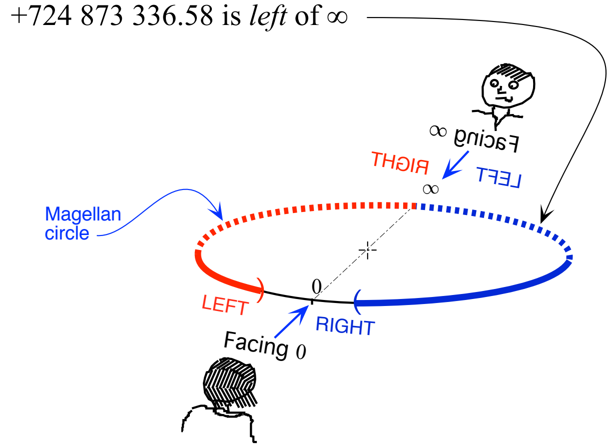
  even though

  We will then use as qualitative ruler:



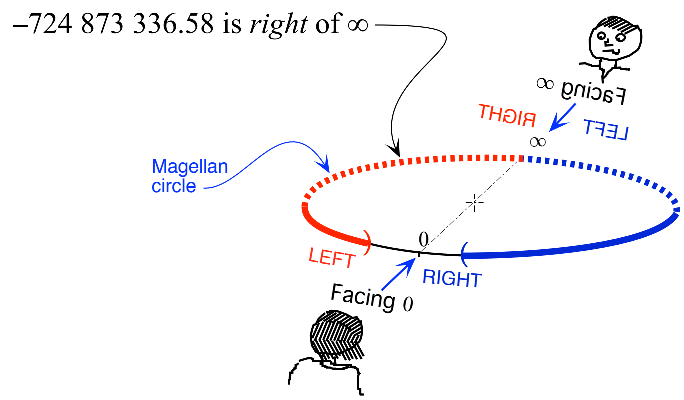
- Keep in mind that it is easy to forget which side is left of ∞ and which side is right of ∞ because it is easy to forget that one must *face* the center of the neighborhood, namely ∞:
  - *Positive* large-size numbers are *left* of ∞ because, to face the center of the neighborhood, we have to imagine ourselves facing ∞, and then *positive* numbers will be to *our* left.

EXAMPLE 0.71.



- *negative* large-size numbers are *right* of ∞ because, to face the center of the neighborhood, we have to imagine ourselves facing ∞, then *negative* numbers would be to *our* right.

EXAMPLE 0.72.



- – A left-neighborhood of x_0 consists of the numbers *near* x_0 that are smaller than x_0,(medium-size numbers that differ from x_0 by only small-size numbers).
  – A right-neighborhood of x_0 consists of the numbers *near* x_0 that are larger than x_0,

7. Interplay between 0 and ∞. As already mentioned in Section 3 - Numbers In General (Page 7), both Numbers In General have intrigued people for a long time:

i. While, as mentioned in Section 3 - Numbers In General (Page 7), both 0 and ∞ are literally without meaning, both 0 and ∞ are absolutely and completely indispensable.

EXAMPLE 0.73. When we have eaten three apples out of five apples, we indicate that there are two apples left by writng:

$$5 \text{ apples} - 3 \text{ apples} = 2 \text{ apples}$$

But when we have eaten three apples out of three apples, how do we indicate that there is none left?

$$3 \text{ apples} - 3 \text{ apples} =? \text{ apples}$$

EXAMPLE 0.74. When we count "eight, nine, ten, eleven" we use a rhythm as indicated by the commas, say:

eight $1sec$ nine $1sec$ ten $1sec$ eleven

And in fact, when we start counting *with* "eight", we think we are counting *from* "seven" and precede "eight" with the same silence:

$1sec$ eight $1sec$ nine $1sec$ ten $1sec$ eleven

But *from* what number are we thinking we are starting *from* when we start counting *with* "one" and precede "one" by the same silence?

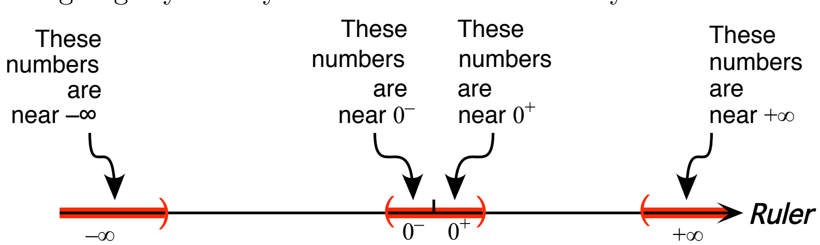
$1sec$ one $1sec$ two $1sec$ three $1sec$ fout

EXAMPLE 0.75. When we get impatient and want to stop counting, we probably end the counting with "etc"

EXAMPLE 0.76. When a number is so large that we cannot even begin to imagine it, we often use the word "infinite".

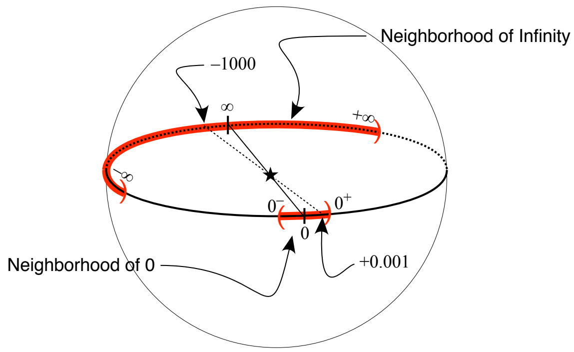
As in "The number of people who want to teach you is infinite."

ii. Even though, as an input, 0 is usually not particularly important, there is an intriguing "symmetry" between ∞ and 0 namely:



More precisley, *small numbers* are some sort of inverted image of *large numbers* since the *reciprocal* of a *large number* is a *small number* and vice versa.

EXAMPLE 0.77. The opposite of the reciprocal of -0.001 is $+1\,000$. In a Magellan view, we have real number



iii. Moreover, since by DEFINITION 0.12 - Medium-size numbers (Page 31), small-size numbers are near 0 and large-size numbers are near ∞, THEOREM 0.8 - Reciprocity of qualitative sizes (Page 38) can be restated as

> **THEOREM 0.8 (Restated) Reciprocity of qualitative sizes**
> - The reciprocal of a number near ∞ is a number near 0,
> - The reciprocal of a number near 0 is a number near ∞.

It then seems somewhat artificial, even though CAUTIONARY NOTE 0.2 - 0 is a *dangerous* number (Page 5) and CAUTIONARY NOTE 0.3 - ∞ is *not* a *number* (Page 6), not to extend the reciprocity of numbers near 0 (small-size numbers) and numbers near ∞ (large-size numbers) to a reciprocity of 0 and ∞ themselves. So,

> **AGREEMENT 0.7** Since we will *not* compute with ∞, this will only be a shorthand for THEOREM (Restated) 0.8 - Reciprocity of qualitative sizes (Page 55).

But what an extremely convenient shorthand!

12* Real Numbers

This section is only for those readers who want to know what are the **real numbers** used by most CALCULUS texts, and what using real numbers instead of real-world numbers would entail.

1. What *are* real numbers? Even though most college mathematics textbooks claim to *use* real numbers, the individualst they ever come to defining real numbers is something along the lines of "*a real number is a value of a continuous quantity that can represent a distance along a*

Which, one has to admit, isn't particularly enlightening. Moreover, the wording in Wikipedia keeps changing with time! A sign of unease?

fraction
root

line." (https://en.wikipedia.org/wiki/Real_number or https://math.vanderbilt.edu/schectex/courses/thereals/)

And of course, there is a very good reason for this vagueness (https://en.wikipedia.org/wiki/Vagueness_and_Degrees_of_Truth): in contrast with real-world numbers, real numbers are so *extremely* complicated to define that it is only done in REAL ANALYSIS, long after CALCULUS.

"*The real number system* $(\mathbb{R};+;\cdot;<)$ *can be defined axiomatically* [...] *There are also many ways to construct "the" real number system, for example, starting from whole numbers,* (https://en.wikipedia.org/wiki/Natural_number) *then defining* rational*numbers algebraically* (https://en.wikipedia.org/wiki/Rational_number)*, and finally defining* real*numbers as equivalence classes of their Cauchy sequences* or (*) *as Dedekind cuts, which are certain subsets of rational numbers.*" (https://en.wikipedia.org/wiki/Real_number#Definition)

(*) One does *not* really have a *choice* between the Dedekind route and the Cauchy route and one should *both*:

i. go the Dedekind route *and* extend the *metric* and then prove that the quotient is *metric*-complete,

and

ii. go the Cauchy route *and* extend the *order* and then prove that the quotient is *order*-complete,

and finally

iii. prove that the two quotients are both *metric*-isomorphic and *order*-isomorphic.

Which, unless you are a mathematician, is not exactly enlightening either. In any case, a very*, very tall order.*

2. Fractions and roots In fact, *at best*, that is even when the real number is a **fraction** or a **root**, a real number is only like a Birth Certificate in that the real number is just a *name* that says where the real number is coming from. But this name certainly does *not* provide by itself any indication of what the size of the real number is.

EXAMPLE 0.78.

- The *fraction* $\frac{4168}{703}$ is just a *name* for the solution of the equation $703\,x = 4168$ (Assuming the equation *has* a solution!)
- The *root* $\sqrt[3]{-17.3}$ is just a *name* for the solution of the equation $x^3 = -17.3$. (Assuming the equation *has* a solution!)

In textbooks it's of course the other way around,

However, this *best* case is actually extremely rare and most real numbers do not tell us by themselves where they are coming from which leaves us with no way to get even a rough idea of what the size of that real number might be.

You just have to find out from somewhere.

Example 0.79.

- π is just a *name* that does *not* say by itself that π is "*the ratio of a circle's circumference to its diameter*". (https://en.wikipedia.org/wiki/Pi)
- e is just a *name* that does *not* say by itself that e is "*a mathematical constant which appears in many different settings throughout mathematics*". (https://en.wikipedia.org/wiki/E_(mathematical_constant))

3. Calculating with real numbers. This can be done directly from the names *only* with the same two kinds of real numbers, that is when the real numbers are fractions or roots:

i. When the real numbers are fractions, there are procedure to compare, add, subtract, multiply and divide directly from the whole numbers that make up the fractions. (https://en.wikipedia.org/wiki/Rational_number#Arithmetic)

Example 0.80. To know which is the larger of $\frac{4168}{703}$ and $\frac{5167}{831}$ there is a procedure that involves only the wholenumbers 4168, 703, 5167 and 831.

ii. When the real numbers are roots, there are procedures to multiply and divide directly with the whole numbers that make up the roots but *not* to add or subtract. (https://en.wikipedia.org/wiki/Nth_root#Identities_and_properties)

Example 0.81. $\sqrt[2]{5} \times \sqrt[3]{7} = \sqrt[2\times 3]{5^3 \times 7^2}$

iii. However, it is usually not possible to calculate with both kinds of real numbers at the same time.

Example 0.82. Add e and π and/or figure out which of the two is larger. (Hint: you can't do either from the names.)

And, even when the real numbers are fractions and roots, things can still be difficult.

Example 0.83. Add $\sqrt[3]{64}$ and $\frac{876}{12}$ and/or figure out which of the two is larger. (Hint: in *this* case you *can* do both but *not* in the only slightly different case of $\sqrt[3]{65}$ and $\frac{875}{12}$.)

iv. Of course, the examples in textbools use mostly fractions and/or roots even though it is at the expense of being immensely misleading if only because *most* real numbers are *neither* fractions *nor* roots.

And at the expense of forcing memorization of scattered recipes.

approximate procedure

13* Approximating Real Numbers

The reason *engineers* and *physicists*, *chemists*, *biologists*, don't worry about real numbers is because they **approximate** real numbers with ... real-world numbers!!!

1. Approximation procedures. To begin with, one way or the other, *all* real numbers, *including* fractions and roots, come with a **procedure** for calculating approximations by numbers.

i. To approximate fractions, we use the division procedure.

EXAMPLE 0.84. To approximate $\frac{4168}{703}$, we *divide* 703 into 4168.

Few divisions end by themselves. Fortunately, though, when they don't, the more we push the division, the better the approximation.

ii. To approximate roots, we essentially proceed by trial and error.

EXAMPLE 0.85. To approximate $\sqrt[3]{17.3}$, we go:

- $1.0^3 = 1.0$
- $2.0^3 = 8.0$
- $3.0^3 = 27.0$,

Since 17.3 is between 8.0 and 27.0, $\sqrt[3]{17.3}$ *must* be somewhere between 2.0 and 3.0. (But *how* do we know that it *must*?) So now we go:

- $2.1^3 = 9.261$

  ...
- $2.5^3 = 15.620$
- $2.6^3 = 17.576$

Since 17.3 is between 15.620 and 17.576, $\sqrt[3]{17.3}$ *must* be between 2.5 and 2.6. (But *how* do we know that it *must*?)

And so on. (The actual procedure is more *efficient* but that's the idea.)

Of course, the more "exotic" the real number is, the more complicated the procedure for approximating is going to be:

EXAMPLE 0.86. There are many ways to approximate π. The simplest one is the Gregory-Leibniz series whose first few terms are:
$\frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} \quad \ldots$
However, even with "*500,000 terms, it produces only five correct decimal digits of π*" (https://en.wikipedia.org/wiki/Pi#Approximate_value) But

there are shorter if more complicated ways to approximate π.

[...]
largest permissible error

EXAMPLE 0.87. One of the very many ways to approximate e is:
$1+\frac{1}{1}+\frac{1}{1\cdot 2}+\frac{1}{1\cdot 2\cdot 3}+\frac{1}{1\cdot 2\cdot 3\cdot 4} \quad \dots$
(https://en.wikipedia.org/wiki/E_(mathematical_constant)#Asymptotics)

2. Approximation error. Since a real number is usually *not* equal to the real-world numbers used to approximate it, in order to write *equalities* we will have to use:

> **DEFINITION 0.16** will be the symbol for "some small-size number, positive or negative, whose size is too small to matter here".

In other words, [...] is a *signed* number about which the only thing we know is that the size of [...] is *less* than the **largest permissible error** whichi is the equivalent here of a tolerance.

EXAMPLE 0.88.

- $\frac{4168}{703} = 5.929 + [...]$ where [...] is less than 0.001 which is the largest permissible error. (Else the procedure would have generated 5.928 or 5.930 instead of 5.929.)
- $\sqrt[3]{17.3} = 2.586\,318\,666\,944\,673 + [...]$ where [...] is less than $0.000\,000\,000\,000\,001$ which is the largest permissible error. (Else the procedure would have generated $2.586\,318\,666\,944\,672$ or $2.586\,318\,666\,944\,674$ instead of $2.586\,318\,666\,944\,673$.)
- $\pi = 3.141\,5 + [...]$ where [...] is less than $0.000\,01$ which is the largest permissible error. (Else the procedure would have generated $3.141\,4$ or $3.141\,6$ instead of $3.141\,5$.)
- $e = 2.718\,281\,82 + [...]$ where [...] is less than $0.000\,000\,01$ which is the largest permissible error. (Else the procedure would have generated $2.718\,281\,81$ or $2.718\,281\,83$ instead of $2.718\,281\,82$.)

Conclusion

So, "*the wheel is come full circle*" (King Lear), from the real numbers all the way back to the real-world numbers, with just one question left:

And a good question it is. But then, it surely depends on what you mean by "learn"

> Why should people who want to *learn* CALCULUS have to use real numbers which they would then have to *approximate* with real-world numbers *anyhow*?

Well, since,

- To fully quote from Gowers in **?? ??** - **??** (**??**), "*Physical measurements* are not real numbers*. That is, a measurement of a physical quantity will . . .*"

And even if you wanted to become a mathematician, "Real Analysis becomes more intuitive when [one thinks of real numbers] as infinite decimals." (Gowers' https://www.dpmms.cam.ac.uk/~wtg10/decimals.html)

and

- Just like people, "[m]*ost calculators do* not *operate on real numbers. Instead, they work with finite-precision* [decimal] *approximations.*"(https://en.wikipedia.org/wiki/Real_number#In_computation.)

the answer must surely be, as *Engineers* used to be fond of saying, that

> "The *real* real numbers are the *decimal* numbers."

And now, Ladies and Gentlemen, let Calculus begin!

Part I

Functions Given By Data

The simplest way to give a function is to give the rekevant data, that is the numbers connected by the function.

Functions of various kinds are "the central things of investigation" in most fields of modern mathematics.

Michael Spivak [0]

connect
pair
2-tuple

Chapter 1

The Name Of The Game

. . . is 'function' of course! (`https://idioms.thefreedictionary.com/the+name+of+the+game`)

1 Relations

Leonardo da Vinci is often quoted as having said that *Everything **Connects** to Everything Else*.[1](`https://medium.com/@nikitavoloboev/everything-connects-to-everything-els`
And, indeed, Da Vinci's statement is at the very heart of all SCIENCES.

Even if we can't always see the connections.

EXAMPLE 1.1. Everything sits on something else: people sit on chairs that sit on floors that sit on joists that sit on walls that sit on . . .

1. Ordered pairs. An **ordered pair** of things is two things in a *given* order. (`https://en.wikipedia.org/wiki/Ordered_pair`)

> **LANGUAGE NOTE 1.1** An ordered pair is also called a **2-tuple** but we will not use the word.

[0] Calculus, 4th edition. Publish or Perish Press. `https://en.wikipedia.org/wiki/Michael_Spivak`

[1] According to `https://quoteinvestigator.com/2022/03/31/connected/`. However, the earliest *published* version is from Gotthold Ephraim Lessing in 1769.

relation
thing
left thing
right thing
diagram

The standard format for writing an ordered pair is to write the two things in the given order, separated by a comma, between parentheses.

EXAMPLE 1.2. The ordered pair (Eiffel Tower, Empire State Building) is *not* the same as the ordered pair (Empire State Building, Eiffel Tower)

CAUTIONARY NOTE 1.1 In MATHEMATICS:
- An *ordered* pair

is not to be confused with
- A *pair*, which is just a collection of two things so that the order in which the two things are given is *irrelevant*.

Nevertheless, since, in this text, we will be using ordered pairs to record how things are connected, the order in which two things are given will *always* be relevant and so

Just as in "a pair of gloves".

AGREEMENT 1.1 We will let the qualifier "ordered" *go without saying* and use the word pair as short for ordered pair.
But, as usual, for a while we will write (ordered) pair as a reminder.

2. Connected things. The mathematical concept behind Da Vinci's connections is that of a **relation** (https://en.wikipedia.org/wiki/Relation_(mathematics)) which has two components:

A. The first component of a relation is two collections of **things** namely:
- A collection of things we will refer to as **left things**,
- A collection of things we will refer to as **right things**

As in Port and Starboard (https://aceboater.com/en/starboard-port-side-defini

AGREEMENT 1.2
To make it easier to distinguish left things from right things, we will use:
- Pink boxes for left things as in, for instance,

  Jill, x, -0.053, x_0, 0 ∞, *small*, *large*,
- Green boxes for right things as in, for instance,

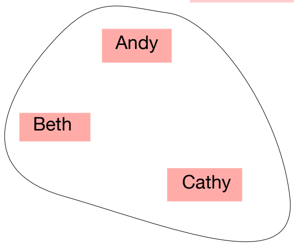
  Jack, y, $+32.14$, y_0, 0 ∞, *small*, *large*,

There are essentially two ways to give the two collections of things:
- One way is by way of **diagrams** which is the most immediately intuitive way, (https://en.wikipedia.org/wiki/Diagram)

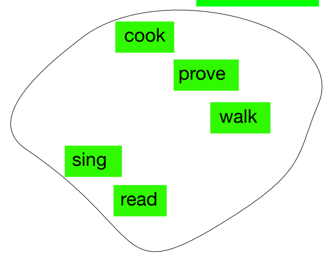
connector
related
arrow connector
arrow diagram

EXAMPLE 1.3. Diagrams for:

A collection of Persons :



A collection of activities



- The other way is by way of lists, which is the way that lends itself to PROCEDURES.

EXAMPLE 1.3. (Continued) Lists for:

A collection of Persons :
Andy , Beth , Cathy .

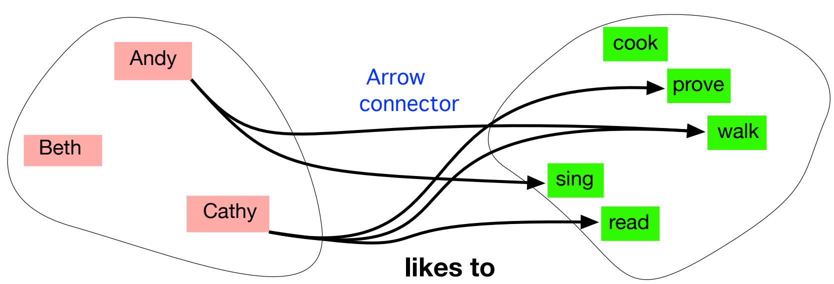
A collection of activities
walk , sing , cook , prove , read .

B. The second component of a relation is the **connector**, that is anything that pairs some of the left things with some of the right things . When the connector pairs a left thing to a right things , we will say that the left thing is **related** to the right things and/or that the (ordered) pair (left thing , right thing) is a related pair.

Why can't we use relator *instead of* connector*?*

- When the collections of things are given with diagrams, the connector will be in the shape of an **arrow connector** which will give the relation as an **arrow diagram**.

EXAMPLE 1.4. The collections in EXAMPLE 1.3 (Page 65) could for instance be connected into the arrow diagram:



which, for instance, says that (Andy , walk) and (Andy , sing) are both related pairs but (Cathy , cook) is not a related pair.

In everyday language: Andy likes both walking and singing but Cathy doesn't like cooking.

- When the collections in a relation are given by lists, the connector will be given by the list of all related (ordered) pairs taken from among all possible (ordered) pairs

EXAMPLE 1.4. (Continued) The connector could also be given as the list of *related* pairs:
(Andy , walk) (Andy , sing)
(Cathy , prove) (Cathy , walk) (Cathy , read)
taken from from the list of all possible pairs
(Andy , cook) (Andy , prove) (Andy , walk) (Andy , sing) (Andy , read)
(Beth , cook) (Beth , prove) (Beth , walk) (Beth , sing) (Beth , read)
(Cathy , cook) (Cathy , prove) (Cathy , walk) (Cathy , sing) (Cathy , read)

3. (left thing , right thing) pairs. Since we will be dealing with relations, the (ordered) pairs we will be dealing with will always be the (left thing , right thing) pairs in some relation. But of course, given a relation, other (ordered) pairs could always be floating around that have little or nothing to do with the given relation.

EXAMPLE 1.4. (Continued)

- Since Jack is *not* in the collection of Persons , (Jack, prove) is an (ordered) pair but *not* a (Person , activity) pair
- Since swimming is *not* in the collection of activities , (Beth , swim) is an (ordered) pair but *not* a (Person , activity) pair.
- Since Cathy *does not* like to cook , (Cathy , cook) is a (Person , activity) pair but *not* a *related* (Person , activity) pair.
- Since Andy likes to walk , (Andy , walk) is not only a (Person , activity) pair but a *related* (Person , activity) pair.

Given a relation, the **set of (left thing , right thing) pairs** is the collection of all the *related* (left thing , right thing) pairs.

EXAMPLE 1.4. (Continued) The set of (Person , activity) pairs is:
(Andy , walk), (Andy , sing), (Cathy , walk),

(Cathy , read), (Cathy , prove)

table
row
column
list table
Cartesian table
Descartes

4. Tables. Using lists to give a relation, though, is tedious and relations are often given in the shape of **tables** in which the collections of things are listed in **rows** and **columns** in a way that shows the (left thing , right thing) pairs. (https://en.wikipedia.org/wiki/Table_(information))

Among other kinds of tables, there are:

- **List tables** in which the collection of left things is listed in the lefthand column and for each left thing the related right thing(s) , if any, are listed horizontally in the righthand column.

Example 1.5. The list table for the relation in Example 1.4 (Page 65) is

Persons	activities , if any, that Persons , if any, like
Andy	walk sing
Beth	
Cathy	read walk prove
	cook

- **Cartesian table**—named after **René Descartes** (https://en.wikipedia.org/wiki/Ren%C3%A9_Descartes)[1]—which are much more systematic than list tables:
  - All the left things are listed in a vertical column on the left,
  - All the right things are listed in a horizontal row on top,
  - For each (left thing , right thing) the word yes or no at the intersection of the horizontal row of the left thing and the vertical column of the right thng indicates whether or not the left thing is related to the right thing .

Just a bit less obvious to read, though.

Example 1.6. The Cartesian table for the relation in Example 1.4 (Page 65) is:

[1]Inventor of Analytic Geometry which links the previously separate fields of Algebra and Geometry (https://en.wikipedia.org/wiki/Analytic_geometry)

likes to	walk	sing	read	prove	cook
Andy	yes	yes	no	no	no
Beth	no	no	no	no	no
Cathy	yes	no	yes	yes	no

where, for instance,

likes to	prove
Cathy	yes

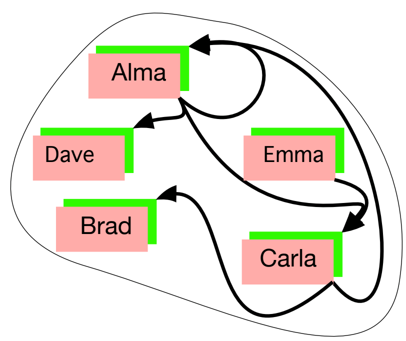
says that Cathy likes to prove.

Just for the sake of precision!

5. Endorelations. There is no reason why the collection of left things and the collection of right things cannot be one and the same.

Example 1.7.

Arrow diagram:



List table:

Persons	Persons, if any, whom Persons like
Alma	Alma Carla Dave
Brad	
Carla	Alma Brad
Dave	
Emma	Carla
	Emma

endorelation

Cartesian table:

likes	Alma	Brad	Carla	Dave	Emma
Alma	yes	yes	yes	no	no
Brad	no	no	no	no	no
Carla	yes	yes	no	no	no
Dave	no	no	no	no	no
Emma	no	no	yes	no	no

LANGUAGE NOTE 1.2 While relations in which left things and right things are from one and the same collection are called **endorelations**, *we* will just keep on using the word relation.
https://en.wikipedia.org/wiki/Homogeneous_relation

6. Relation problems. Given a relation, there are of course many questions we can ask.

a. How we will *proceed* to answer these questions will depend on how the relation is given:

- Arrow diagrams are intuitive but only so long as there are very few things in the collections and so we will not use arrow diagrams very often.
- List tables are clear and allow for quite a few things in the collections—but still not too many,
- Cartesian tables allow for just about any number of things in the collections.

b. The simplest question we may ask is if a given (left thing, right thing) pair is or is not related.

This question will in fact turn out to be basic for picturing relations.

EXAMPLE 1.8. In EXAMPLE 1.4 (Page 65) we may ask:

Does Cathy like to sing?

Answer: No, so the pair (Cathy, sing) is not a related pair

Does Cathy like to prove?

Answer: Yes, so the pair (Cathy, prove) is a related pair.

c. A consequence of Da Vinci's statement is that, in fact, any given thing is known only by what is known of the things that the given thing is connected to.

relation problem
left problem

EXAMPLE 1.9. Variants of the idea that things are known by what is known of the things they are connected to are found in many cultures:

You tell me the company you keep, I will then tell you what you are (Dutch)
You tell me who's your friend, I will then tell you who you are (Russian)
You tell me your company, I will then tell you who you are (Irish)
You tell me what you are eager to buy, I will then tell you what you are (Mexican)
You tell me with whom you go, I will then tell you what you do (English)
You tell me who your father is, I will then tell you who you are (Philippine)
You tell me what you eat, I will then tell you what you are (French)
(https://www.linkedin.com/pulse/show-me-your-friends-ill-tell-you-who-really-jar

So, the more consequential questions we may ask about a given relation fall into two general kinds of **relation problems**:

- **Left thing problems** in which we want to find information about a given left thing in terms of the right thing(s), if any, that the given left thing is related to.

i. List tables make it particularly easy to solve left thing problems: look up the given left thing in the left column and you will see the right things that the given left thing is related to listed on that row.

EXAMPLE 1.10. If, for the relation given in EXAMPLE 1.4 (Page 65), we ask for all the activities which Cathy likes, the list table in EXAMPLE 1.5 (Page 67) shows:

Cathy	read	walk	prove

If we ask for all the activities which Beth likes, the list table in EXAMPLE 1.5 (Page 67) shows:

Beth			

And, similarly, the list table in EXAMPLE 1.5 (Page 67) even gives answers to questions such as:

Is there any activity Beth likes? (Answer: No)
Does Cathy like *all* activities? (Answer: No)
Does Andy like *at least one* activity? (Answer: Yes)

ii. Cartesian tables are just a bit harder to use: look up the given left thing in the left column and the right things that the given left thing is related to, if any, will be in the columns with the word **yes**.

right problem

EXAMPLE 1.11. If we ask for all the activities which Cathy likes, the Cartesian table in EXAMPLE 16.16 (Page 490) shows:

likes to	walk		read	prove	
Cathy	**yes**	no	**yes**	**yes**	no

And if we ask for all the activities whichh Beth likes, the Cartesian table in EXAMPLE 16.16 (Page 490) shows:

likes to					
Beth	no	no	no	no	no

And, similarly, the Cartesian table in EXAMPLE 16.16 (Page 490) even gives answers to questions such as:
Is there any activity Beth likes? (Answer: No)
Does Cathy like *all* activities? (Answer: No)
Does Andy like *at least one* activity? (Answer: Yes)

- **Right thing problems** in which we want to find information about a given right thing in terms of the left thing(s), if any, that are related to the given right thing.

i. List tables are fairly unsuited to solving right thing problems because you have to hunt for the given right thing in all the rows of the right hand column.

EXAMPLE 1.12. If, for the relation given in EXAMPLE 1.4 (Page 65), we ask for all the Persons who like to walk, the list table in EXAMPLE 1.5 (Page 67) showa:

Persons	activities, if any, that Persons like
Andy	**walk** sing
Beth	
Cathy	read **walk** prove
	cook

If we ask for all all the Persons who like to cook, the list table in EXAMPLE 1.5 (Page 67) showa:

Persons	activities, if any, that Persons like
Andy	walk sing
Beth	
Cathy	read walk prove
	cook

And similarly, the list table in EXAMPLE 1.5 (Page 67) even gives answers to questions such as:
Is there at least one Person who likes cook? (Answer: No)
Is there at least one Person who likes walk? (Answer: Yes)
Do *all* Persons like walk? (Answer: No)

ii. Cartesian tables make it just as easy to solve right thing problems as to solve left thing problems: look up the given right thing in the top row and the left thing(s) that are related to the given right thing, if any, will be in the rows with the word yes.

EXAMPLE 1.13. If we ask for all the Persons who like to walk, the Cartesian table in EXAMPLE 16.16 (Page 490) shows:

likes to	walk				
Andy	**yes**				
Beth	no				
Cathy	**yes**				

If we ask for all the Persons who like to cook, the Cartesian table in EXAMPLE 16.16 (Page 490) shows:

likes to					cook
Andy					no
Beth					no
Cathy					no

And, similarly, the Cartesian table in EXAMPLE 16.16 (Page 490) even gives answers to questions such as:
Is there at least one Person who likes to cook? (Answer: No)
Is there at least one Person who likes to walk? (Answer: Yes)
Do *all* Persons like to walk? (Answer: No)

2 Picturing Relations

left number
right number

Arrow diagrams are a very natural and very visual way to picture relations but we will need ways to picture relations that are a lot more systematic.

1. Basic picture. A relation that involves only just a few related (left thing , right thing) pairs can be easily pictured with just a ruler for left things and a ruler for right things .

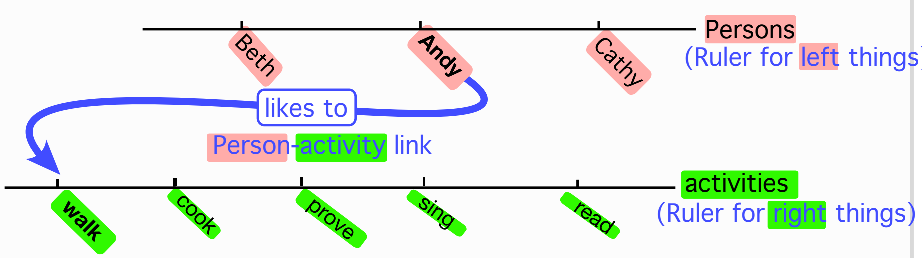
PROCEDURE 1.1 To picture a (left thing , right thing) pair,

i. Mark the given left thing as in Subsection 6.1 - Comparing given numbers (Page 19) on the ruler for left things ,

ii. Mark the given right thing as in Subsection 6.1 - Comparing given numbers (Page 19) on the ruler for right things ,

iii. If the given left thing is related to the given right thing draw a **link** from the given left thing marked on the ruler for left things to the given related right thing marked on the ruler for right things .

DEMO 1.1 Picture the related pair (Andy , walk) in EXAMPLE 1.7 (Page 68)



where the link says that Andy likes to walk ,

2. Quantitative Cartesian setup. Now, while relations can involve any kind of things, this text will deal only with (endo)relations involving *numbers*—hence **left numbers** and **right numbers**—and so the set of

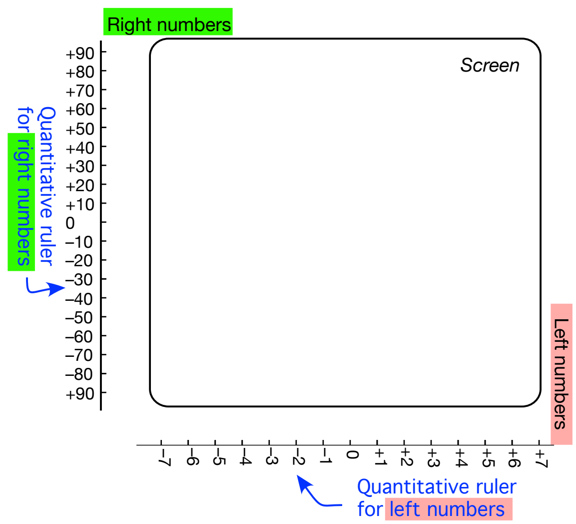
quantitative Cartesian setup
screen
quantitative ruler for left numbers
quantitative ruler for right numbers
left number level line
right number level line
plot dot

related pairs will usually be large and so picturing those relations will require a more efficient setup than just two rulers.

A **quantitative Cartesian setup** consists of:

- A rectangular area which we will call the **screen**.
- A **quantitative ruler for left numbers** *below* the screen (horizontal)
- A **quantitative ruler for right numbers** *left* of the screen (vertical)

EXAMPLE 1.14.



Yeah, sure enough, Cartesian setups *are upside down from Cartesian* tables.

3. Plotting pairs of numbers.

PROCEDURE 1.2 To **plot** a given pair of numbers,

i. Tickmark the given left number on the quantitative ruler for left numbers (horizontal),
ii. Draw a **left number level line**, that is a *vertical* line through the tickmark for the given left number,
iii. Tickmark the given right number on the quantitative ruler for right numbers (vertical),
iv. Draw a **right number level line**, that is a *horizontal* line through the tickmark for the given right number,
v. Then, at the intersection of the left number level line and the right number level line, mark the **plot dot** with:

- A **solid dot** if the (left number , right number) pair is related.

or, as we wll need occasionally,

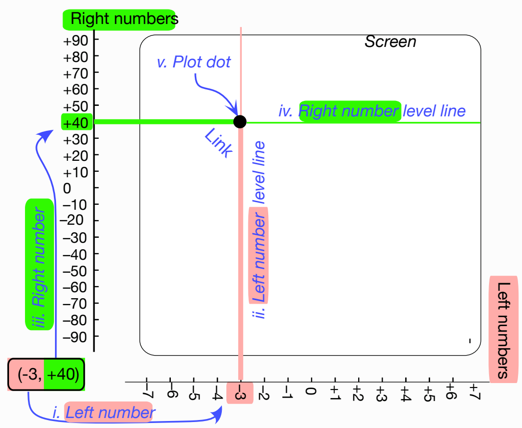
- A **hollow dot** if the (left number , right number) pair is *not* related

solid dot
hollow dot
data point
plot point

.

DEMO 1.2
Plot the related pair (-3 , $+40$).

i. We tickmark -3 on the ruler for left numbers ,
ii. We draw a left number level line (vertical) through -3 ,
iii. We tickmark $+40$ on the ruler for right numbers ,
iv. We draw a right number level line (horizontal) through $+40$,
v. Since the given pair is *related*, we use a solid dot to mark the intersection of the left number level line with the right number level.



Note that the plot dot is at the elbow of the link.

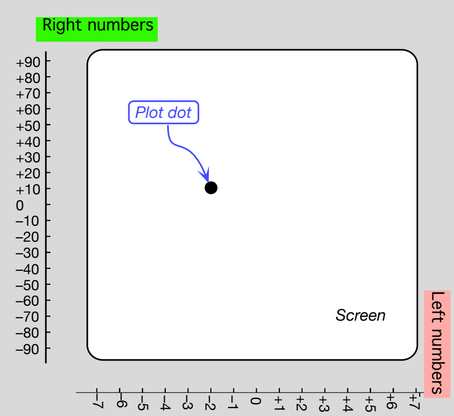
LANGUAGE NOTE 1.3 The word usually used in MATHEMATICS instead of plot dot is **plot point** and, in the experimental sciences, **data point** but *we* cannot do that since we are already using the word point with a different meaning. Subsection 4.1 - Non-zero digits (Page 14)

4. Reading plot dots. The other way round,

PROCEDURE 1.3 To get the pair of numbers from a given plot dot,

i. Draw a left number level line (vertical) through the given plot dot,
ii. Read the left number where the left number level line intersects the ruler for left numbers,
iii. Draw a right number level line (horizontal) throught the given plot dot.
iv. Read the right number where the right number level line intersects the ruler for right numbers,

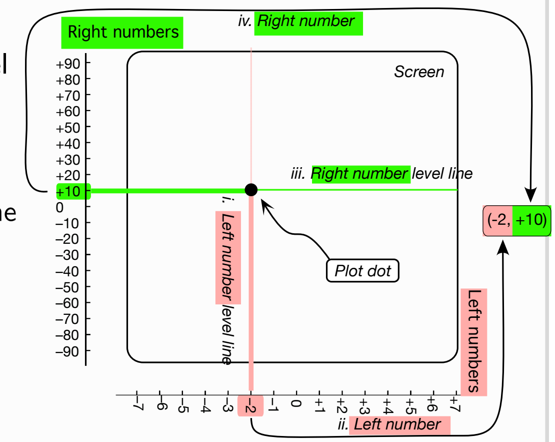
DEMO 1.3
Given the plot dot



get the pair of numbers.

set of plot dots

i. We draw a left number level line (vertical) through the plot dot,
ii. We read the left number where the left number level line intersects the ruler for left number : -2
iii. We draw a right number level line (horizontal) through the plot dot,
iv. We read the right number where the right number level line intersects the ruler for right numbers : $+10$

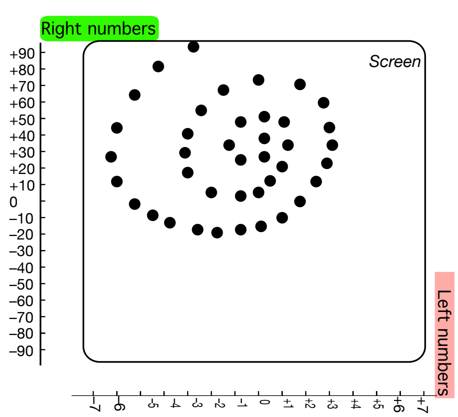


So the pair of numbers pictured by the given plot dot is $\left(-2, +10\right)$

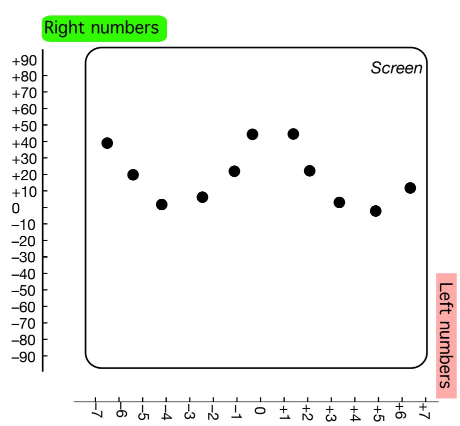
3 Relations Given By Sets Of Plot Dots

1. Sets of plot dots. Since quantitative Cartesian setups allow us to picture large sets of pairs, we can picture a given relation with a **set of plot dots**, that is with the plot dots for all the *related* left number right number pairs.

Example 1.15.

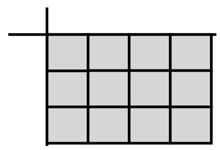


Example 1.16.

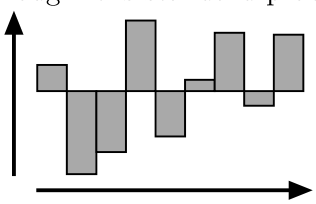


histogram
bar graph
x-axis
y-axis
axis

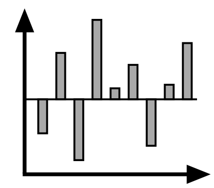
2. Axes. Keeping the ruler for left numbers and the ruler for right numbers *away* from the screen as we do in the Cartesian setup is not the usual practice in MATHEMATICS even though it is standard practice in the *real world*:



Cartesian Table

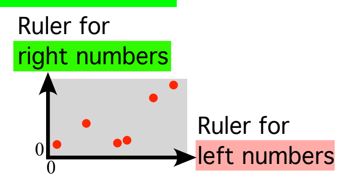


Histogram [2]



Bar graph[3]

As indicated by the word Cartesian, the quantitative Cartesian setup is due to Descartes who, since he did not use *negative* numbers, had no problem using the 0 level line as ruler for left numbers and the 0 level line as ruler for right numbers since they were not in the way.



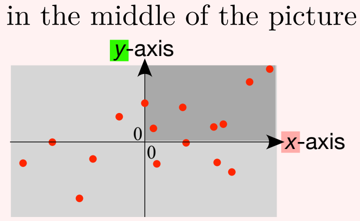
Descartes

But when mathematicians eventually did accept *negative* numbers, they continued to use:

- the 0 level line as ruler for left numbers—which they then called ***x*-axis** and
- the 0 level line as ruler for right numbers—which they then called ***y*-axis**

even though :

CAUTIONARY NOTE 1.2
Since **axes** are in the middle of the picture



Modern

using the x-axis as ruler for left numbers can be confusing because:

---

[2]See https://en.wikipedia.org/wiki/Histogram
[3]See https://en.wikipedia.org/wiki/Bar_chart

quincunx

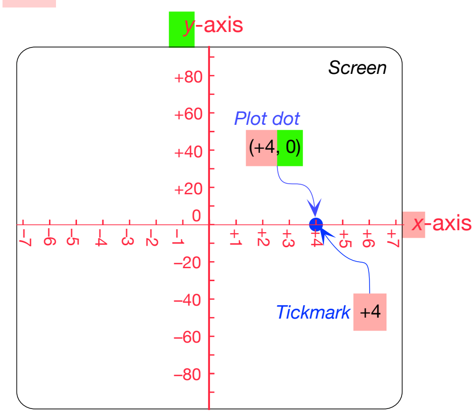
- The plot dot for the pair $\left(x_0, 0\right)$ will then be on top of the tickmark for the left number x_0 which makes it unclear which is intended,

and, similarly, using the y-axis as ruler for right numbers can be confusing because:

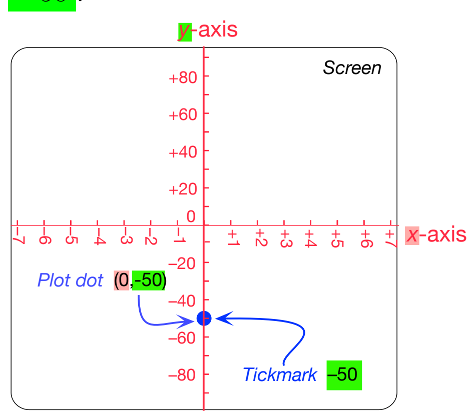
- The plot dot for the pair $\left(0, y_0\right)$ will then be on top of the tickmark for the right number y_0 which makes it unclear which is intended.

EXAMPLE 1.17. When using the axes as rulers.

The plot dot for the pair $(+4, 0)$ is on top of the tickmark for the left number $+4$:



The plot dot for the pair $(0, -50)$ is on top of the tickmark for the right number -50:



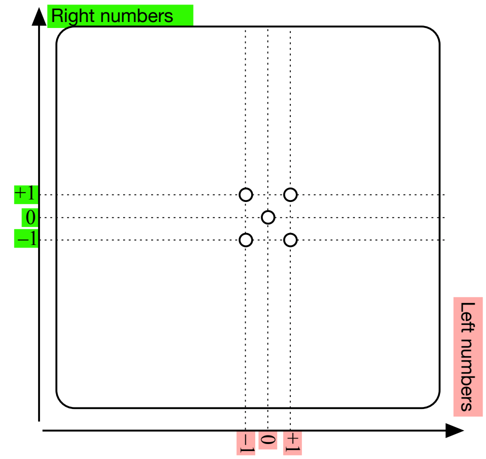
3. The quincunx. We will call **quincunx** (https://en.wikipedia.org/wiki/Quincunx) the set of plot dots for the following five pairs:

left number problem

$(-1, +1)$ $(+1, +1)$

$(0, 0)$

$(-1, -1)$ $(+1, -1)$



Note that here the plot dots are hollow dots because, at this time, we don't know which of the five left number right number pairs in the quincunx, if any, are *related*. In fact, we will see that which of the five left number right number pairs in the quincunx are *related* will play a central role with 'power functions'.

4. Sparseness of sets of plot dots. In *engineering* and the *experimental sciences*, aside from being given by Cartesian tables, relations are often given by a set of plot dots generated by some machinery (`https://en.wikipedia.org/wiki/Plotter`) on the screen of a quantitative Cartesian setup.

However, while, when the relation is given by a set of plot dots, the PROCEDURES for solving relation problems are fairly obvious as we will now see, what can complicate matters is that

CAUTIONARY NOTE 1.3 Sets of plot dots are **sparse**, that is, there are only so many plot dots surrounded by a lot of empty space.

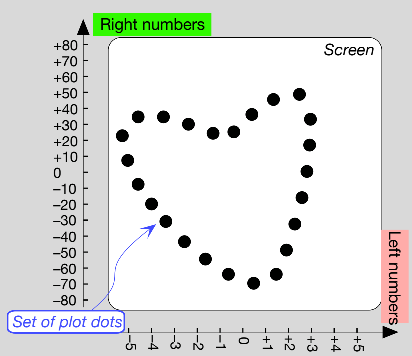
5. Left number problems. To solve a **left number problem** when the relation is given by a set of plot dots, we use

PROCEDURE 1.4 To get the right number(s) (if any) related to a given left number when the relation is given by a set of plot dots,

i. Tickmark the given left number on the ruler for left numbers,
ii. Draw a left number level line through the tickmark for the given left number,
iii. Mark the plot dot(s), if any, where the left number level line intersect the given set of plot dots, (This is where the fact that sets of plot dots are sparse can come in with a vengeance.)
iv. Draw a right number level line through each plot dot,
v. Read the right number(s) related to the given left number, if any, where the right number level line(s) intersect(s) the ruler for right numbers,

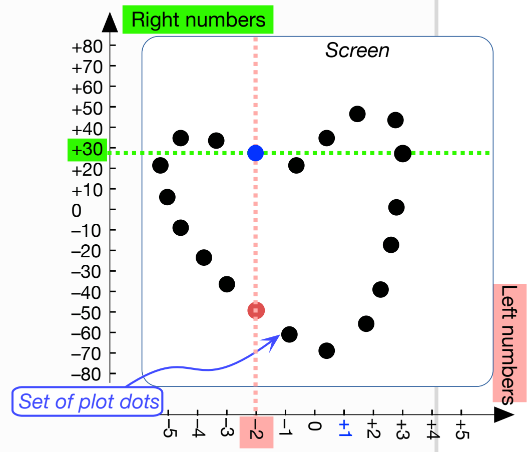
DEMO 1.4a

Given the set of plot dots



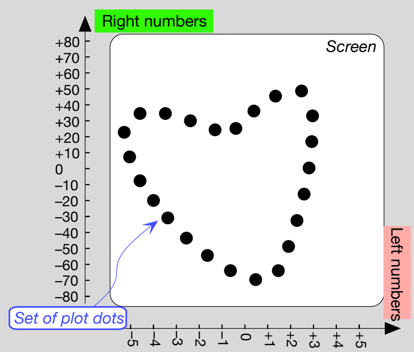
and given the left number -2, get the right number(s) related to -2, if any.

i. We tickmark -2 on the ruler for left numbers
ii. We draw a left number level line through the tickmark for -2,
iii. We mark the plot dot(s) where the left number level line through -2 intersects the set of plot dots, if at all
iv. We draw the right number level line through the marked plot dot(s), if any,
v. We read the right number related to -2 where the right number level line intersect the ruler for right numbers: $+30$.



DEMO 1.4b

Given the set of plot dots



and given the left number $+2$, get the right number(s) related to $+2$, if any..

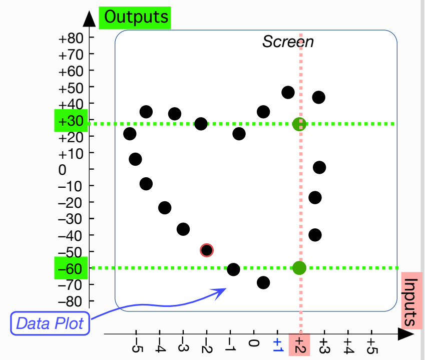
i. We tickmark $+2$ on the ruler for left numbers

ii. We draw a left number level line through the tickmark for $+2$,

iii. We mark the plot dot(s) where the left number level line through $+2$ intersects the set of plot dots, if at all

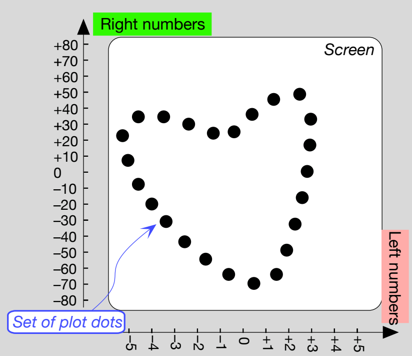
iv. We draw the right number level line through eacn of the marked plot dot(s), if any,

v. We read the right number(s) related to $+2$ where the right number level line intersect the ruler for right numbers: $+30$ and -60..



DEMO 1.4c

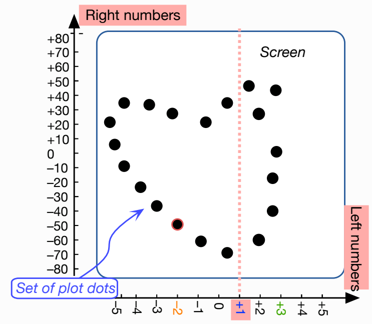
Given the set of plot dots



and given the left number $+1$, get the right number(s) related to $+1$, if any.

right number problem

i. We tickmark $+1$ on the ruler for left numbers
ii. We draw a left number level line through the tickmark for $+1$,
iii. There is no plot dot where the left number level line through $+1$ intersects the set of plot dots. (This is where the fact that sets of plot dots are sparse comes in.)
iv. We cannot draw any right number level line since there isn't any marked plot dot,
v. There is no right number related to $+1$.



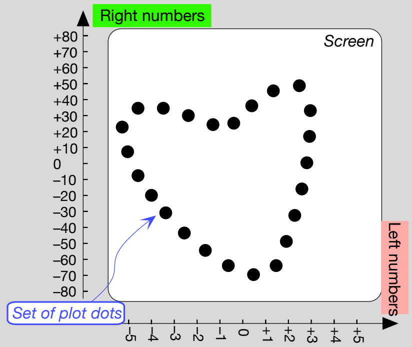
6. Right number problems. To solve a **right number problem** when the relation is given by a set of plot dots, we use

PROCEDURE 1.5 To get the left number(s) (if any) related to a given right number when the relation is given by a set of plot dots,

i. Tickmark the given right number on the ruler for right numbers,
ii. Draw a right number level line through the tickmark for the given right number
iii. Mark the plot dot(s), if any, where the right number level line intersect the given set of plot dots, (This is where the fact that sets of plot dots are sparse can come in with a vengeance.)
iv. Draw a left number level line through each plot dot,
v. Read the left number(s) related to the given right number, if any, where the left number level line(s) intersect(s) the ruler for left numbers,

DEMO 1.5a

Given the set of plot dots



and given the right number $+30$, get the left number related to $+30$, if any.

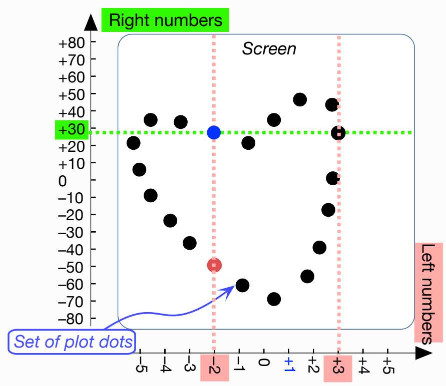
i. We tickmark $+30$ on the ruler for right numbers

ii. We draw a right number level line through the tickmark for $+30$,

iii. We mark the two plot points where the right number level line intersects the set of plot dots,

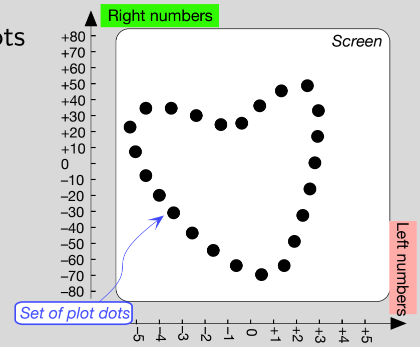
iv. We draw a left number level line through each of the two marked plot dots,

v. We read the related left numbers where the left number level lines intersect the ruler for left number: -2, $+3$.



DEMO 1.5b

Given the set of plot dots



and given the right number -50, get the left number related to -50, if any.

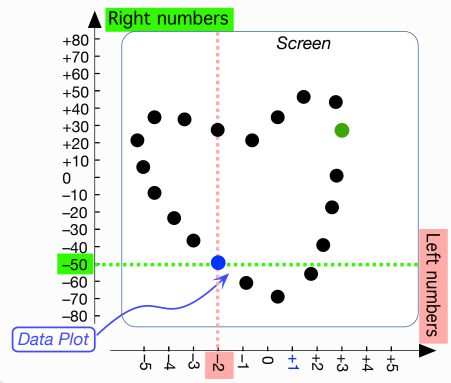
i. We tickmark -50 on the ruler for right numbers

ii. We draw a right number level line through the tickmark for -50,

iii. We mark the single plot point where the right number level line intersects the set of plot dots,

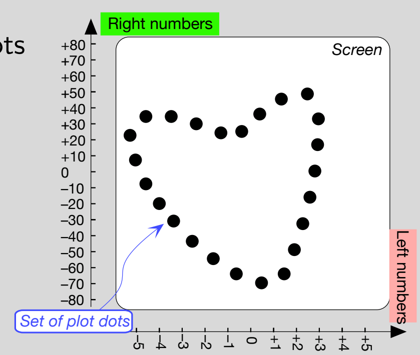
iv. We draw a left number level line through the marked plot dot,

v. We read the related left number where the left number level line intersect the ruler for left number: -2.



DEMO 1.5c

Given the set of plot dots



and given the right number -30, get the left number related to -30, if any.

change

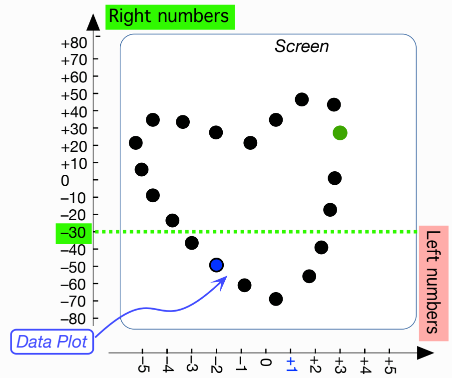
i. We tickmark -30 on the ruler for right numbers

ii. We draw a right number level line through the tickmark for -30,

iii. There is no plot point where the right number level line intersects the set of plot dots. (This is where the fact that setx of plot dots are sparse comes in.)

iv. We cannot draw a left number level line through a marked plot dot,

v. There is no left number related to the given -30.



4 Functions

To see that something is **changing**, and to record the change, we *must* look at that thing in relation to something else.

EXAMPLE 1.18. To realize that:

- The airplane we are sitting in is moving, we must look out the window.
- The tree we see out our window is growing, we must look at it in relation with somethings like a building..

This is even more the case for *quantitative* changes.

EXAMPLE 1.19. We might say that someone's income tax was \$2 270 but, *by itself*, that wouldn't be much information because, for instance,

- \$2 270 of income tax was a lot more money in Year 1913—the year income tax was first established, than, say, a century later, in Year 2013. So, for saying that someone's income tax is \$2 270 to be information, we would have to have some relation pairing years with Income Tax,
- \$2 270 of income tax is a lot more money for the rest of us than for billionaires. So, for saying that someone's income tax is \$2 270 to be information, we would have to have some relation pairing Incomes with

function
function requirement

Income Tax.

However, the fact that a relation can relate *one* same left number to *many* different right numbers can make *differences* difficult to see.

Example 1.20. Consider the following:

- A *slot machine* can pair a number of coins with just about any number of coins which makes it quite hard to decide if *this* slot machine is better for gambling than *that* other slot machine.

- A *parking meter* can pair a number of coins with only *one* number of minutes which makes it easy to decide if *this* parking meter is better for parking than *that* other parking meter.

1. Function requirement. So, from now on we will restrict ourselves to **functions**, that is to relations that meet

Definition 1.1 The **function requirement** for a relation:

No left number can be related to *more than one* right number.

that is, in other words,

A left number can be related to *no more than one* right number.

that is, still in other words,

A left number can be related to *at most one* right number.

Example 1.21. In Example 1.14 (Page 74)

- The *slot machine* does *not* meet the Local behaviour coding format (Definition 2.1, Page 135) because even when two Persons put the same amount of money in a given slot machine, the slot machine can give different amounts of money to the two Persons.
- The *parking meter does* meet the Local behaviour coding format (Definition 2.1, Page 135) because whenever two Persons put the same amount of money into a given parking meter, the parking meter will always give the two Persons the same number of minutes.

Example 1.22. As opposed to the relation in Example 1.5 (Page 67), the relation given by the table

input/output device
I-O device
input
output
return

Persons :	Activities, if any, these Persons like:
Dave	skate
Eddy	
Fran	sing

satisfies the Local behaviour coding format (Definition 2.1, Page 135)

2. Inputs and outputs. The Local behaviour coding format (Definition 2.1, Page 135) actually makes a big qualitative difference among relations in that, while relations are essentially embodiments of Da Vinci's statement so that there is no precedence whatsoever between left things and right things, with functions, as we will see, left things somehow come "before" right things and, in fact, functions are seen as **input/output devices**, **I-O device** for short. (https://en.wikipedia.org/wiki/Input/output)

So, to acknowledge this precedence in the case of a function, we will use the word **input** instead of left number and the word **output** instead of right number. Then, instead of saying that an output is related to an input we will say that the function **returns** the output for the given input.

We can then rephrase the Local behaviour coding format (Definition 2.1, Page 135) as follows:

Definition 2.1 (Restated) Local behaviour coding format

Given an input, a function can*not* return *more than one* output.

words

Given an input, a function can return *no more than one* output.

that is, still in other words,

Given an input, a function can return *at most one* output.

So, according to Definition (Restated) 2.1 - Local behaviour coding format (Page 89):

domain

A. Given an input, a function may or may not return an output, that is there *may* or may *not* be an output related to the given input.

And even inside *"casual"* *MATHEMATICS !*

Now, while this is fairly standard practice *outside* of MATHEMATICS, strictly speaking, functions should *not* be allowed to return *no* output because that would eventually cause a *theoretical* difficulty and so one should introduce the word **domain** for the collection of inputs for whch the function *does* return an output.

EXAMPLE 1.22. (Continued) For a mathematician, the given relation is not a function and only its restriction to its domain, that is Dave, Fran, is a function. satisfies the Local behaviour coding format (DEFINITION 2.1, Page 135)

However, inasmuch as *we* will not be anywhere near encountering this theoretical difficulty:

In other words, this *text is trading rigor for expository simplicity. And this* theoretical *difficulty wouldn't come up any time soon anyhow.*

> **CAUTIONARY NOTE 1.4** In *this* text, given an input, a function *may* return *no* output.

Of course, you might say that no tax = $0.00 *so this may not a very good example for* CAUTIONARY NOTE *1.4 - Inputs with no output (Page 90).*

EXAMPLE 1.23. The relation given by the income tax tables of the IRS satisfies the Local behaviour coding format (DEFINITION 2.1, Page 135) and is thus, in the real world, a function even though incomes below the minimum income cause no income tax.

B. On the other hand, it is quite possible for a function to pair *many* different inputs to *one* and the same output. In other words, the very same output may be returned by a function for *many* different inputs.

> **CAUTIONARY NOTE 1.5** A function *may* return the *same* output for several *different* given insputs

EXAMPLE 1.24. A business may be looked upon as the function given by the *input-output table* of its profits/losses over the years:

Fiscal Year	Profit/Loss
1998	$+5\,000$
1999	$-2\,000$
2000	
2001	$+5\,000$
2002	$-2\,000$
2003	$-1\,000$
2004	
2005	$+5\,000$

In 1998, 2001, and 2005 the business returned the same profit/loss namely $+5\,000$

capital script letters
f
$f(x)$

3. Language for functions. Since functions "*are widely used in science, and in most fields of mathematics.*" (https://en.wikipedia.org/wiki/Function_(mathematics)), just as with ARITHMETIC, a whole language was created as the development of CALCULUS proceeded, with variants depending on what aspect of CALCULUS was being developed, so we will now descrlbe the particular variant *we* will use.

In many ways, functions *are to* CALCULUS *very much what* numbers *are to* ARITHMETIC.

i. Individual symbols: We will use **capital script letters** to write the names of *given* functions.

EXAMPLE 1.25. Say $\mathcal{JOE}$ is the name of our favorite parking meter. Then 25 cents might be what we want to input in the function $\mathcal{JOE}$ and 10 minutes might be the parking time that $\mathcal{JOE}$ will return.

ii. Generic symbols: The following symbols are completely standard:

a. We will use $\boldsymbol{f}$ as symbol for a *generic* function.

b. We will use x as a global variable for input *numbers*, that is as a placeholder fot insputs

c. We will then use $\boldsymbol{f(x)}$, to be read f of x, as symbol for the output, if any, that the function f returns for the input x

And absolutely necessary if you don't happen to have color pens.

CAUTIONARY NOTE 1.6 Even though, because of the color boxes, we *could* write just $f\,x$ instead of $f(x)$, we will still use parentheses because that's what is done by absolutely everybody.

By the way, there *is* an alternate, parenthesis-free, notation:

$\xrightarrow{f}$
arrow notation
send
function problem
direct problem

Try to find a CALCULUS *text that does!*

LANGUAGE NOTE 1.4 Reverse Polish Notation
The Reverse Polish Notation (RPN) is another way to write the *output* of a function f for an input x, namely $x\,f$ instead of $f(x)$.

RPN is a much better notation, if only because $x\,f$ is parenthesis *free*, but we shall *not* use it as, unfortunately, about no one in the *mathematical* world does.
(https://en.wikipedia.org/wiki/Reverse_Polish_notation)

Er, $\xrightarrow{f}$ *isn't standard in* ...standard CALCULUS *texts.*

d. We *will* use the symbol $\xrightarrow{f}$ to write functions in **arrow notation**:

$$x \xrightarrow{f} f(x)$$

Inasmuch as we read from left to right, though, the arrow notation

$$x \xrightarrow{f} f(x)$$

tends to place an emphasis on the input rather than on the function f and there is an

LANGUAGE NOTE 1.5 Alternate arrow notation. In order to place the emphasis on the function instead of on the input, a standard alternative is to write:

$$f : x \longrightarrow f(x)$$

which is read

$$f \textbf{ sends } x \text{ to } f(x)$$

While we will *not* use the *alternate* arrow notation, we *will* use the worf send because it is symmetrical to the word return.

4. Function problems. Just as in the general case of Relation problems (Subsection 1.6, Page 69), there will be two kinds of **function problems** whose name acknowledge the fact that with functions, going from input to output is the privileged direction.

- A **direct problem** is a function problem in which we are given an input and we are looking for the output that the function may return.

EXAMPLE 1.26. In EXAMPLE 1.22 (Page 88), a *direct* problem might for instance be: What was the profit/loss in 1999?
Answer: -2000

reverse problem
input-output pair
I-O pair

EXAMPLE 1.27. In EXAMPLE A.18 (Page 511), a *direct* problem might for instance be:

$$\text{75 cents} \xrightarrow{\mathcal{JOE}} \mathcal{JOE}\left(\text{75 cents}\right) = y \text{ minutes}$$

that is, how many minutes parking time will $\mathcal{JOE}$ return for 75 cents ?

We will see that *direct* problems are generally relatively easy to solve.

EXAMPLE 1.28. Solving *direct* problem in the real world like figuring how much parking time will three quarters buy you is easy: if nothing else, just put three quarters in the parking meter and see how much parking time you get!

- A **reverse problem** is a function problem in which we are given an output and are looking for the input for which the function returns the given output.

EXAMPLE 1.29. In EXAMPLE A.18 (Page 511), a *reverse* problem might for instance be:

$$x \text{ cents} \xrightarrow{\mathcal{JOE}} \mathcal{JOE}\left(x \text{ cents}\right) = \text{50 minutes}$$

that is, how many cents should we input for $\mathcal{JOE}$ to return 50 minutes parking time?

EXAMPLE 1.30. In EXAMPLE 1.22 (Page 88), a *reverse* problem might for instance be: In what year(s) (if any) did the business return $+5\,000$? Answer: 1998, 2001, 2005.

Of course, neither direct problems nor reverse problems *need* have a solution.

EXAMPLE 1.31. In EXAMPLE 1.22 (Page 88),
- There is no Profit/Loss for Year 2000.
- There is no Year for which the Profit/Loss is 6000.

5. Input - Output pairs. Given a function, an input and an output, we will use the word **input - output pair**, **I - O pair** for short, when the function returns the given output for the given input.

We will continue to use in the case of input - output pairs the format which we introduced in Subsection 1.6 - Relation problems (Page 69), par-

input-output function format
I-O function format

ticularly for plotting purposes, but in the cae of functions we will also use another two formats:

"function" because of DEFINITION 2.1 - Local behaviour coding format (Page 135)

DEFINITION 1.2 The two **input-output formats** which say that x_0 and y_0 are related by a *function* f, are:

- For *computational* purposes, the equality $f(x_0) = y_0$
- For *conceptual* purposes, the arrow notation $x_0 \xrightarrow{f} y_0$

And of course, we can combine the two as: $x_0 \xrightarrow{f} f(x_0) = y_0$

EXAMPLE 1.32. In EXAMPLE A.18 (Page 511), we could have written:

- For *computational* purposes, the equality $\mathcal{JOE}(\text{25 cents}) = \text{10 minutes}$

or

- For *conceptual* purposes, the arrow notation $\text{25 cents} \xrightarrow{\mathcal{JOE}} \text{10 minutes}$

Of course, we can combine the two:

$$\text{25 cents} \xrightarrow{\mathcal{JOE}} \mathcal{JOE}\left(\text{25 cents}\right) = \text{10 minutes}$$

Usually, though, we will *not* include units in either inputs or outputs.

EXAMPLE 1.33. To say that -5 and $+6.75$ are related by the function $\mathcal{JILL}$, we can write

- For *computational* purposes, the equality $\mathcal{JILL}(-5) = +6.75$

or

- For *conceptual* purposes, the arrow notation $-5 \xrightarrow{\mathcal{JILL}} +6.75$

Of course, we can combine the two: $-5 \xrightarrow{\mathcal{JILL}} \mathcal{JILL}(-5) = +6.75$

6. I - O pair problems. In a I - O pair problem, we will be given both an input and an output and we will want information about the input-output pair, for instance whether or not the function returns the output or whether the input and the output have the same sign or whether the output is larger than the input, etc.

EXAMPLE 1.34. In EXAMPLE 1.22 (Page 88), we may ask
In 2002, did the business really return $+5000$?

As might perhaps have been expected, it is solving *reverse* problems (which, as we will see, is what 'solving equations' is all about) that matters most in the real world.

Input-Output plot
I-O plot
input level line
output level line

EXAMPLE 1.35. What we usually need to solve in the real world is, for instance, figuring how many quarters we need to get, say, two hours parking time.

5 Functions Given by I-O Plots

In keeping with our introduction in Subsection 4.2 of the words inputs and outputs instead of left number and right number in the case of functions,

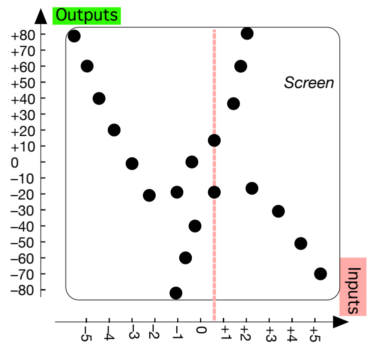
- instead of the words set of plot dots which we introduced in Subsection 3.5 for relations, in the case of functions we will use the word **Input-Output plot**, **I-O plot** for short, .
- instead of the words left number level line and right number level line which we introduced in Subsection 2.1 for relations, in the case of functions we will use the words **input level line** and **output level line**

1. Input - Output plots. Since functions are a special kind of relation, Input-Output plots can give functions but we need to restate the **??** (**??** **??**, **??**) in words of set of plot dots:

> **DEFINITION 2.1 (Restated) Local behaviour coding format**
> In order for a set of plot dots to give a function,
>
> No input level line shall intersect the set of plot dots more than *once*.
>
> that is, in other words,
>
> Any input level line shall intersect the set of plot dots at most *once*.

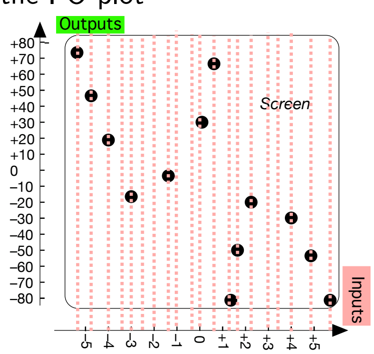
EXAMPLE 1.36. Given the set of plot dots

interpolate
intermediate plot dot



since there is at least one input-level line that does intersect the set of plot dots more than *once*, the set of plot dots *does not* give a *function*

EXAMPLE 1.37. Given the I-O plot



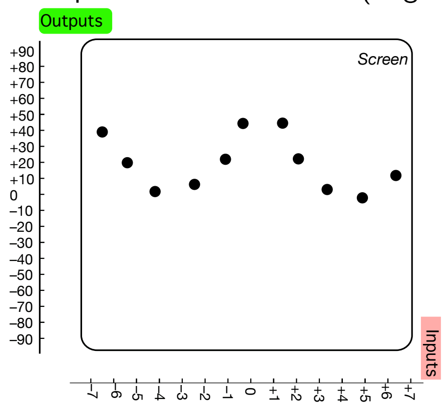
since *no* input-level line intersects the I-O plot more than *once*, the I-O plot *does* give a *function*

2. Interpolation As visual as I-O plot can be, a major difficulty with functions given by I-O plots is that sets of plot dots are sparse (CAUTIONARY NOTE 1.3, Page 80) so that functions given by an I-O plot cannot return any output for most inputs.

So, in many real world situations, one has to **interpolate** the I-O plot, that is somehow create **intermediate plot dots**. For instance, one can reset the plotter and make another run. The trouble, though, is that just about anything can happen with these intermediate plot dots:

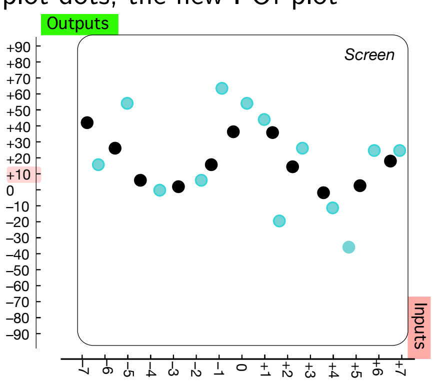
i. There is not even any guarantee that the interpolated I-O plot will still meet the Local behaviour coding format (DEFINITION (Restated) 2.1, Page 95).

EXAMPLE 1.38. The I-O plot in EXAMPLE 1.16 (Page 77),



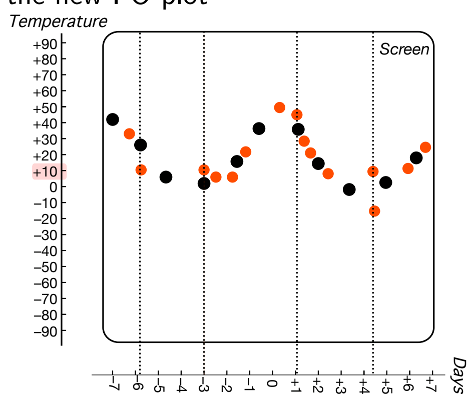
meets the Local behaviour coding format (DEFINITION (Restated) 2.1, Page 95) but:

while with the *blue* intermediate plot dots, the new I-Of plot



still gives a function,

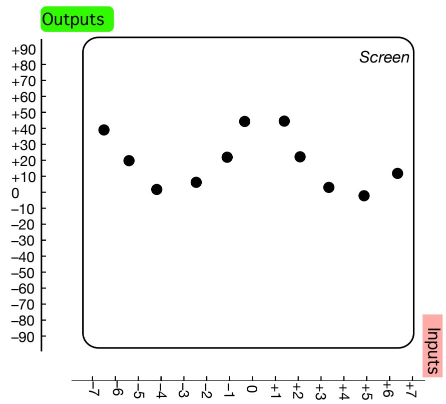
with the *red* intermediate plot dots, the new I-O plot



does *not* give a function.

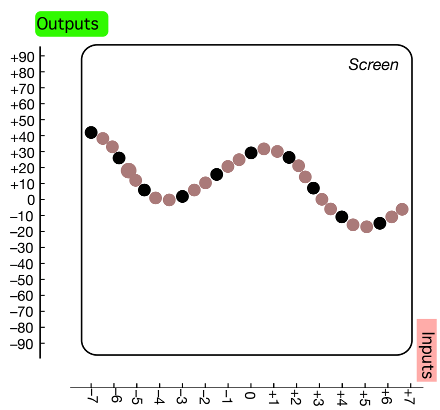
ii. Even wnen the interpolated I-O plot does give a function, that function can be just about *any* function

EXAMPLE 1.39. In the case of the I-O plot in EXAMPLE 1.16 (Page 77) declare

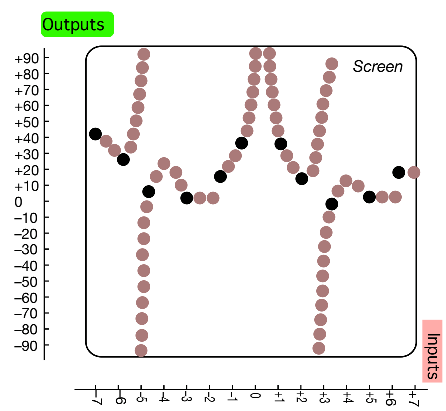


the following two interpolations both give a function but

While the intermediate plot dots could of course be:



the intermediate plot dots could just as well be:



iii. In fact, *how* to interpolate an I-O plot is not at all a simple matter and there are many methods for coming up with likely outputs for missing intermediate inputs. (`https://en.wikipedia.org/wiki/Interpolation`)

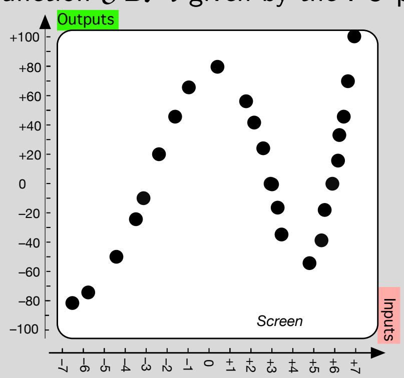
3. Direct function problems Keeping in mind that I-O plots are sparse:

i. When the given input is a number, solving direct problems goes exactly as with a *relation* given by an I-O plot and we just use, suitably rephrased, PROCEDURE 1.3 - Read a Plot dot (Page 76):

PROCEDURE 1.6 To get $f\left(x_0\right)$ for a given x_0 when f is given by an I-O plot,

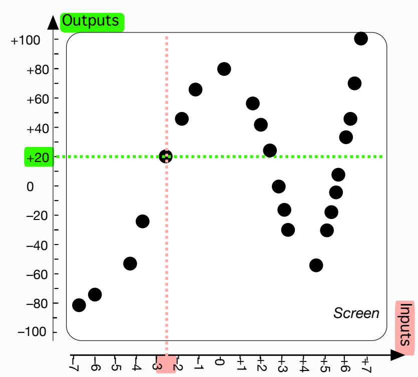
i. Tickmark x_0 on the input ruler,
ii. Draw an input level line through x_0,
iii. Mark the plot dot at the intersection, *if any*, of the input level line with the I-O plot,
iv. Draw an output level line through the plot dot (if any),
v. Read $f\left(x_0\right)$ where the output level line intersects the output ruler,

DEMO 1.6a With the function $\mathcal{JIM}$ given by the I-O plot



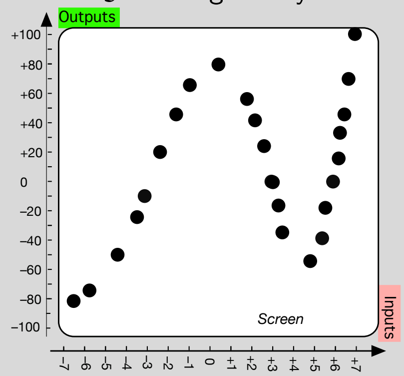
get $\mathcal{JIM}\left(-2.5\right)$

i. We tickmark -2.5 on the input ruler,
ii. We draw an input level line through the tickmark,
iii. We mark the plot dot at the intersection of the input level line with the I-O plot,
iv. We draw an output level line through the plot dot,
v. We read $\mathcal{JIM}\left(-2.5\right)$ where the output level line intersects the output ruler: $+20$



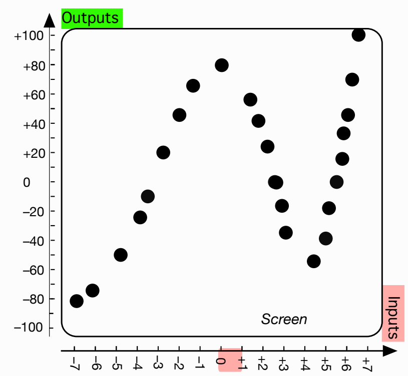
regular input

DEMO 1.6b With the function $\mathcal{GWEN}$ given by the I-O plot



get $\mathcal{GWEN}\left(+0.7\right)$

i. We tickmark $+0.7$ on the input ruler,
ii. We draw an input level line through the tickmark,
iii. There is no plot dot at the intersection of the input level line with the I-O plot,
iv. $\mathcal{GWEN}$ does *not* return any output for $+0.7$



A function given by an I-O plot cannot of course return ∞.

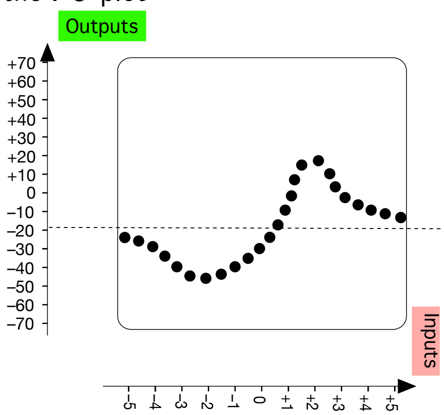
A **regular input** will be an input number for which the function returns an output *number*.

ii. When the given input is infinity, since inputs and outputs are mid-size numbers, an I-O plot cannot provide any information about the outputs for *large-size* input numbers.

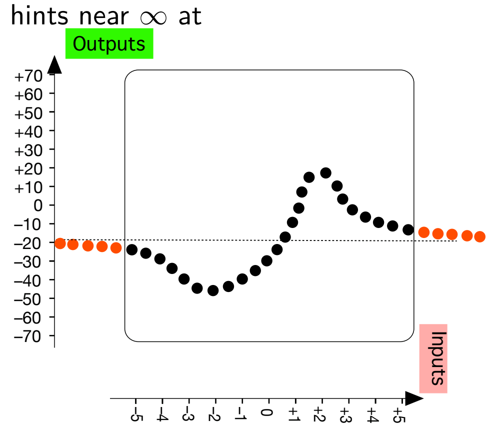
However, occasionally, the I-O plot can *hint* at what the function might return *near* infinity

EXAMPLE 1.40. It might seem that

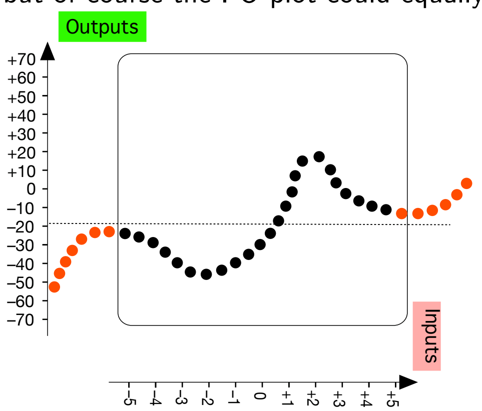
the I-O plot

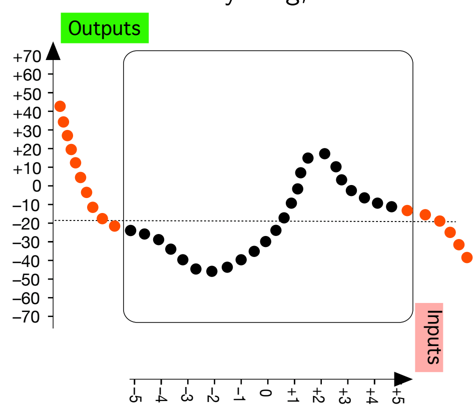


hints near ∞ at



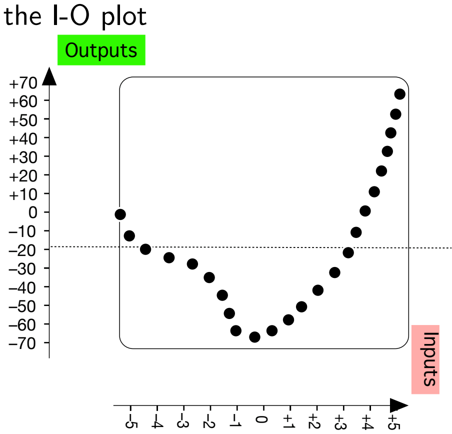
but of course the I-O plot could equally well be almost anything, for instance



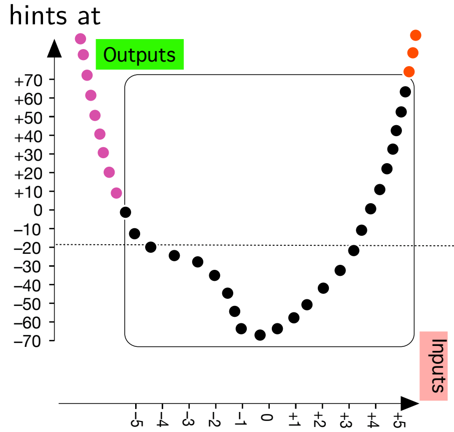


EXAMPLE 1.41. It might seem that, near ∞,

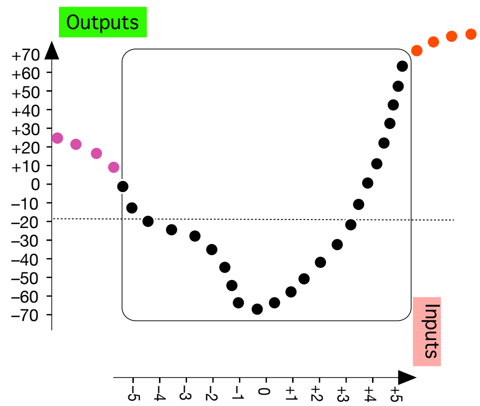
the I-O plot

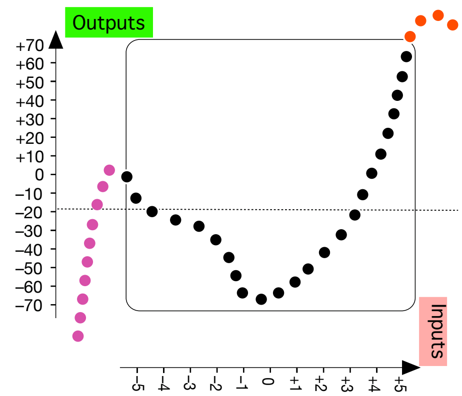


hints at



but of course the I-O plot could equally well be almost anything, for instance locate





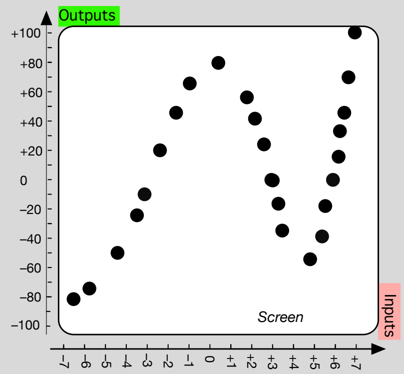
4. Reverse function problems For a given functions, a reverse problem is to **locate** the input(s) if any for which the function returns a given output

When the given output is a number, we just use, suitably rephrased, PROCEDURE 1.5 - right number for a left number (set of plot dots (Page 84):

PROCEDURE 1.7 To get x_0 for a given y_0 when f is given by an I - O plot

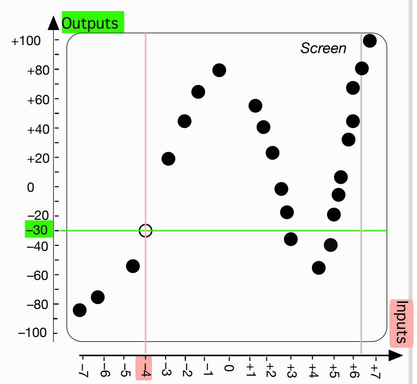
i. Tickmark y_0 on the output ruler,
ii. Draw an output level line through y_0,
iii. Mark the plot dot(s), if any, where the output level line intersects the I - O plot
iv. Draw an input level line through each marked plot dot,
v. Read x_0 where the input level line(s) intersect the input ruler.

DEMO 1.7a Get the input(s), if any, for which the function $\mathcal{RON}$ given by the I-O plot

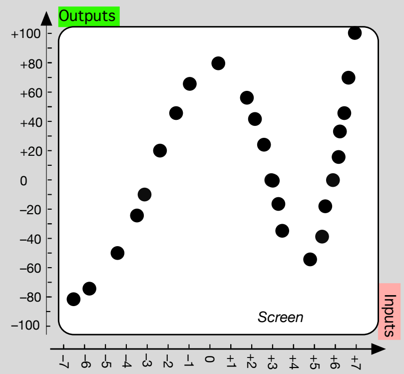


returns the output -30.

i. We tickmark the output number -30 on the *output ruler*,
ii. We draw an output level line through the tickmark,
iii. We mark the plot point(s), if any, at the intersection of the output level line with the I-O plot
iv. We draw an input level line through the plot dot(s), if any,
v. The input number(s), if any, is/are at the intersection(s), if any, of the input level line(s), if any, with the input ruler: -4

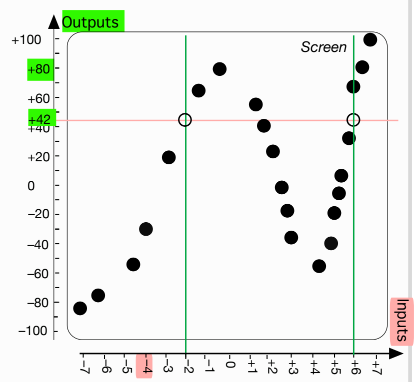


Demo 1.7b Get the input(s), if any, for which the function $\mathcal{MAE}$ given by the I-O plot



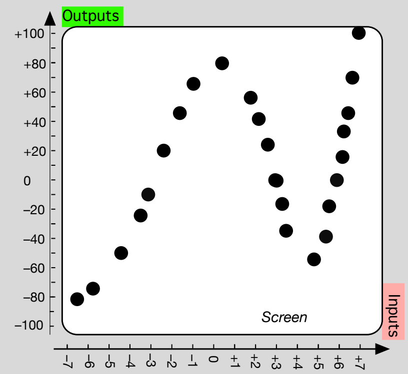
returns the output -30.

i. We tickmark the output number -30 on the *output ruler*,
ii. We draw an output level line through the tickmark,
iii. We mark the plot point(s), if any, at the intersection of the output level line with the I-O plot
iv. We draw an input level line through each plot dot(s), if any,
v. The input number(s), if any, is/are at the intersection(s), if any, of the input level line(s), if any, with the input ruler: -4, $+3$, $+5$



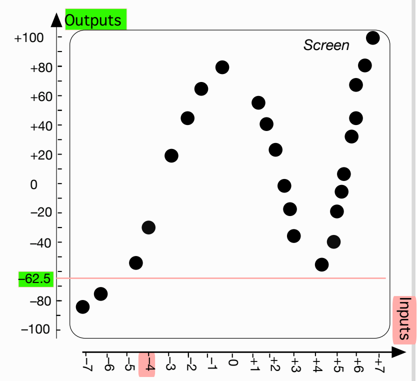
DEMO 1.7c Get the input(s), if any, for which the function $\mathcal{SALLY}$ given by the I-O plot

zero



returns the output -62.5.

i. We tickmark the output number -62.5 on the *output ruler*,
ii. We draw an output level line through the tickmark,
iii. We mark the plot point(s), if any, at the intersection of the output level line with the I-O plot
iv. There is no intersection therefore there is no input level line through the plot dot(s), if any,
v. The input number(s), if any, is/are at the intersection(s),therefore there is no input number.



5. Zeros. The fact that reverse problems usually have in fact no solution because I-O plots are sparse is particularly unfotunat when we are looking for the **zero(s)** of a given function, that is for the inputs whose output is 0.

Zeros will be important because, as we will see, inputs whose output is 0 often separate inputs whose output is positive from inputs whose output is negative.

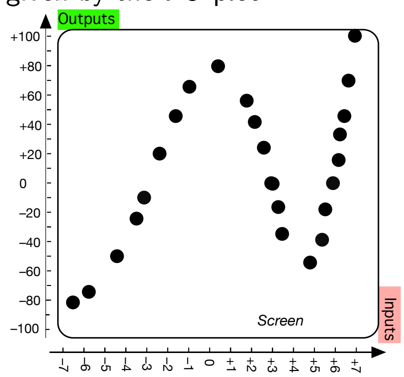
And, even though 0 is a *dangerous* number (Cautionary Note 0.2, Page 5), a zero is a regular input.

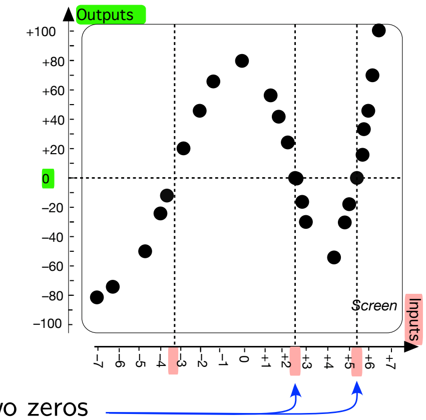
However, with functions given by I-O plots, we will have to keep even more seriously in mind that **??** (**??** **??**, **??**).

Example 1.42. The function $\mathcal{EMMY}$

pole

given by the I-O plot





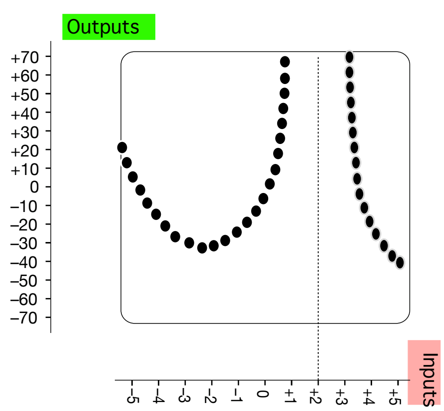
has two zeros

but it certainly looks like $\mathcal{EMMY}$ also has a zero between -3 and -4

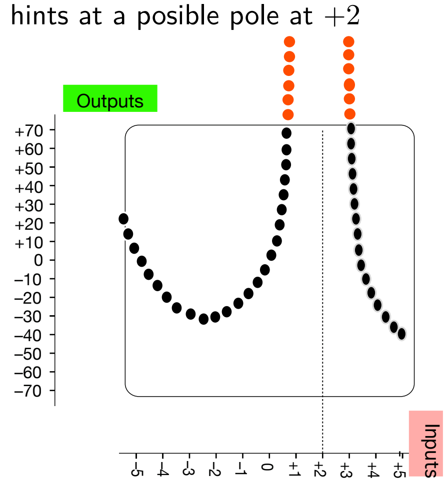
6. Poles. An actually even more important reverse problem will be locating the **pole(s)** if any, of a function, that is those inputs for which the function returns ∞.

Of course, a pole is *not* a regular input since a function given by a I-O plot cannot have pole(s) since all the outputs are medium-size numbers. Yet, I-O plots can hint at possible pole(s).

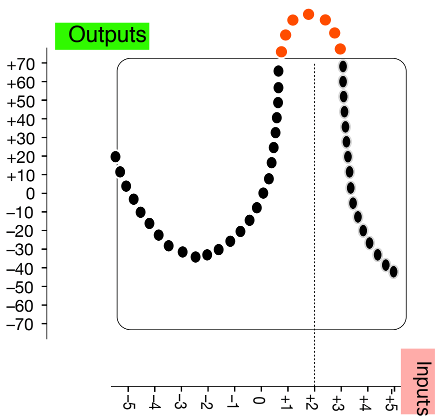
EXAMPLE 1.43. It might seem that the I-O plot

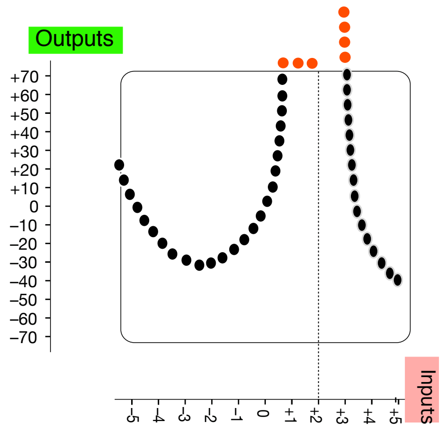


hints at a posible pole at $+2$



but of course the I-O plot could equally well be almost anything, for instance





discrete function curve

7. Discrete Calculus. In this text, though, we discussed functions given by I-O plots only for introductory purposes and, from now on, functions will *not* be given by I-O plots anymore.
Nevertheless,

> **LANGUAGE NOTE 1.6** Functions given by I-O plots are a particular kind of functions usually called **discrete functions**.

And in fact, the DISCRETE CALCULUS, that is the calculus of discrete functions is a very important piece of MATHEMATICS. (https://en.wikipedia.org/wiki/Discrete_calculus)

6 Functions Given By Curves

Functions given by I-O plot involved only *medium-size* numbers but functions given other than by I-O plots will also involve both

- small-size numbers.

and

- large-size numbers,

So, we will not be able to use *quantitative* Cartesian setups any more than we could use quantitative rulers back in Section 12* - Real Numbers (Page 55).

1. Qualitative Cartesian setup. In order to use large-size numbers and small-size numbers as well as medium-size numbers, we will then:

▶ Give functions with **curves** instead of with I-O plot,

qualitative Cartesian setup
offscreen
global graph
onscreen graph
solid line
offscreen graph
dotted line
graph dot
Mercator view

Don't worry, you don't have to know the calculus meaning of smooth curve and go by just the everyday meaning but you can always look-up `https://en.wikipedia.org/wiki/Curve`

and

- Use **qualitative Cartesian setups** that is Cartesian set ups in which:
  - The screen is within a surrounding area we will call **offscreen**,
  - The input ruler is a *qualitative* ruler,
  - The output ruler is a *qualitative* ruler.

Then, a curve that satisfies the **??** (**?? ??**, **??**) will give a function

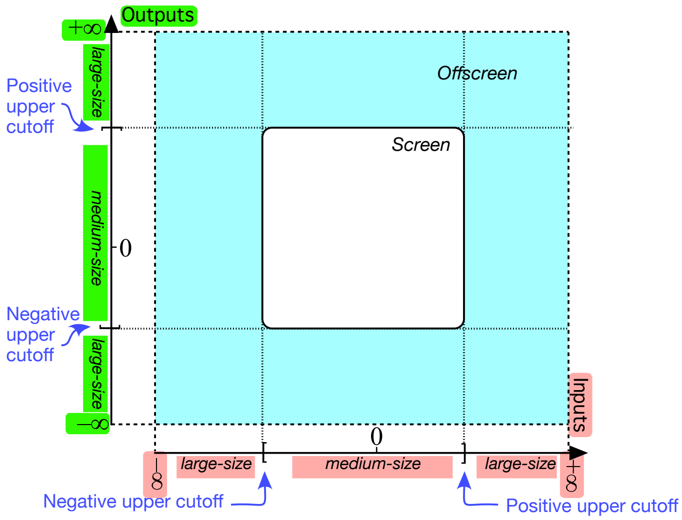
- whose **global graph** is the given curve,
- whose **onscreen graph** is the part of the global graph which is *on* the screen. We will picture the onscreen graph with a **solid line**.
- whose **offscreen graph** is the part of the global graph which is *not on* the screen. We will picture the offscreen graph with a **dotted line**.

Also, in the case of functions given by curves, we will use the word **graph dot** instead of the word plot dot.

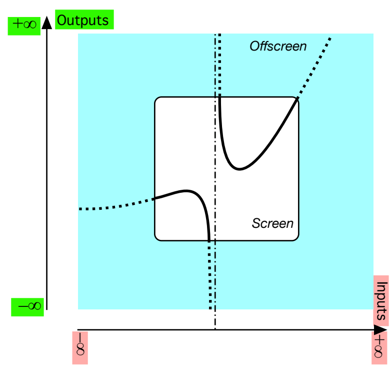
Since our purpose in this Part I - Functions Given By Data (Page 63) is *introductory*, we will use curves to *give* functions but eventually we will want curves to *picture* functions that will have been given otherwise in Part II - Functions Given By Rules (Page 207) and after. In any case,

> **Cautionary note 1.7** Functions given by curve are not necessarily simple and certainly not as simple as those used here.

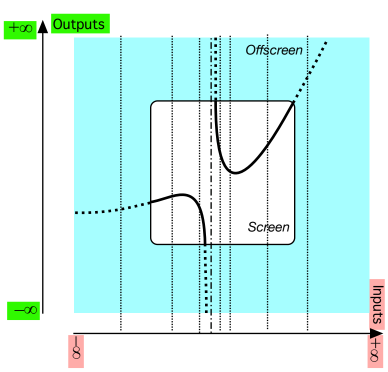
2. Mercator view. By far the simplest way to view a qualitative Cartesian setup is by way of a **Mercator view** which is a flat view that shows the 2 pt compactification of each ruler:



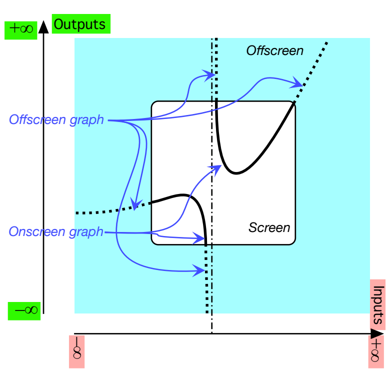
EXAMPLE 1.44. The curve



satisfies the ?? (?? ??, ??):



and so it is the global graph of a function:



conclusive

3. Mercator views are *not* conclusive. But even though the Mercator view is by far the most commonly used, it is important to be aware of the severe limitations to the information which Mercator views can provide about a function as the Mercator view shows mostly the onscreen graph and therefore depends *very much* on the cutoff sizes for medium-size numbers.

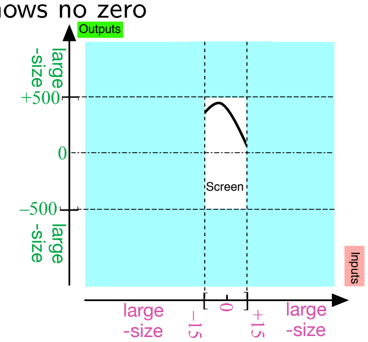
The problem is a difficult one and Mercator's solution, https://en.wikipedia.org/wiki/Mercator_projection, was the first in a long list: https://en.wikipedia.org/wiki/List_of_map_projections

i. How much an *onscreen* graph shows about a function depends *very much* on the cutoff size for medium-size *input* numbers.

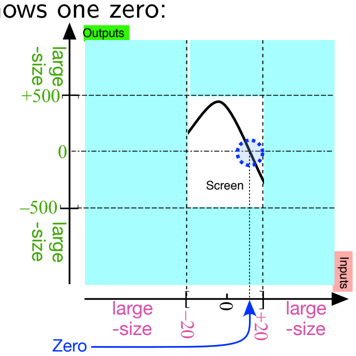
For instance, Mercator views do not necessarily show all the zeros of a function.

EXAMPLE 1.45. The following *onscreen* graphs of the function $\mathcal{ZANY}$ are all at the same scale and differ only by the cutoff size for medium-size *input* numbers:

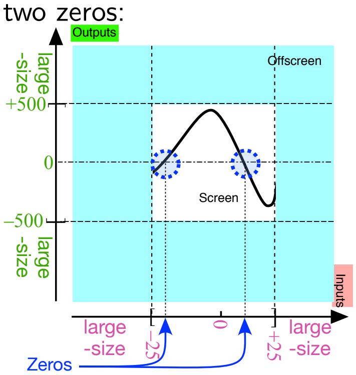
With medium-size *input* numbers cutoff at 15, the onscreen graph shows no zero



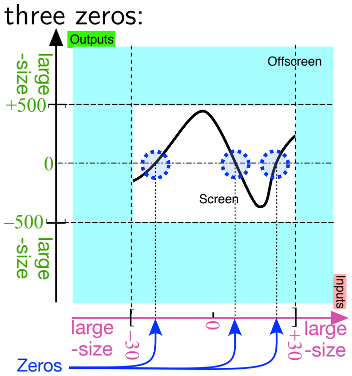
With medium-size *input* numbers cutoff at 20, the onscreen graph shows one zero:



With medium-size *input* numbers cutoff at 25, the onscreen graph shows two zeros:



With medium-size *input* numbers cutoff at 30, the onscreen graph shows three zeros:



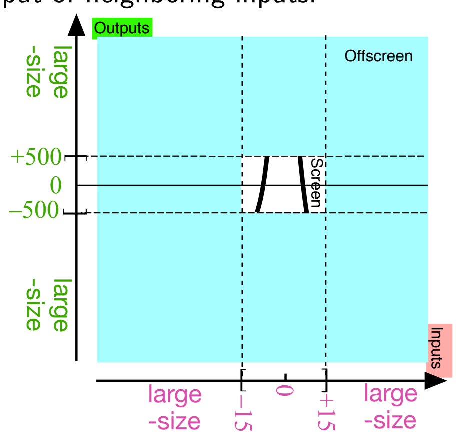
In other words, the Mercator views of a given function are *not* **conclusive** as to the zeros of that function.

ii. How much an *onscreen* graph shows about a function depends also *very much* on the cutoff size for medium-size *output* numbers.

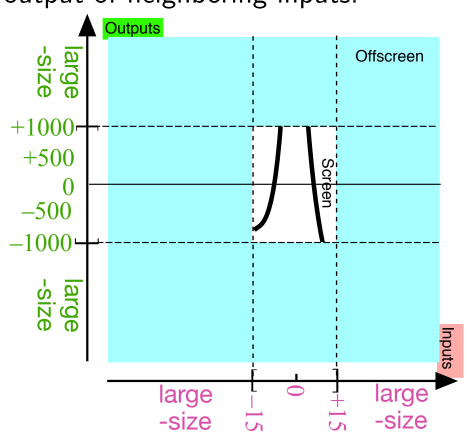
For instance, another very important reverse problem will be locating the pole(s), if any, of a function, that is those inputs for wich the function returns ∞ but of course Mercator views cannot do that.

EXAMPLE 1.46. The following onscreen graphs of the function $\mathcal{COTY}$ are all at the same scale and differ only by the cutoff size for medium-size *output* numbers:

With medium-size *output* numbers cutoff at 500, the onscreen graph does not show whether or not there is an input between -15 and $+15$ whose output is larger than the output of neighboring inputs:

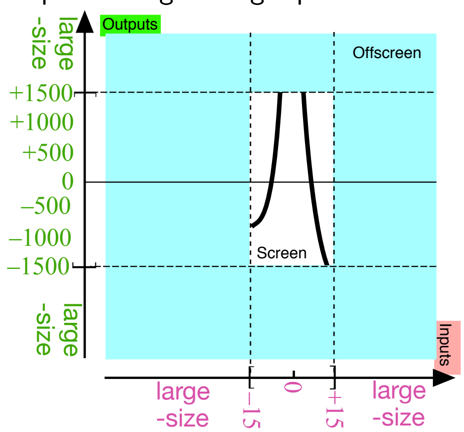


With medium-size *output* numbers cutoff at 1000, the onscreen graph still does not show whether or not there is an input between -15 and $+15$ whose output is larger than the output of neighboring inputs:

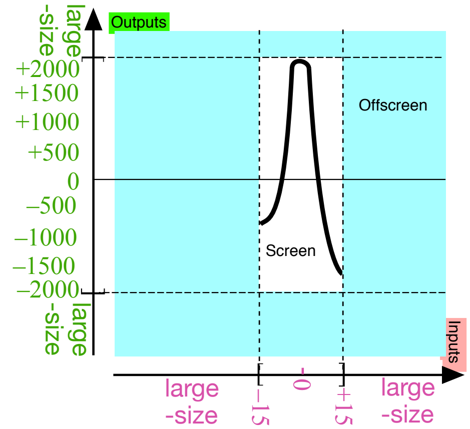


tube view

With medium-size *input* numbers cutoff at 1500, the onscreen graph still does not show whether or not there is an input between -15 and $+15$ whose output is larger than the output of neighboring inputs:



With medium-size *input* numbers cutoff at 2000, the onscreen graph does show there is an input between -15 and $+15$ whose output is larger than the output of neighboring inputs:



In other words, the Mercator views of a given function are *not* necessarily conclusive as to the inputs whose output is larger than the output of nearby inputs.

Altogether then:

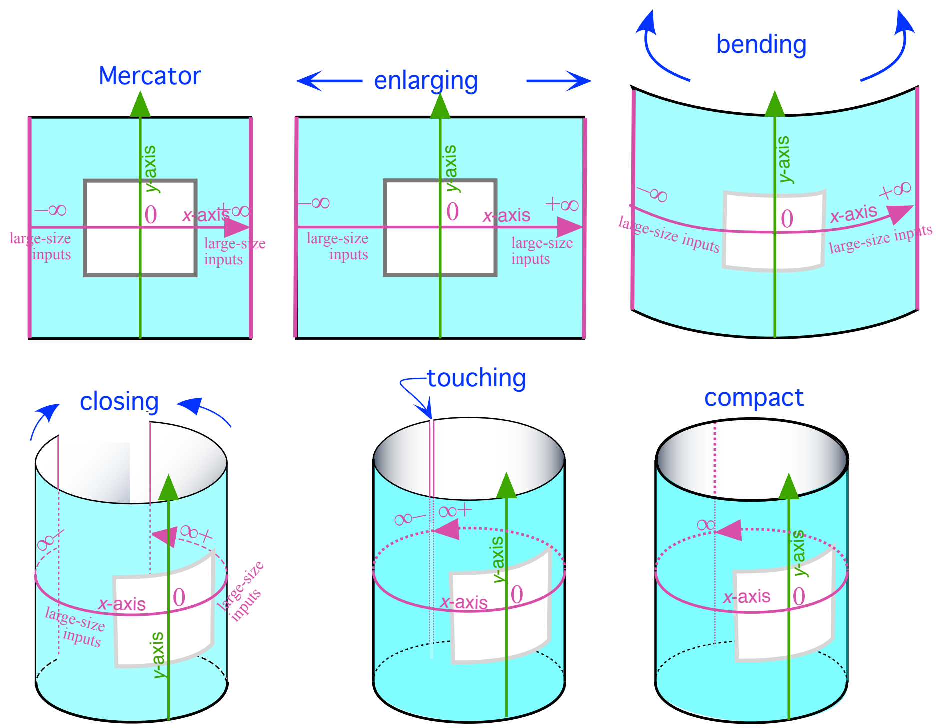
CAUTIONARY NOTE 1.8 On-screen graphs are *not necessarily* **conclusive** as to the output(s), if any, for medium-size inputs.

To see why axes rather than rulers, just try to draw rulers *in any of the following compact views!*

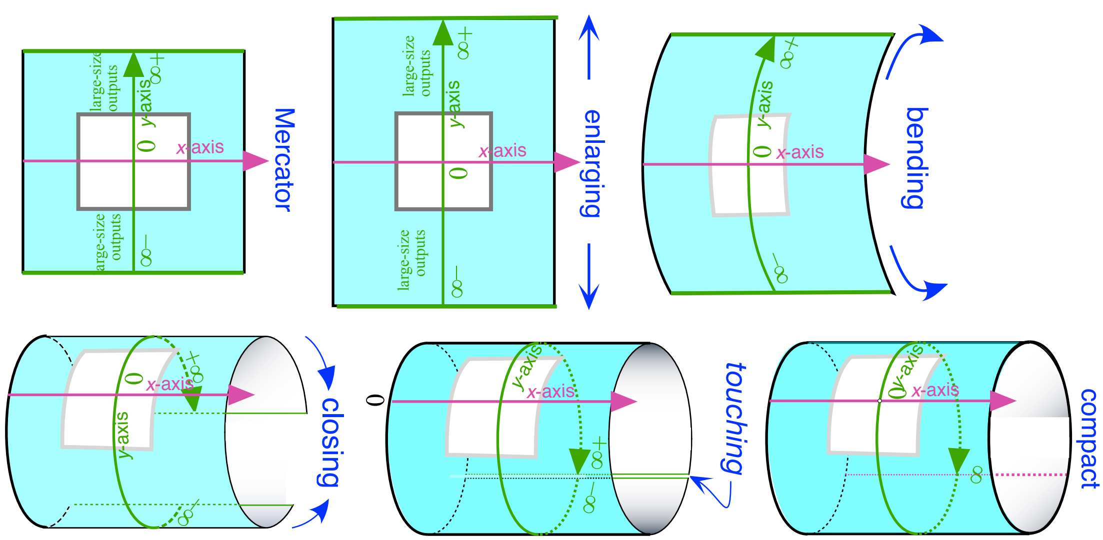
4. Compact views. In order to see the off-screen graph which shows the 'behavior' of a function for large-size *inputs* and for pole(s), if any, we need to use one-point compactifications of the *axes* instead of using rulers.

i. We can get a **tube view** by compactifying the *input* axis:

donut view

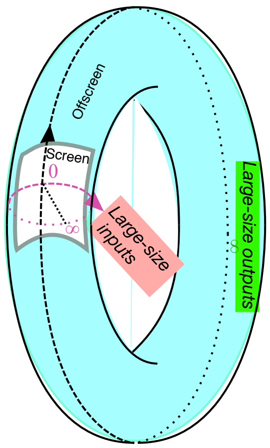


ii. We can another kind of tube view by compactifying the *output* axis:

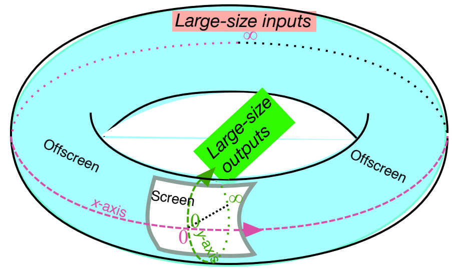


iii. We can get two kinds of **donut views** by compactifying the input axis and the output axis *one after the other*:

Magellan view

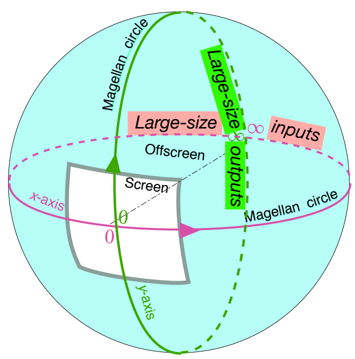


Input axis then output axis



Output axis then input axis

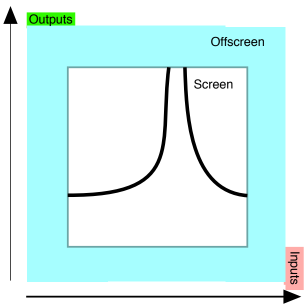
iv. We can get a **Magellan view** by compactifying the input axis and the output axis *simultaneously*:



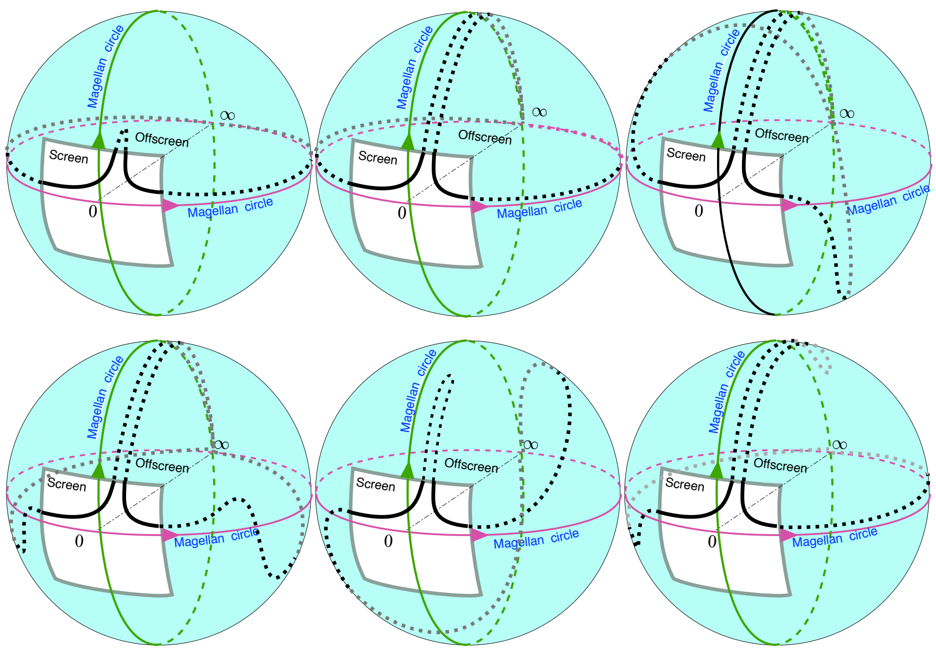
5. Compact views *are* conclusive. Magellan views are particularly good at showing why a Mercator view cannot *give* a function: different functions can have the same onscreen graph but different off-screen graphs.

EXAMPLE 1.47. The *onscreen* graph

smooth continuation



is the onscreen graph of any of the following functions viewed in Magellan view

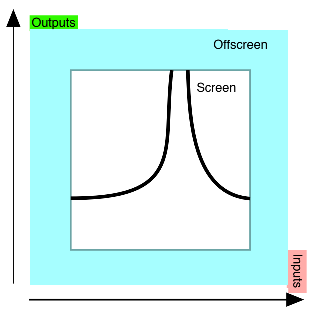


as well as, in fact, many, many others.

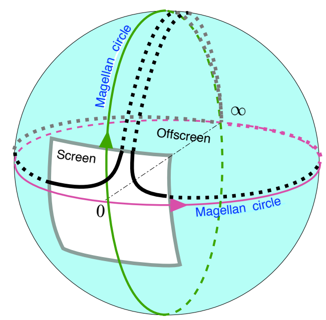
So, in order for an onscreen graph to be able to give a function, we will make the following

AGREEMENT 1.3 With functions given by curve, the upper cutoff sizes for medium-size inputs and medium-size outputs will be such that the off-screen graph is simply a **smooth continuation** of the onscreen graph. (However, with other types of functions, there are different kinds of continuations as, for instance, with the 'periodic' functions investigated in VOL. II.)

EXAMPLE 1.48.
Given the onscreen graph in EXAMPLE 2.37 (Page 160):



by AGREEMENT 1.3 - (Page 115), the global graph can only be



7 "Simple" Functions?

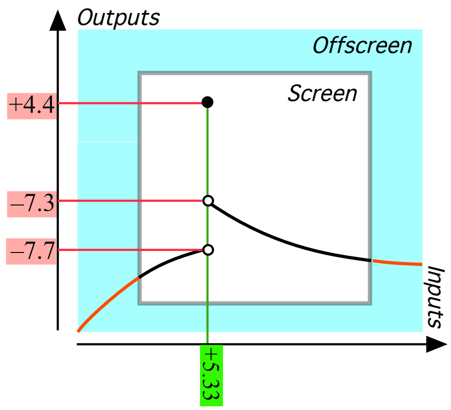
Aaargh!

=======Begin HOLDING=======

As we saw in Subsection 8.2 - Output level band (Page 121) and xxx, the information provided by the plot dot for an input need not *necessarily* extend to even just *nearby* inputs.

For instance, we might expect that the outputs for inputs near a given input will have outputs that are near the output for the given input but, while this is often the case, this is absolutely not *necessarily* the case.

EXAMPLE 1.49. The function given by the global graph



- Sends +5.33 to a *positive* number, +4.4

but

- Sends *all* other numbers to *negative* numbers.

=======End HOLDING =======

So far, the reader would have every right to think that functions are something essentially fairly "simple" but, not only are there many functions whose 'behavior' is unimaginably 'complicated', it is impossible to draw the line and define anything like "simple" functions.

Mathematicians and scientists keep being amazed at the behavior of some of the functions which have recently come up in mathematics and the sciences.

Basically, the difficulty is that there is nothing in what we have said so far about functions, including the functional requirement, to prevent abrupt, even wild, changes in the outputs for nearby inputs,

The use of nearby inputs instead of the given input raises a most important question: To what extent are the nearby outputs (outputs for *nearby* inputs) *all* near the output *at* the given input? And, as it turns out, the question has no simple answer. So, as a backdrop to the functions which we will investigate in this text, we will just illustrate some of the many different possible answers.

https://en.wikipedia.org/wiki/Extended_real_number_line. https://math.stackexchange.com/questions/354319/can_a_function_be_considered_heightcontinuous_if_it_reaches-infinity-at-one-point

This poses a most vexing expository problem inasmuch as making general statements about functions becomes extremely difficult ... and dangerous in that we may end up stating on the basis of something true for 'nice' functions something that would be false for these unimaginably 'complicated' functions. More precisely,

- If general statements about functions are worded so as to apply to really *all* functions, including 'complicated' functions—which the reader is not likely to encounter anytime soon, the reader is going to have a very hard time—for no good reason,

and

- If general statements about functions are worded so as to apply *only* to

simple
smooth

'simple' functions, how is the reader to know when the functions have become too 'complicated' for the general statement to height continue to apply?

EXAMPLE 1.50. It is fairly intuitive that plot dots should make up some probably curved line. But, while this is indeed the case for many functions, it is not *necessarily* the case and in fact *not* the case, for most functions.

One way out would be to define, say, Type A functions and then Type B functions and then Type C functions, etc and to make general statements for each type of functions as we go. This of course would work but would force us to restate a lot as we go from Type A functions to Type B functions, and then again as we go to Type C functions, etc. Another drawback aside from the hassle of having to keep restating, is that this tends to lose the bigger represent and there is a price to that too.

Nevertheless, one look at the TABLE OF CONTENTS of this text will show that this is indeed what we will do in the following chapters *but*, before that, if only for the sake of not having to repeat things and of the bigger represent, we will spend the rest of *this* chapter discussing the possible behaviors of 'nice' functions.

So, even though we cannot *define* 'simple' functions and we cannot even pin down some of the things it would mean for a functions to be 'simple' so as to prevent general statements from applying to complicated functions,

> **AGREEMENT 1.4** will be, by the sole fact that they appear in this text, guaranteed to be **simple** functions and the general statements we will make in this text are guaranteed to apply to these functions. (Which does not imply that these general statements apply *only* to these functions.)

Roughly, **smoothness** extends to slope and concavity the requirements that height continuity made on the height namely that slope and concavity should not change abruptly. There is a big difference though:

- In the case of height continuity, we need to look at what happens *at* the given input and then to what happens *near* the given input but only to see if there is a jump and not even when there is a gap at x_0.
- In the case of slope and concavity, on the other hand, even with local graphs, neither slope nor concavity makes sense *at* the given input and what matters is only what happens *near* the given input.

isolated input
input level band

CAUTIONARY NOTE 1.9 Most unfortunately, the *usual* mathematical concept of smoothness implies height continuity which is not the way we think of smoothness in the real world.

For that matter, educologists well know that, in order to define smoothness at x_0 in the usual way one needs room in which to have a limit.

EXAMPLE 1.51. A PVC sewer and drain pipe is usually perceived as being "smooth" regardless of whether or not it is solid or perforated and a smoothly bending copper pipe doesn't stop being so if and when it develops a pinhole.

So, in this text and in trying to *represent* smoothness, *we* will go by $f(x_0 + h)$ and not pay any attention to $f(x_0)$.

https://en.wikipedia.org/wiki/Smoothness.
https://en.wikipedia.org/wiki/Analytic_function
https://en.wikipedia.org/wiki/Singularity_(mathematics)
https://en.wikipedia.org/wiki/Nowhere_heightcontinuous_function
https://en.wikipedia.org/wiki/Weierstrass_function
https://en.wikipedia.org/wiki/Fractal_curve

8 Local graph near a point

The main reason *we* will be dealing with functions given by curves is that, in contrast with functions given by I-O plot, in the case of functions given by curves we will be able to look at the *neighborhoods* of the inputs instead of having to deal with **isolated inputs** inasmuch as sets of plot dots are sparse (CAUTIONARY NOTE 1.3, Page 80).

But we will first need to do a bit of preparatory work, and, in this section, we will introduce in the case of functions given by curves, several different kinds of things that, in fact, we will also use later on with functions given otherwise.

In other words, in the case of functions given by curves, we will have "elbow room"!

1. Input level band. In order to observe the 'behavior' of a function *near* a given point, be it a number x_0 or ∞, we will of course need to fatten the point into a neighborhood of that point bur we will also need to fatten the input level *line* into an **input level *band***, that is a band centered on the input level line whose width is equal to the width of the neigborhood.

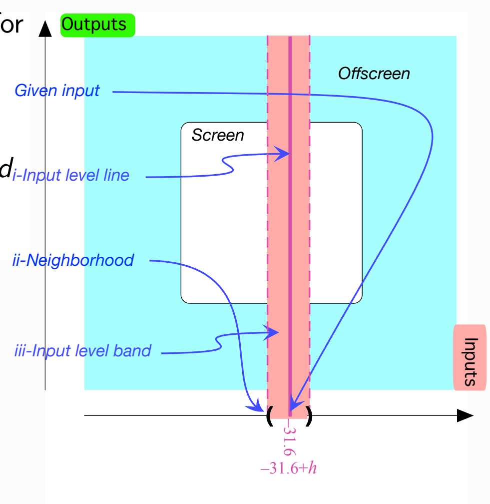
The details of the PROCEDURE, though, depend on whether the given input point is a number x_0 or is ∞:

PROCEDURE 1.8 To get the Input level band for a neighborhood of a given point.

- When the given point is a number x_0:
  - i. Draw the input level line for x_0,
  - ii. Fatten x_0 into a neighborhood of x_0,
  - iii. Fatten the input level line for x_0 into an input level band for the *neighborhood* of x_0
- When the given point is ∞:
  - i. Draw the input level lines for $+\infty$ and $-\infty$
  - ii. Fatten ∞ into a neighborhood of ∞ (In Mercatot view),
  - iii. Fatten the input level lines for $+\infty$ and $-\infty$ into rectangles corresponding to the width of the *half neighborhoods* of $+\infty$ and $-\infty$

DEMO 1.8a To get the input level band for a neighborhood of the input number -31.6

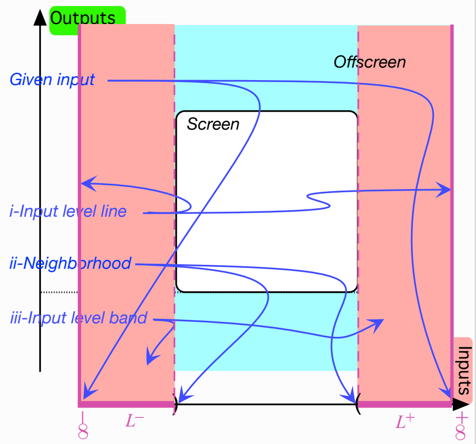
i. We draw the input level *line* for -31.6
ii. We mark a *neighborhood* of -31.6 on the *input ruler*,
iii. We draw the *input level band* as a rectangle with the width of the *neighborhood* of -31.6,



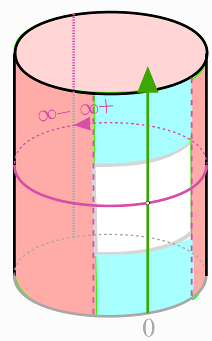
DEMO 1.8b To get the input level band for a neighborhood of the input ∞

output level band

i. We draw the input level *lines* for $+\infty$ and $-\infty$
ii. We fatten ∞ into a neighborhood of ∞ (In Mercatot view),
iii. We fatten the input level lines for $+\infty$ and $-\infty$ into rectangles corresponding to the width of the *half neighborhoods* of $+\infty$ and $-\infty$



In the above Mercator view, there appears to be *two* level bands for ∞ but a tube view shows they are only the two sides of the input level band near ∞:



2. Output level band. When we fatten a given output *point*, be it a number y_0 or ∞, into a *neighborhood*, we must also fatten the output level *line* for y_0 or ∞ into an **output level *band*** for the *neighborhood* of the point, y_0 or ∞, that is a rectangle corresponding to the width of the *neighborhood* of the point.

The details of the PROCEDURE, though, depend on whether the given output point is a number y_0 or ∞:

PROCEDURE 1.9 To get the output level band for a given output point.

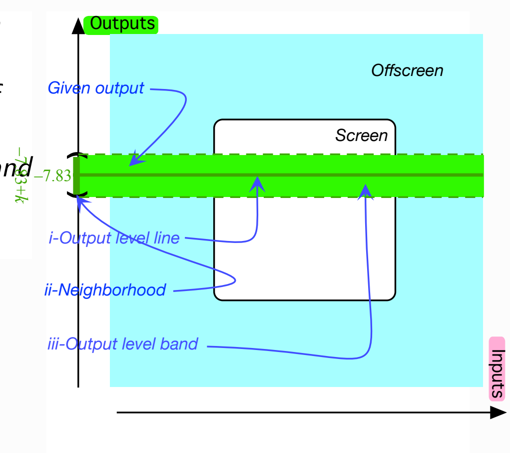
- When the given output point is a number y_0:
  i. Draw the output level line for y_0,

ii. Fatten y_0 into a neighborhood of y_0,
iii. Fatten the output level line for y_0 into an input level band for the *neighborhood* of y_0

- When the given output point. is ∞:
  i. Draw the output level lines for $+\infty$ and $-\infty$,
  ii. Fatten ∞ into a neighborhood of ∞ (In Mercaator view),
  iii. Fatten the output level lines for $+\infty$ and $-\infty$ into rectangles corresponding to the width of the half *neighborhoods* of $+\infty$ and $-\infty$

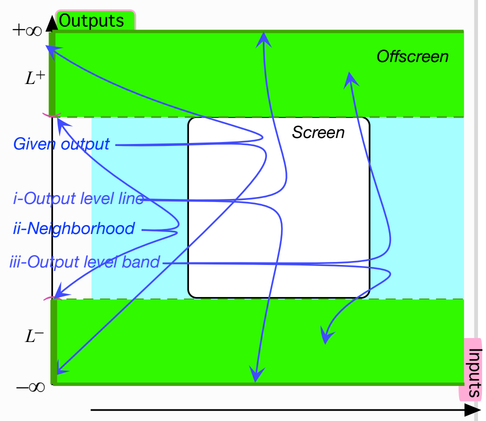
DEMO 1.9a
To get the output level band for a neighborhood of the output number -7.83

i. We draw the output level *line* for -7.83
ii. We fatten a *neighborhood* of -7.83 on the *output ruler*,
iii. We draw the *output level band* as a rectangle with the width of the *neighborhood* of -7.83,

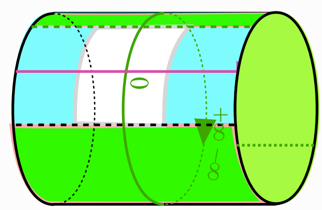


DEMO 1.9b
To get the output level band for a neighborhood of the output point ∞

i. We draw the output level *lines* for $-\infty$ and $+\infty$
ii. We fatten ∞ into a *neighborhood* in Mercator view,
iii. We fatten the output level *lines* for $-\infty$ and $+\infty$ into rectangles the width of the half neighborhoods of $-\infty$ and $+\infty$



In the above Mercator view, there appears to be *two* level bands for ∞ but a tube view shows they are only the two sides of the level band near ∞:



3. A Few Words of Caution Though. Starting with Part II - Functions Given By Rules (Page 207) though, functions will cease to be given by a global graph and will be given instead by an I-O rule

When a function will be given by an I-O rule instead of a global graph, though, we will have to be very careful before we use **??** because

In Subsection 8.4 - Frames (Page 124) we discussed how to get a local graph when the function is given by a curve. When the function is given by an I-O rule, though, we start out with no global graph, though, and getting a local graph is much more complicated and will require the knowledge of the global graphs of 'power functions'.

Since $x_0 \oplus h$ is a fattening of x_0, it is most tempting and natural to think of $f\left(x_0 \oplus h\right)$ as a fattening of $f\left(x_0\right)$ but, even though it is "often" the case, unfortunately

mostly the case in CALCULUS ACCORDING TO THE REAL WORLD *texts* that $f\left(x_0 \oplus h\right)$ is a neighborhood of some output number, be it $f\left(x_0\right)$ or some other output number y_0 so that one can fatten the output level line frame

into an **output level band**

Not even in the privacy of the reader's mind!

CAUTIONARY NOTE 1.10 One should absolutely *never* use the words "neighboring outputs" as a short for outputs for neighboring inputs because the output numbers $f\left(x_0 \oplus h\right)$ returned by the function f for $x_0 \oplus h$, that is for the input numbers in a neighborhood of x_0, need *not* make up a neighborhood of *any* output number y_0, let alone make up a neighborhood of the output number $f\left(x_0\right)$

.

EXAMPLE 1.52. In EXAMPLE 1.29, even though the inputs 27.2 and 27.4 can be considered to be near, their outputs, respectively around $+70$ and -25, certainly cannot be considered anywhere near each other.

4. Frames. However, just like the plot dot for an *ordinary* input x_0, that is for an input-output pair of *numbers* $\left(x_0, y_0\right)$, is at the intersection of:

- the input level *line* for the input *number* x_0
- the output level *line* for the output *number* y_0,

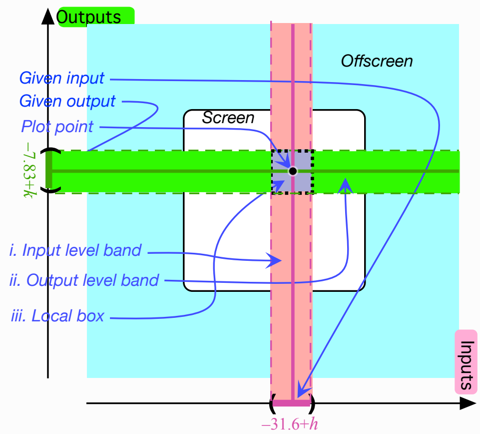
similarly, the local graph for a neighborhood of a point will be within the **frame** which is the border of the intersection of the input level band and the output level band:

PROCEDURE 1.10 To get the frame for an (point, point):
$\left(x_0, y_0\right)$ or $\left(x_0, \infty\right)$ or $\left(\infty, y_0\right)$ or (∞, ∞)

i. Get the input level band for x_0 or ∞
ii. Get the output level band for y_0 or ∞
iii. Frame the intersection of the input level band and the output level band

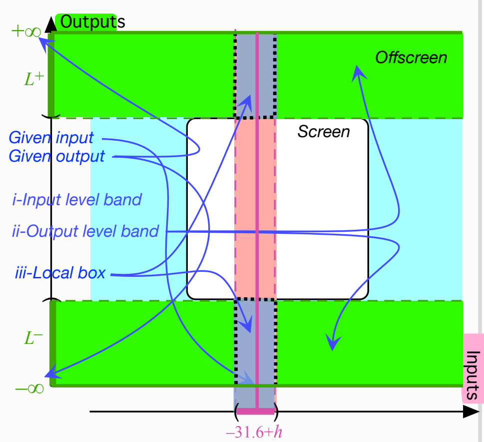
DEMO 1.10a
To get the frame for the input-output pair $\left(-3.16, -7.83\right)$

i. We get the input level *band* for -3.16
ii. We get the output level *band* for -7.83
iii. We *frame* the intersection of the input level bands for -3.16 and -7.83



Demo 1.10b
To get the frame for the input-output pair $\left(-3.16, \infty\right)$

i. We get the input level *band* for -31.6
ii. We get the output level *band* for ∞
iii. We frame the intersection of the input level band for -31.6 and the output level band for ∞

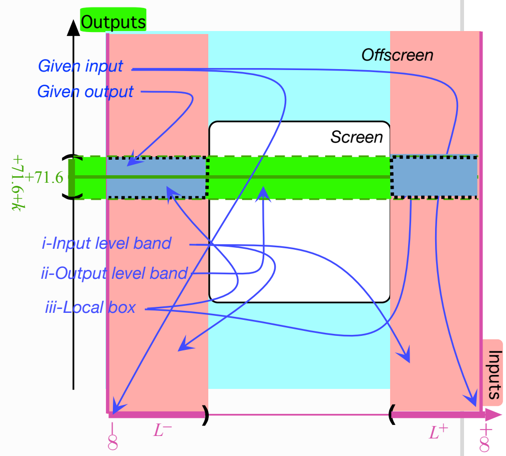


In the above Mercator view, there appears to be two frames for ∞ but a donut view shows they are only the two halves of the same frame.

But then Magellan views are a lot harder to draw.

Demo 1.10c
To get the frame for the input-output pair of *numbers* $\left(\infty, +71.6\right)$

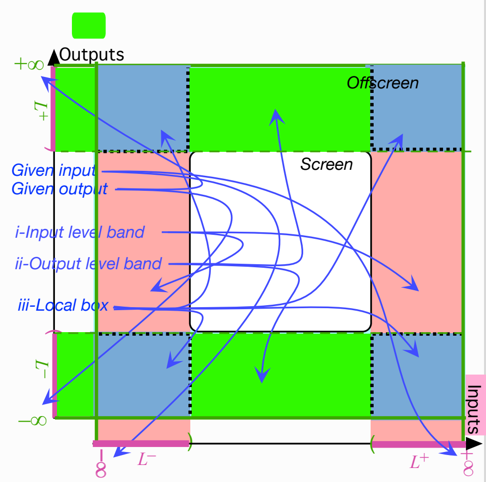
i. We get the input level *band* for ∞
ii. We get the output level *band* for $+71.6$
iii. We *frame* the intersection of the input level band for ∞ and the output level band for $+71.6$



In the above Mercator view, there appears to be two frames for ∞ but a donut view shows they are only the two halves of the same frame.

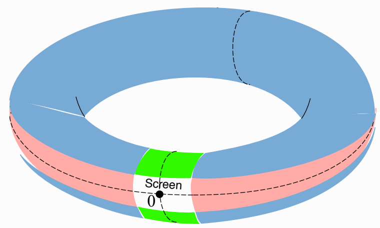
DEMO 1.10d
To get the frame for (∞, ∞)

i. We get the input level *band* for ∞
ii. We get the output level *band* for ∞
iii. We *frame* the intersection of the half input level bands for ∞ and the half input level bands for ∞



local graph

What appears to be four frames are actually parts of *the* frame because we are using a Mercator view instead of a Magellan view in which they would appear as the four quarters of a single frame.



5. zzzzzzzzzzzzzz Just the way a plot dot shows the inpui-output pair for a given input *number*, a **local graph** will show the inpui-output pairs for the input *numbers* in a *neighborhood* of a given input *point*:

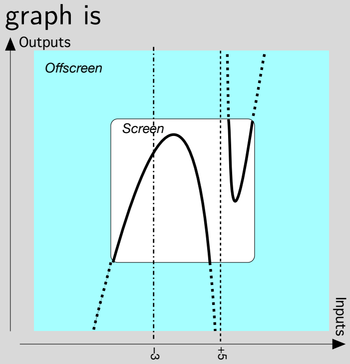
PROCEDURE 1.11 To get the local graph for inputs in a neighborhood of a given point when the function is given by a global graph

i. Mark a neighborhood of the point on the input ruler,
ii. Draw the input level band for the neighborhood of the point using **??** **??** - **??** (**??**),
iii. The local graph near the point is at the intersection of the input level band and the global graph.

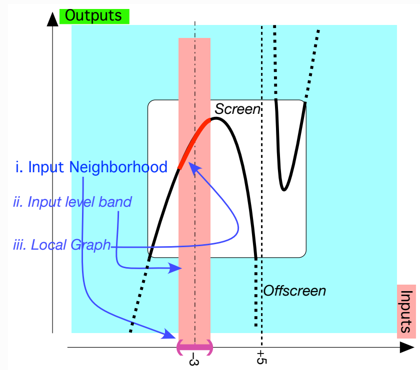
While the procedure is the same regardless of the nature of the point, we will look at the difference cases separately

6. Local graph near x_0.

DEMO 1.11a
To get the local graph near -3 of the function $\mathcal{MARE}$ whose global graph is

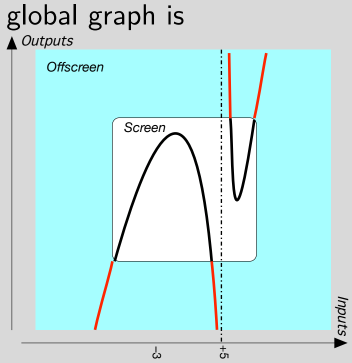


i. We mark a *neighborhood of* -3 on the *input ruler*,
ii. We draw the *input level* band for the *neighborhood* of -3,
iii. The *local graph* of $\mathcal{MARE}$ near -3 is at the intersection of the input level *band* with the global graph,

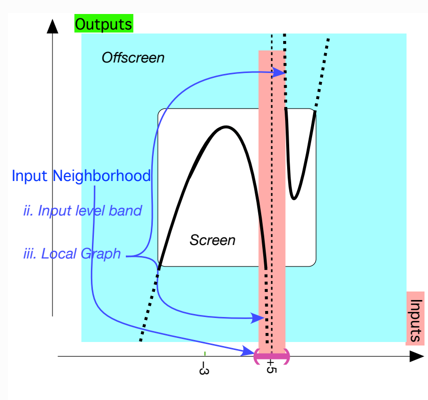


DEMO 1.11b

To get the local graph near the pole $+5$ of the function $\mathcal{JEN}$ whose global graph is



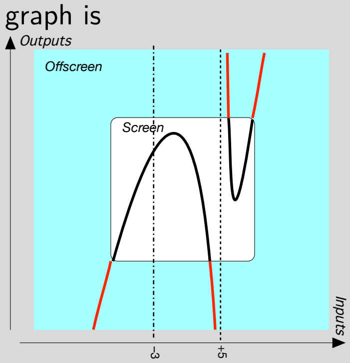
i. We mark a *neighborhood of* $+5$ on the *input ruler*,
ii. We draw the *input level* band through the *neighborhood* of $+5$,
iii. The *local graph* of $\mathcal{JEN}$ near $+5$ is the intersection of the input level *band* with the global graph,



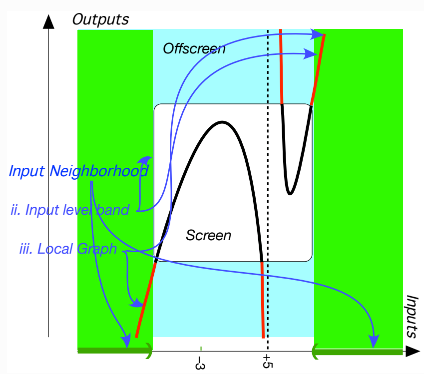
7. Local graph near ∞.

Keep in mind that even for large inputs, a function may return outputs of any qualitative size, medium-size: large-size or small-size.

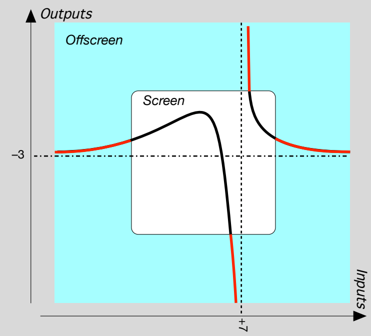
Demo 1.11c
To get the local graph near ∞ of the function $\mathcal{REN}$ whose global graph is



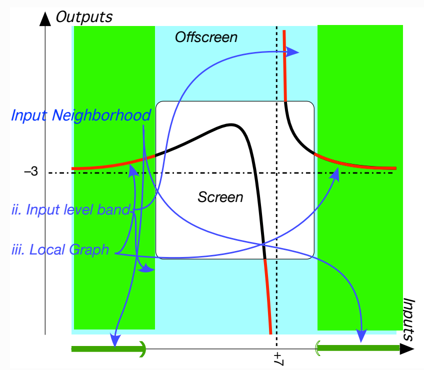
i. We mark a *neighborhood of* ∞ on the *input ruler*,
ii. We draw the *input level* band through the *neighborhood* of ∞,
iii. The *local graph* of $\mathcal{REN}$ near ∞ is the intersection of the input level *band* with the global graph,



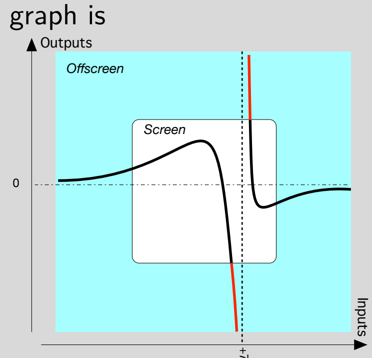
Demo 1.11d
To get the local graph near ∞ of the function $\mathcal{MINA}$ whose global graph is



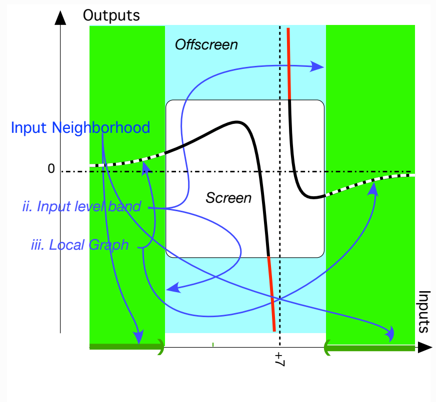
i. We mark a *neighborhood of* ∞ on the *input ruler*,
ii. We draw the *input level* band through the *neighborhood* of ∞,
iii. The *local graph* of $\mathcal{MINA}$ near ∞ is the intersection of the input level *band* with the global graph,



Demo 1.11e
To get the local graph near ∞ of the function $\mathcal{RHEA}$ whose global graph is



i. We mark a *neighborhood of* ∞ on the *input ruler*,
ii. We draw the *input level* band through the *neighborhood* of ∞,
iii. The *local graph* of $\mathcal{RHEA}$ near ∞ is the intersection of the input level *band* with the global graph,



	OKsoFAR		OKsoFAR		OKsoFAR
	OKsoFAR		OKsoFAR		OKsoFAR

Chapter 2

Local Features Functions May Have

Keep in mind that:

► **Cautionary note 2.1** We will define 'local behavior' as a *calculus* word later. In the meantime, we will use 'local behavior' as an *everyday* word.

► **Cautionary note 2.2** The functions we are discussing in this Part I - Functions Given By Data (Page 63) are given by *curves* and, while in *this* text what we will say will also apply to the functions given other than by a curve, this will not necessarily be the case for any and all functions.

1 Local Code

There is no reason to expect the local behavior of a function to be the same on both sides of a input point, be it x_0 or ∞, see **?? ??** - **??** (**??**)) and **??**

?? - **??** (**??**)).

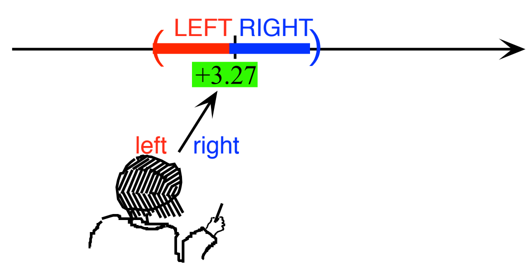
1. Facing the neighborhood. In order to deal *separately* with each side of a neighborhood of a given point, we first need to state precisely which side of the given point is going to be LEFT of the given point and which side of the given point is going to be RIGHT of the given point.

Example 2.1. Given a neighborhood of the *number* $+3.27$, JILL can face the center of the neighborhood and then:

- what is to JILL's left will be what is LEFT of $+3.27$

and

- what is to JILL's right will be what is RIGHT of $+3.27$.

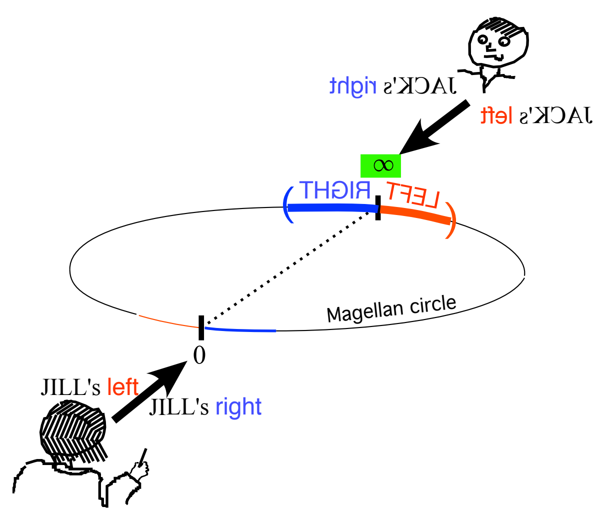


Example 2.2. Given a neighborhood of ∞, JILL cannot face the center of the neighborhood and so, using a Magellan circle, she must imagine JACK facing a neighborhood of ∞ and then:

- what is to JACK's left will be what is LEFT of ∞

and

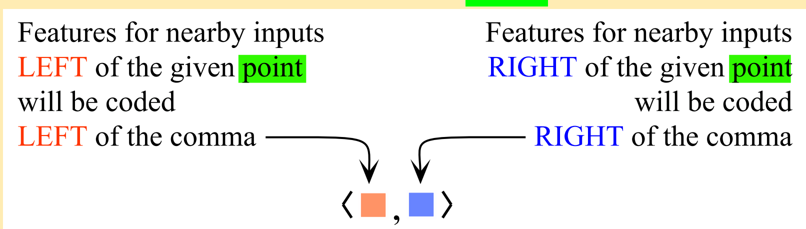
- what is to JACK's right: will be what is RIGHT of ∞



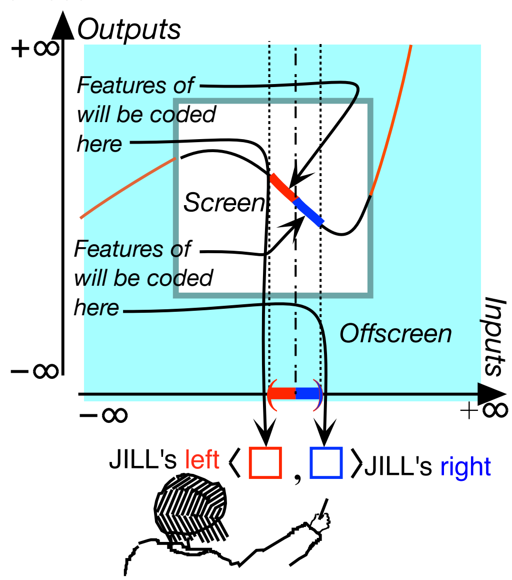
2. Local code. in order to describe *separately* the 'local behavior' on each side of the given input, we will use the following format:

basic format angle

DEFINITION 2.1 To code the features of the local graph near a given point, we will write the codes for the feature on each side between two **angles** with a *comma* to separate the behaviors on the sides of the neighborhood of the given point:

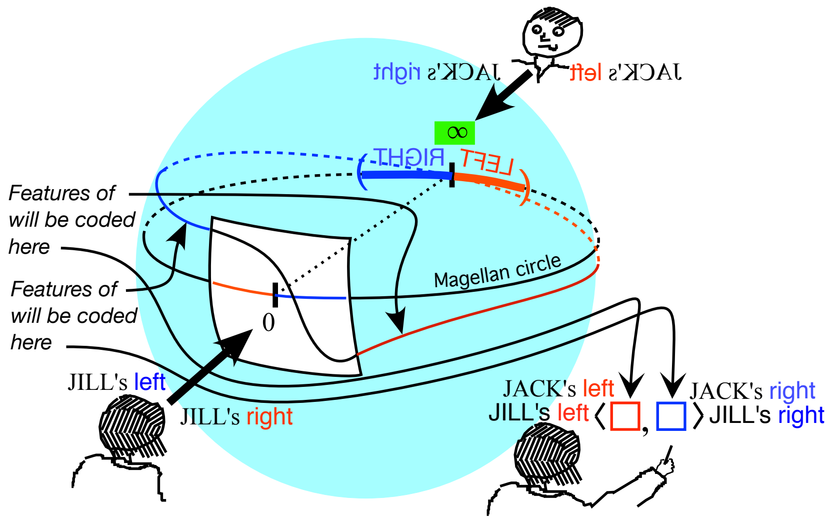


EXAMPLE 2.3. When the local graph is near a number, JILL can face the center of the neighborhood:



EXAMPLE 2.4. When the local graph is near ∞ and since JILL can only imagine JACK facing *infinity* on the far side of a Magellan circel:

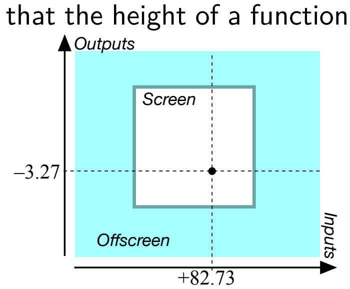
height



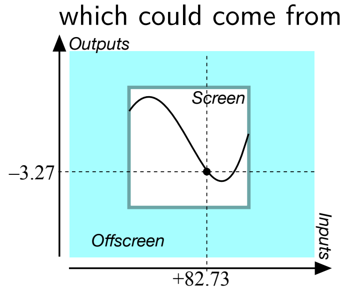
2 Local Height

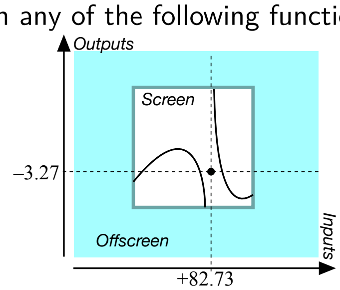
The **height** of a function f *at* a given *number* x_0 is just the *output* $f\left(x_0\right)$ provides almost no information about the graph of the function.

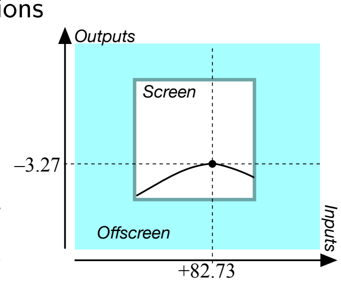
EXAMPLE 2.5. To say that the height of a function at $+82.73$ is -3.27 gives



which could come from any of the following functions







. . . and from many more.

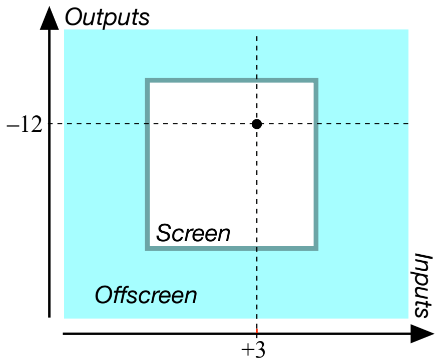
1. Local height near a given point. Given a function f and given a point, the height of f *near* x_0 is

local height-sign

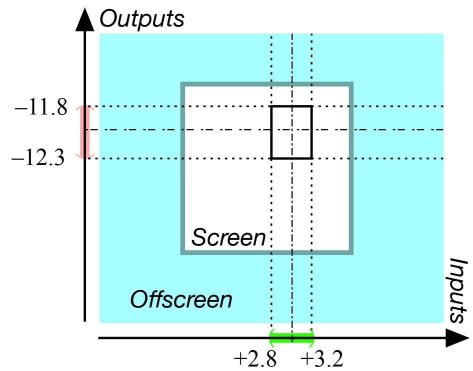
we want a thick version of the height of f *at* x_0 that is the height of f *near* x_0.

EXAMPLE 2.6. Given a function f, to say that

Height f *at* $+3 = -12$
says



Local height$f$ *near* $+3 = -12 \oplus h$
sats



As will become clear why, though, we have to introduce and discuss the sign and the size of the local height separately.

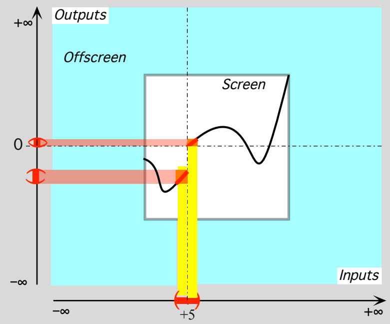
2. Local height-sign. The **local height-sign** of f near x_0 is the sign, $+$ or $-$, of the outputs for nearby inputs on each side of the given input.

PROCEDURE 2.1 To get the local height-sign near x_0 of a function given by a curve,

i. Highlight the *local graph* near x_0 using PROCEDURE 5.2 - Local graphs of a $\mathcal{UNIT}$ function (Page 230)
ii. Get from the local graph the sign, $+$ or $-$, of the *outputs* for nearby inputs on each side of the given input,
iii. Code the local height-sign f using **??** **??** - **??** (**??**)

DEMO 2.1 To get the local height-sign near $+5$ for the function IAN from the local graph near $+5$

local height-size



i. We get from the local graph the sign of the *outputs* for nearby inputs on each side of $+5$:

- The sign of the outputs *left* of $+5$ is $-$
- The sign of the outputs *right* of $+5$ is $+$

ii. We code the local height-sign:

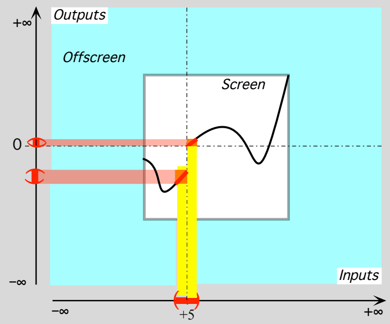
Local height-sign IAN near $+5 = \langle -, + \rangle$

3. Height-size The **local height-size** of f near a given input is the qualitative size, *large*, *medium* or *small*, of the *outputs* for nearby inputs on each side of the given input.

PROCEDURE 2.2 To get the height-size near a given input of a function from its global graph,

i. Highlight the *local graph* near the given input using PROCEDURE 5.2 - Local graphs of a $\mathcal{UNIT}$ function (Page 230)
ii. Mark a neighborhood of the given point
iii. Get from the local graph the qualitative size, *large*, *medium* or *small*, of the *outputs* for nearby inputs on each side of the given input,
iv. Code height-size f according to ?? ?? - ?? (??)

DEMO 2.2a Get height-size near $+5$ for the function IAN from the local graph near $+5$



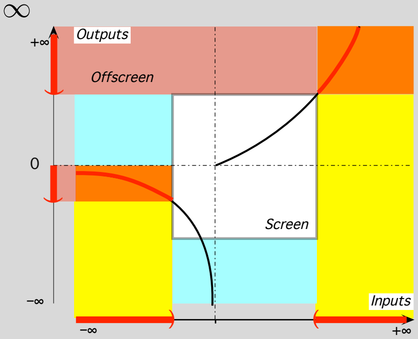
i. We get from the local graph the qualitative size, $large$, $medium$ or $small$, of the ***outputs*** for nearby inputs on each side of $+5$:

- The size of the outputs *left* of $+5$ is $medium$
- The size of the outputs *right* of $+5$ is $medium$

ii. We code the height-size:

height-size IAN near $+5 = \langle medium, medium \rangle$

DEMO 2.2b Get height-size near ∞ for the function IAN from the local graph near ∞



i. We get from the local graph the qualitative size, $large$, $medium$ or $small$, of the ***outputs*** for nearby inputs on each side of ∞:

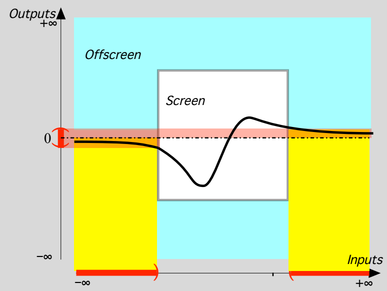
- The size of the height *left* of ∞ is $large$
- The size of the height *right* of ∞ is $small$

ii. We code the height-size:

height-size IAN near $\infty = \langle large, small \rangle$

$x_{\infty\text{-height}}$
$x_{0\text{-height}}$
height

DEMO 2.2c For the function



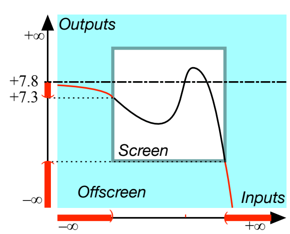
the Magellan input ∞ is a *zero* because:
the outputs for nearby inputs, both inputs *right* of ∞ and inputs *left* of ∞, are all $small$,

==========OK SO FAR ===========

4. Local height near ∞ The concept of height provides us with conveniently systematic names:

- For a pole: $\boldsymbol{x}_{\infty\text{-height}}$
- For a zero: $\boldsymbol{x}_{0\text{-height}}$

The height near ∞



is $-large$ for inputs left of ∞ and $-small$ for inputs right of ∞

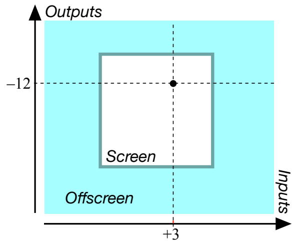
Given a function f, we will thicken the output *at* a given input, be it x_0 or ∞, into the **height** *near* the given input.

EXAMPLE 2.7.

local maximum-height input

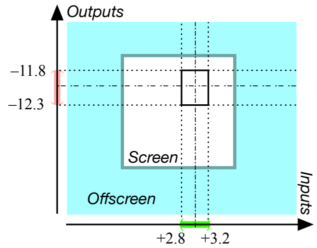
$x_{\text{maxi-height}}$

The output at +3



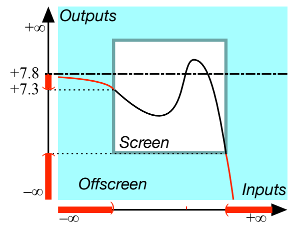
is -12

The height near +3



is $-12 \pm$ *small*

The height near ∞



is $-$*large* for inputs left of ∞ and $-$*small* for inputs right of ∞

3 Local extreme

We will often compare the *output at* a given *medium-size* input with the *height near* the given *medium-size* input.

1. Local maximum-height input. A **local maximum-height input** is a *medium-size* input whose output is *larger* than the height near the medium-size input. In other words, the output *at* a local maximum-height input is *larger* than the outputs for all nearby inputs.

x_0 is al local maximum-height input whenever $f(x_0) > f(x_0 + h)$

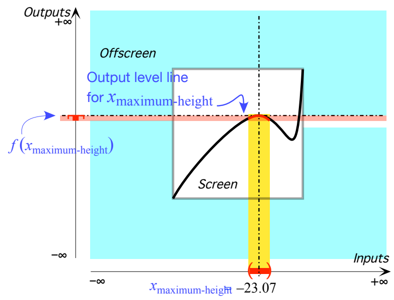
We will use $\boldsymbol{x}_{\textbf{max-height}}$ as a name for a local maximum-height input.

Language note 2.1 $\boldsymbol{x}_{\textbf{max}}$ is the usual name for a local maximum-height input but x_{max} tends to suggest that it is the input x that is *maximum* while it is the *output*, $f(x_{\text{max}})$, which is "maximum".

Graphically, the local graph near $x_{\text{max-height}}$ is *below* the output-level line for $x_{\text{max-height}}$.

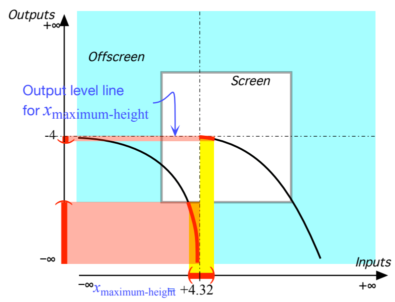
local minimum-height input
$x_{\text{min-height}}$

EXAMPLE 2.8. The function



has a local *maximum* at -23.07 because the output *at* -23.07 is *larger* than the outputs for *nearby inputs*

EXAMPLE 2.9. The function



has a local *maximum* at $+4.32$ because the output *at* $+4.32$ is *larger* than the outputs for *nearby inputs*

2. Local minimum-height input. A **local minimum-height input** is a *medium-size* input whose output is *smaller* than the height near the given input. In other words, the output *at* a local minimum-height input is *smaller* than the outputs for all nearby inputs.

$$\boxed{x_0 \text{ is al local minimum-height input whenever } f(x_0) < f(x_0 + h)}$$

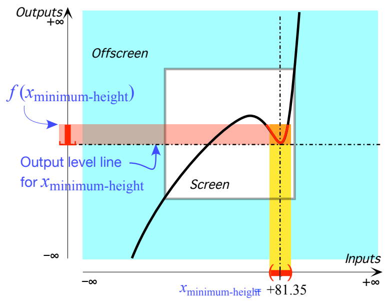
We will use $\boldsymbol{x_{\text{min-height}}}$ as name for a local minimum-height input.

LANGUAGE NOTE 2.2 $\boldsymbol{x_{\text{min}}}$ is the usual name for a local minimum-height input but x_{min} tends to suggest that it is the input x that is *minimum* while it is its *output*, $f(x_{\text{min}})$, which is "minimum".

Graphically, the *local graph* near $x_{\text{min-height}}$ is *above* the output-level line for $x_{\text{min-height}}$.

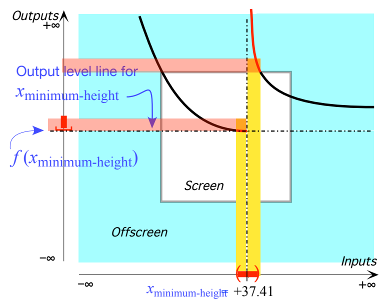
local extreme-height input

EXAMPLE 2.10. The function



has a local *minimum* at $+81.35$ because the output *at* $+81.35$ is *smaller* than the outputs for *nearby inputs*.

EXAMPLE 2.11. The function



has a local *minimum* at $+37.41$ because the output *at* $+37.41$ is *smaller* than the outputs for *nearby inputs*.

3. Local extreme-height input. **Local extreme-height input** are *medium-size* inputs which are either a local maximum-height input or a local minimum-height input.

CAUTIONARY NOTE 2.3 can only be *medium-size* inputs.

4. Optimization problems. Minimization problems and maximization problems (https://en.wikipedia.org/wiki/Mathematical_optimization) as well as min-max problems (https://en.wikipedia.org/wiki/Minimax) are of primary importance in *real life*. So,

- It would be pointless to allow ∞ as a local extreme-height input since it cannot be reached in the *real world*,
- It would be meaningless to allow $+\infty$ as a locally largest output since $+\infty$ is *always* larger than any output or to allow $-\infty$ as a locally smallest output since $-\infty$ is *always* smaller than any output.

local maximum-height input
$x_{\text{maxi-height}}$
local minimum-height input

5. Local extreme We will often compare the *output at* a given *medium* input with the *height near* the given *medium* input.

6. Local maximum-height input. A **local maximum-height input** is a *medium* input whose output is *larger* than the height near the medium input. In other words, the output *at* a local maximum-height input is *larger* than the outputs for all nearby inputs.

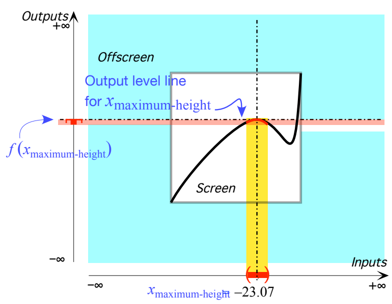
$$\boxed{x_0 \text{ is al local maximum-height input whenever } f(x_0) > f(x_0 + h)}$$

We will use $\boldsymbol{x_{\text{max-height}}}$ as a name for a local maximum-height input.

LANGUAGE NOTE 2.3 $\boldsymbol{x_{\text{max}}}$ is the usual name for a local maximum-height input but x_{max} tends to suggest that it is the input x that is *maximum* while it is the *output*, $f(x_{\text{max}})$, which is "maximum".

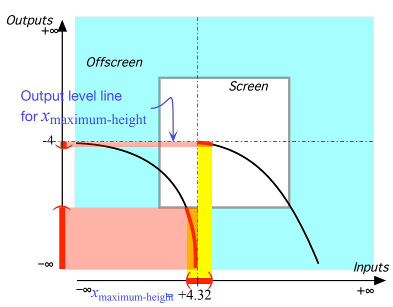
Graphically, the local graph near $x_{\text{max-height}}$ is *below* the output-level line for $x_{\text{max-height}}$.

EXAMPLE 2.12. The function



has a local *maximum* at -23.07 because the output *at* -23.07 is *larger* than the outputs for *nearby inputs*

EXAMPLE 2.13. The function



has a local *maximum* at $+4.32$ because the output *at* $+4.32$ is *larger* than the outputs for *nearby inputs*

7. Local minimum-height input. A **local minimum-height input** is a *medium* input whose output is *smaller* than the height near the given input. In other words, the output *at* a local minimum-height input is *smaller* than the outputs for all nearby inputs.

$x_{\text{min-height}}$ local extreme-height input

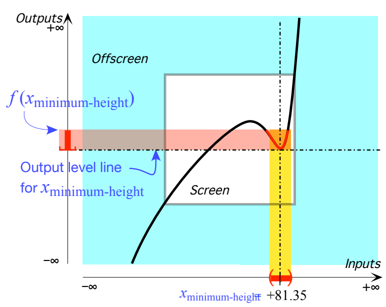
x_0 is al local minimum-height input whenever $f(x_0) < f(x_0 + h)$

We will use $\boldsymbol{x}_{\textbf{min-height}}$ as name for a local minimum-height input.

LANGUAGE NOTE 2.4 $\boldsymbol{x}_{\textbf{min}}$ is the usual name for a local minimum-height input but x_{min} tends to suggest that it is the input x that is *minimum* while it is its *output*, $f(x_{\text{min}})$, which is "minimum".

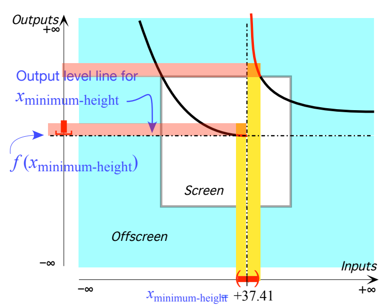
Graphically, the *local graph* near $x_{\text{min-height}}$ is *above* the output-level line for $x_{\text{min-height}}$.

EXAMPLE 2.14. The function



has a local *minimum* at $+81.35$ because the output *at* $+81.35$ is *smaller* than the outputs for *nearby inputs*.

EXAMPLE 2.15. The function



has a local *minimum* at $+37.41$ because the output *at* $+37.41$ is *smaller* than the outputs for *nearby inputs*.

8. Local extreme-height input. Local extreme-height input are *medium* inputs which are either a local maximum-height input or a local minimum-height input.

CAUTIONARY NOTE 2.4 can only be *medium* inputs.

zero
parity
even zero
odd zero

9. Optimization problems. Minimization problems and maximization problems (https://en.wikipedia.org/wiki/Mathematical_optimization) as well as min-max problems (https://en.wikipedia.org/wiki/Minimax) are of primary importance in *real life*. So,

- It would be pointless to allow ∞ as a local extreme-height input since it cannot be reached in the *real world*,
- It would be meaningless to allow $+\infty$ as a locally largest output since $+\infty$ is *always* larger than any output or to allow $-\infty$ as a locally smallest output since $-\infty$ is *always* smaller than any output.

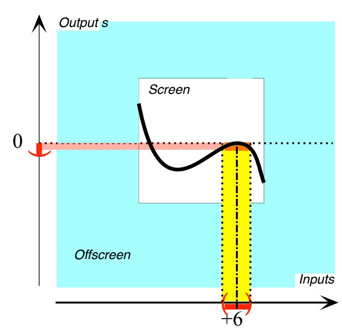
4 Zeros And Poles

Given a function f, a **zero** of f is a *medium* input whose height-size is $\langle small, small\rangle$.

1. Zeros. We will distinguish two kinds of zeros according to their **parity**: https://en.wikipedia.org/wiki/Zeros_and_poles

- ▸ An **even zero** is a zero whose height-sign is either $\langle +,+\rangle$ or $\langle -,-\rangle$.

EXAMPLE 2.16. For the function f given by the global graph

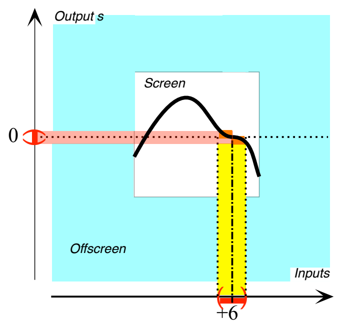


the medium input $+6$ is an *even zero* because:

- ▸ the *outputs* for inputs *near* $+6$ are all $small$,
- ▸ height-sign f near $+6 = \langle -,-\rangle$ (Same signs.)

- ▸ An **odd zero** is a zero whose height-sign is either $\langle +,-\rangle$ or $\langle -,+\rangle$.

EXAMPLE 2.17. For the function f given by the global graph



the medium input $+6$ is an *odd zero* because:

- the *outputs* for inputs *near* $+6$ are all $small$,
- height-sign f near $+6 = \langle +, - \rangle$ (Opposite signs.)

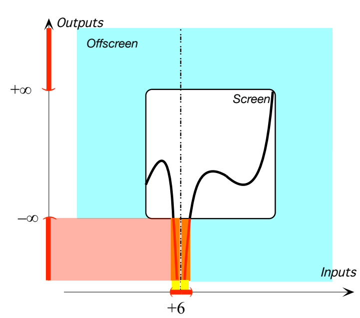
pole
parity
even pole
odd pole

2. Poles. Given a function f, a **pole** of f is a *medium* input whose height-size is $\langle large, large \rangle$. We will distinguish two kinds of poles according to their parity:

We will distinguish two kinds of poles according to their **parity**:

- An **even pole** is a pole whose height-sign is either $\langle +, + \rangle$ or $\langle -, - \rangle$.

EXAMPLE 2.18. For the function f given by the global graph



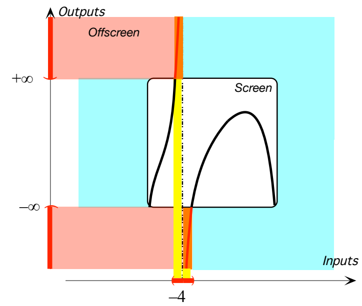
the medium input $+6$ is an *even pole* because:

- the *outputs* for inputs *near* $+6$ are all $large$,
- height-sign f near $+6 = \langle -, - \rangle$ (Same signs.)

- An **odd pole** is a pole whose height-sign is either $\langle +, - \rangle$ or $\langle -, + \rangle$.

slope
slope-sign

EXAMPLE 2.19. For the function f given by the global graph



the medium input $+-4$ is an *odd pole* because:

- the *outputs* for inputs *near* -4 are all $large$,
- height-sign f near $-4 = \langle +, - \rangle$ (Opposite signs.)

5 Local Slope

1. Slope-sign. Inasmuch as, in this text, we will only deal with *qualitative* information we will be mostly interested in the **slope-sign**: .

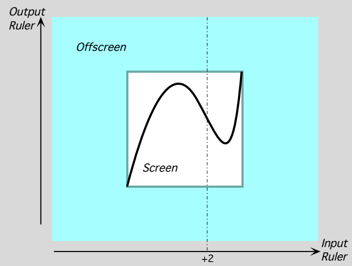
PROCEDURE 2.3 To get Slope-sign near a given input for a function given by a global graph

i. Mark the local graph near the given input
ii. Then the slope-sign is:
╱ when the local graph looks like ╱ or ╭, that is when the *outputs* are **increasing** as the inputs are going the way of the input ruler,
╲ when the local graph looks like ╲ or ╮, that is when the *outputs* are **decreasing** as the inputs are going the way of the input ruler.
iii. Code Slope-sign f according to **??** **??** - **??** (**??**)

LANGUAGE NOTE 2.5 Slope-sign The usual symbols are + Instead of ╱ and − instead of ╲ but, in this text, in order not to overuse + and −, we will use ╱ and ╲.[1]

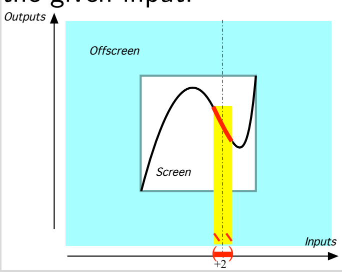
DEMO 2.3a Let HIC be the function whose Mercator graph is

[1] Educologists will surely appreciate "Sign-slope $f =$ ╱ *iff* Sign-heigth $f' = +$".



and let the given input be $+2$. Then to get Slope-sign HIC near $+2$

i. We get the local graph near the given input:



ii. We then get

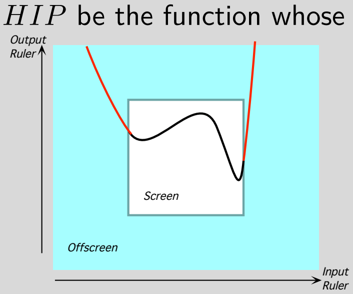
· The slope sign *left* of $+2$ is $\diagdown$

· The slope sign *right* of $+2$ is $\diagdown$

which we code as:

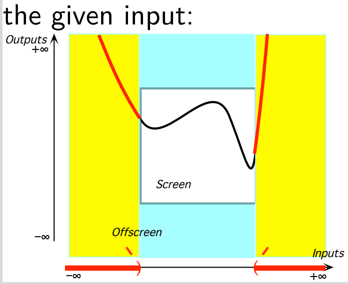
Slope-sign HIC near $+2 = \langle \diagdown, \diagdown \rangle$

Demo 2.3b Let HIP be the function whose Mercator graph is



and let the given input be ∞. Then to get Slope sign HIP near ∞

i. We get the local graph near the given input:



ii. We then get that:

· The slope sign *left* of ∞, that is near $+\infty$, is $\diagup$

· The slope sign *right* of ∞, that is near $-\infty$, is $\diagdown$

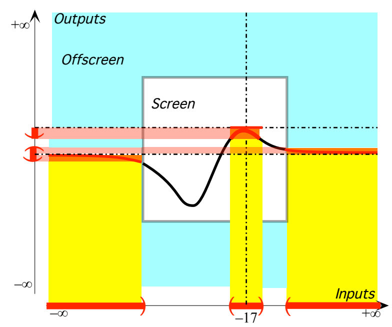
which we code as:

Slope-sign HIP near $\infty = \langle \diagup, \diagdown \rangle$

slope-size
concavity
concavity-size
concavity-sign

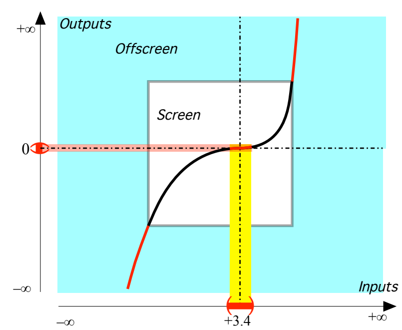
2. Slope-size In this text, we will not deal with **slope-size** other than in the case of a **0-slope input** that is an input, be it x_0 or ∞, near which slope-size is *small*. This is because 0-slope inputs will be extremely important in *global analysis* as finding 0-slope inputs comes up all the time in direct "applications" to the real world:

Example 2.20. The function



has both -17 and ∞ as 0-slope inputs

Example 2.21. The function



Only $+3.4$ is a 0-slope input.

6 Local Concavity

1. Concavity-sign Inasmuch as, in this text, we will be only interested in *qualitative analysis* we will not deal with the **concavity-size** but only with the **concavity-sign**:

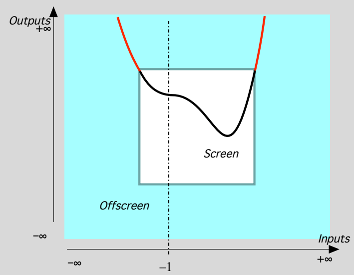
Procedure 2.4 To get Concavity-sign near a given input for a function given by a *global graph*

i. Mark the local graph near the given input
ii. Then the concavity-sign is:
$\cup$ when the local graph is *bending up* like ⟍ or ⟋,
$\cap$ when the local graph is *bending down* like ⟋ or ⟍.
iii. Code Slope-sign f according to ?? ?? - ?? (??)

Language note 2.6 Concavity-sign The usual symbols are $+$ Instead of $\cup$ and $-$ instead of $\cap$ but, in this text, in order not to overuse $+$ and $-$, we will use $\cup$ and $\cap$.[2]

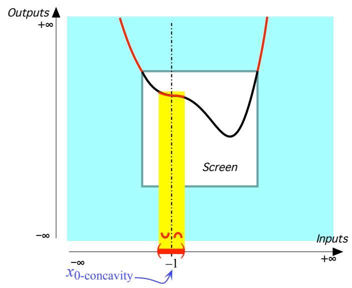
[2]Educologists will surely appreciate "Sign-concavitye $f = \cup$ *iff* Sign-heigth $f'' = +$".

DEMO 2.4 Let KIP be the function whose Mercator graph is



and let the given input be -1. Then to get Concavity sign KIP near -1

i. We get the local graph near the given input:



ii. We then get that:

- The concavity sign *left* of -1, is $\cup$
- The concavity sign *right* of -1, is $\cap$

which we code as:

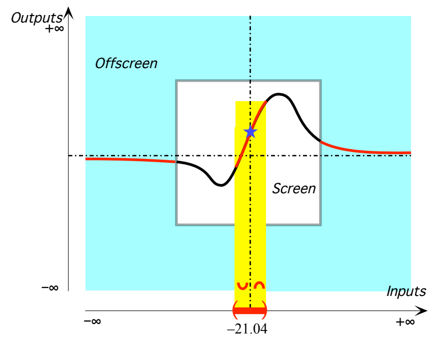
Concavity Sign KIP near $-1 = \langle \cup, \cap \rangle$

2. 0-concavity input. Given a function f, the inputs whose Concavity-size is 0 will be particularly important in *global analysis*:

A *medium* input x_0 is a 0-**concavity input** if inputs that are near x_0 have *small concavity*. We will use $x_{0\text{-concavity}}$ to refer to 0-concavity inputs.

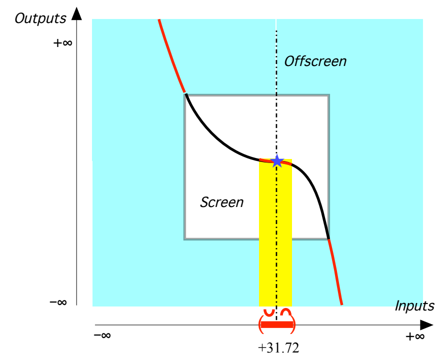
extremity

EXAMPLE 2.22. Given the function whose Mercator graph is



$x_{0-concavity} = -21.04$

EXAMPLE 2.23. Given the function whose Mercator graph is

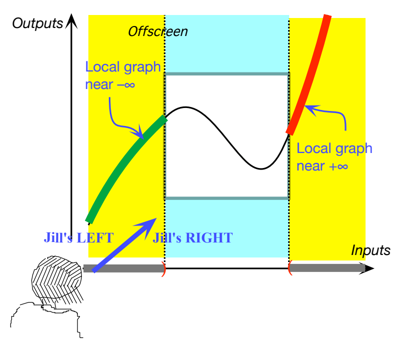


$x_{0-concavity} = +31.72$

Under AGREEMENT 1.2 - Colors for left things and Tight things (Page 64), with only a Mercator view of the global graph, there is of course no way we can get the whole local graph near ∞ and we will have to content ourselves with just the **extremities** of the local graph near ∞. However, since we cannot face ∞ and can only face the screen, we have to keep in mind **??** **??** - **??** (**??**)) so that

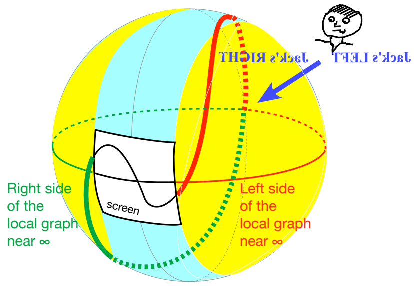
- The extremity of the local graph near $+\infty$ (*left of* ∞) is to *our right*,
- The extremity of the local graph near $-\infty$ (*right of* ∞) is to *our left*.

EXAMPLE 2.24.

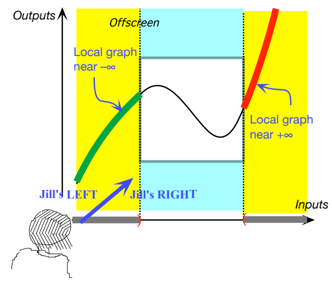


Jill is facing the *screen* so she can only see the *extremities* of the local graph near ∞ and she must keep in mind **??** **??** - **??** (**??**)) so that the local graph near $+\infty$ (to *her right*) is *left* of ∞ and the local graph near $-\infty$ (to *her left*) is *right* of ∞.

EXAMPLE 2.25.



When facing the *screen*, though, Jill can only see the *extremities* of the local graph near ∞ and she must keep in mind that the local graph near $+\infty$ (*left* of ∞) is to Jill's *right* and the local graph near $-\infty$ (*right* of ∞) is to Jill's *left*.



When facing the *screen*, though, Jill can only see the *extremities* of the local graph near ∞. As a result, the local graph near $+\infty$ (*left* of ∞) is to Jill's *right* and the local graph near $-\infty$ (*right* of ∞) is to Jill's *left*.

that is the largest error that will not change the qualitative information we are looking for. The largest permissible error, which is the equivalent of a tolerance, will turn out to be easy to determine.

We can see from Chapter 3 that the reason could not possibly give us a global graph is that, if a plot point may tell us where the global graph "is at", a plot point certainly cannot tell us anything about where the global graph "goes from there". And, since the latter is precisely what local graphs do with slope and concavity, we are now in a position to:

==================================

Something wrong with references here

1. *Describe* how to interpolate local graphs into a global graph. This corresponds to the second of the **?? ??** - **??** (**??**)

2. Discuss questions about interpolating local graphs which correspond to the other two **?? ??** - **??** (**??**)

i. How will we know near which inputs to get the local graphs?

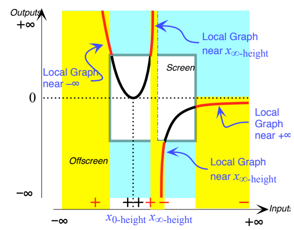
ii. After we have interpolated the local graphs, how will we know if the curve we got *is* the global graph?

7 Feature Sign-Change Inputs

We will often need to find *medium* inputs such that the outputs for nearby inputs left of x_0 and the outputs for nearby inputs right of x_0 have given feature-signs.

1. height sign-change input An input is a **height sign-change input**whenever height sign $= \langle +, - \rangle$ or $\langle -, + \rangle$. We will use $x_{\textbf{height sign-change}}$to refer to a *medium* height sign-change input.

EXAMPLE 2.26.
Let f be the function given by the global graph

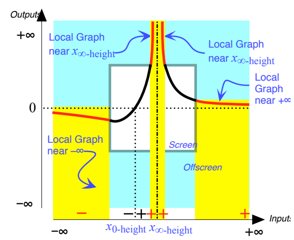


Then,

- $x_{0\text{-height}}$ *is not* a height sign-change input,
- $x_{\infty\text{-height}}$ *is* a height sign-change input.
- ∞ *is* a height sign-change input.

EXAMPLE 2.27.

Let f be the function given by the global graph



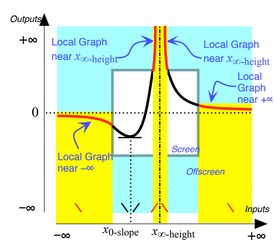
Then,

- $x_{0\text{-height}}$ *is* a height sign-change input,
- $x_{\infty\text{-height}}$ *is not* a height sign-change input,
- ∞ *is* a height sign-change input.

2. Slope sign-change input An input is a **Slope sign-change input**whenever Slope sign $= \langle \diagup, \diagdown \rangle$ or $\langle \diagdown, \diagup \rangle$. We will use $x_{\textbf{Slope sign-change}}$to refer to a Slope sign-change input.

EXAMPLE 2.28.

Let f be the function given by the global graph

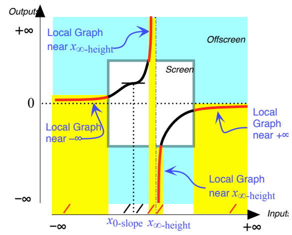


Then,

- $x_{0\text{-slope}}$ *is* a Slope sign-change input,
- $x_{\infty\text{-height}}$ *is* a Slope sign-change input,
- ∞ *is not* a Slope sign-change input.

EXAMPLE 2.29.

Let f be the function given by the global graph



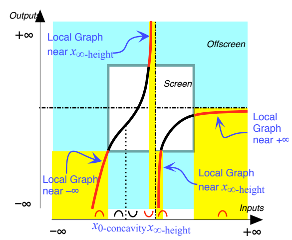
Then,

- $x_{0\text{-slope}}$ *is not* a Slope sign-change input,
- $x_{\infty\text{-slope}}$ *is not* a Slope sign-change input,
- ∞ *is not* a Slope sign-change input.

3. Concavity sign-change input An input is a **Concavity sign-change input**whenever Concavity sign $= \langle \cup, \cap \rangle$ or $\langle \cap, \cup \rangle$. We will use $x_{\textbf{Concavity sign-change}}$to refer to a Concavity sign-change input.

EXAMPLE 2.30.

Let f be the function given by the global graph

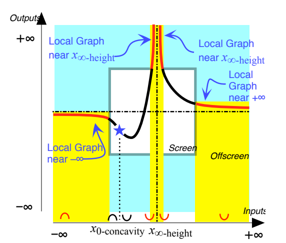


Then,

- $x_{0\text{-concavity}}$ *is* a Concavity sign-change input,
- $x_{\infty\text{-height}}$ *is* a Concavity sign-change input.
- ∞ *is not* a Concavity sign-change input.

EXAMPLE 2.31.

Let f be the function given by the global graph



Then,

- $x_{0\text{-concavity}}$ *is* a Concavity sign-change input,
- $x_{\infty\text{-height}}$ *is not* a Concavity sign-change input,
- ∞ *is* a Concavity sign-change input.

One case where the picture gets a bit complicated is when the *output* point is ∞, that is when the *input* point is a *pole*

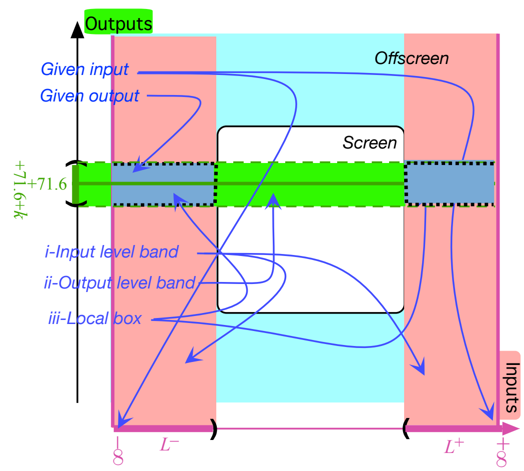
The two other cases where the picture gets a bit complicated are when the *input* point is ∞, whether the *output* point is a number y_0 or ∞.

EXAMPLE 2.32. Local box for the input-output pair $\left(\infty, +71.6\right)$

sided local graph box

i. We get the input level *band* for ∞
ii. We get the output level *band* for $+71,6$
iii. We box the intersection of the input level bands for ∞ and $+71.6$

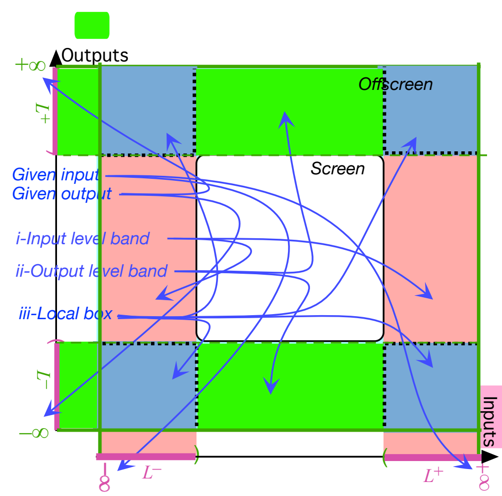
What appears to be two boxes are actually parts of *one* box. This is because we are using the Mercator view. In a Magellan view they would appear as a single box.



EXAMPLE 2.33. Local box for the input-output pair (∞, ∞)

i. We get the input level *band* for ∞
ii. We get the output level *band* for ∞
iii. We box the intersection of the input level bands for ∞ and ∞

What appears to be four boxes are actually parts of *one* box. This is because we are using the Mercator view. In a Magellan view they would appear as a single box.



Actually, we will very often want to keep the two sides of. separate and the **sided local graph box** will then consist of two smaller rectangles, one on each side of the input level line. To get a sided local graph place then,

PROCEDURE 2.5
i. Mark a *neighborhood* of the input on the input ruler,
ii. Draw the *input level band*,
iii. Mark a *neighborhood* of the output on the output ruler,
iv. Draw the *output level band*,

v. Mark which side of the input neighborhood is linked to which side of the output neighborhood,

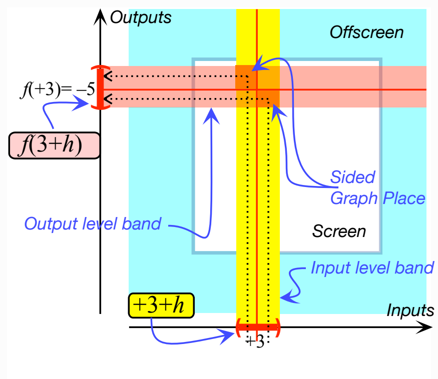
vi. The place for the given input - output pair is at the intersection of the corresponding *sides* of the level bands.

Demo 2.5 Get the sided place for ($+3$, -5) given that:

- $+3^{-} \longrightarrow -5^{+}$
- $+3^{+} \longrightarrow -5^{-}$

i. We mark a *neighborhood of* $+3$ on the *input ruler*,
ii. We draw the *input level band* through the *neighborhood* of $+3$,
iii. We mark a *neighborhood of* -5 on the *output ruler*,
iv. We draw the *output level band* through the *neighborhood* of -5,
v. Mark:

- left of $+3$ $\rightarrow$ above -5
- right of $+3$ $\rightarrow$ below -5



vi. The *sided graph box* for ($+3$, -5) is at the intersection of the corresponding *sides* of the level bands.

We are now going to sketch the way we will graph functions given by I-O rules which we will illusttrate with an extended Example.

Quite a long way away from "just plugging" numbers into the global input-output rule and joining smoothly the plot dots". But that will be graphing that makes sense.

The big missing piece is that we will only be able to get the local framees and will nut be able to really justify the local graphs until Chapter 3.

The general idea will be to

4. Offscreen graph. Local graph(s) near the control input(s)

i. Local graph near ∞. We saw in **??** that $\Big(L\,,\ -2 \oplus [\ldots]\Big)$

ii. Local graph(s) near the pole(s), if any.

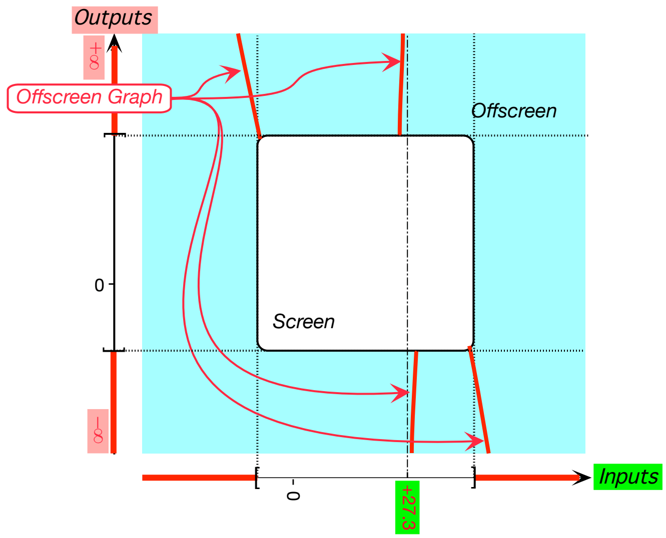
We saw in Example 1.30 that -7 is a pole for the function $JILL$.

We saw in Example 1.32 that $\Big(-7 \oplus h,\ L + [\ldots]\Big)$

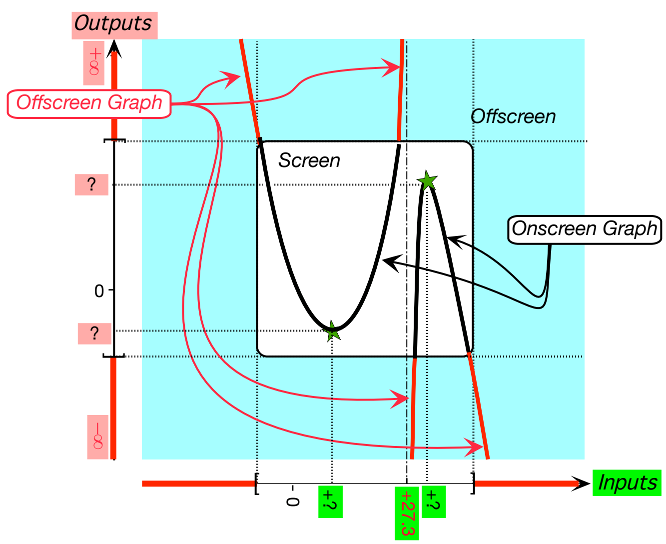
Very roughly speaking! The smooth talk will begin in the next chapter.

iii. Offscreen graph.

EXAMPLE 2.34. Consider the offscreen graph of the function IAN in EXAMPLE 1.29:



Joining smoothly this offscreen graph on-screen gives something like:



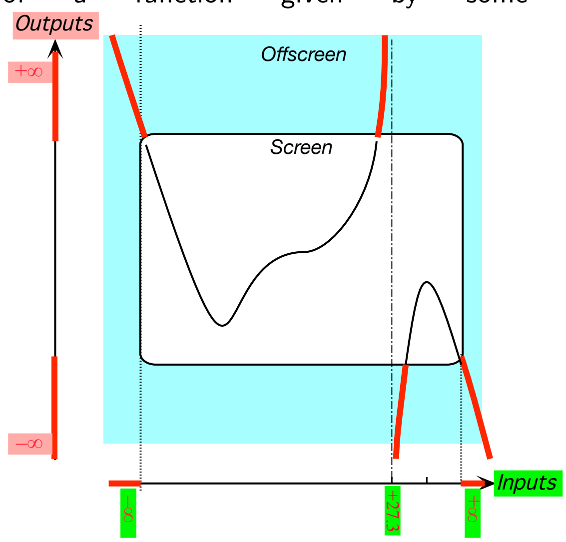
which is pretty much like IAN's actual on-screen graph and even shows IAN's 'essential' features, namely that:

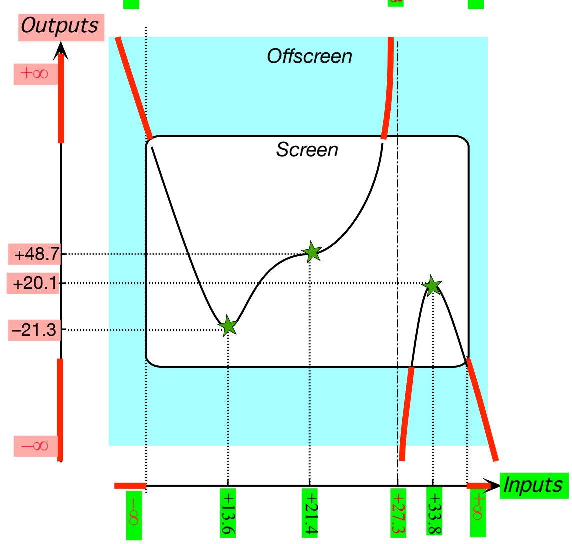
- IAN has *a* 'minimum point', (But of course does *not* show what the input-output pair is.)
- IAN has *a* 'maximum point', (But of course does *not* show what the input-output pair is.)

but does *not* show that IAN has an 'inflection point'.

==============OK SO FAR ===============

EXAMPLE 2.35. Say the following is the global graph of a function given by some I-O rule:





We can *see* from the picture that the given function has:

- What we will call a 'pole': ($+27.3$, ∞).

and

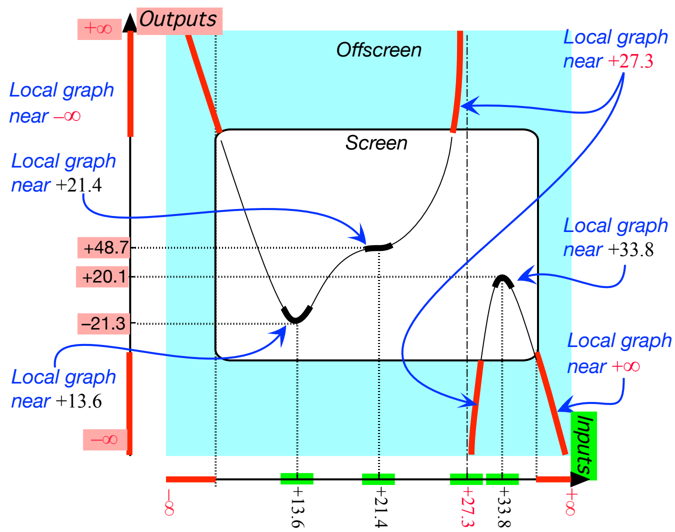
- What we will call a 'minimum point': ($+13.6$, -21.3), *Sneak preview!*

join smoothly

Most *important!*

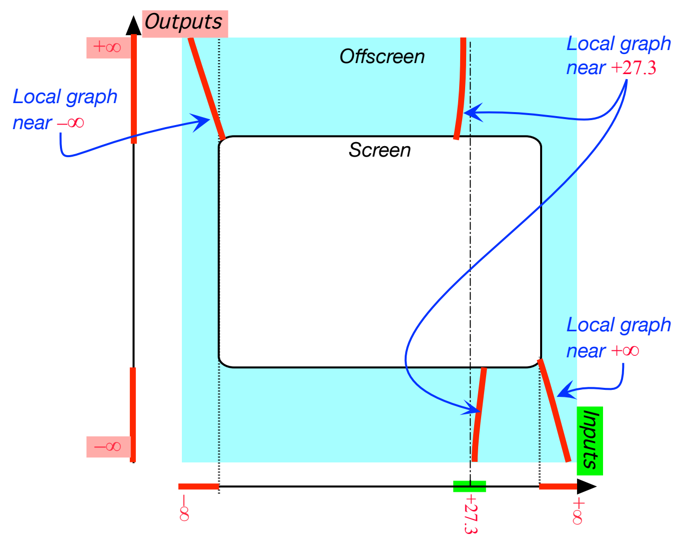
- What we will call an 'inflection point': (+21.4, +48.7),
- What we will call a 'maximum point': (+33.8, +20.1),

EXAMPLE 2.36. In EXAMPLE 1.31, the local graphs are:

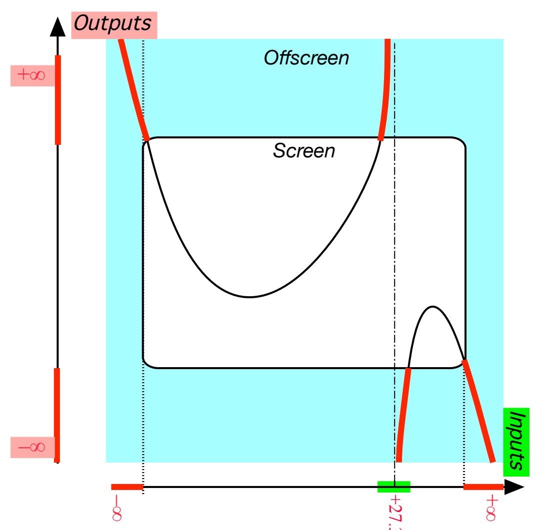


Conversely, our approach to getting the global graph of a function given by an I-O rule will be to use the I-O rule to get the poles of the given function, if any, and then **join smoothly** the local graphs near the pole(s), if any, and near ∞.

EXAMPLE 2.37. To get the global graph in EXAMPLE 1.32 we first get the control local graphs:



which we then join smoothly:



Notice, though, that we while we *did* recover the 'existence' of a 'maximum point' right of $+27.3$ and the 'existence'of a 'minimum point' left of $+27.3$, we did *not* recover the 'existence' of an 'inflection point'.

5. Sided local frame.

We obtain the procedure to get a sided local graph frame just by thickening **??** (**??**):

PROCEDURE 2.6

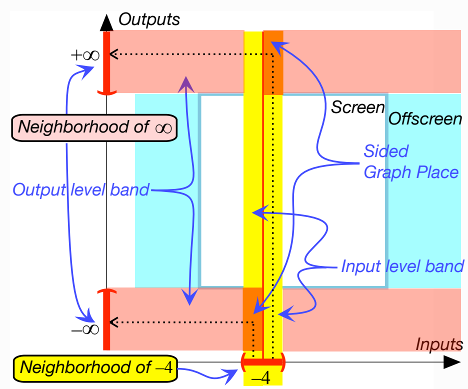
i. Mark a *neighborhood* of the input on the input ruler,
ii. Draw the *input level band*,
iii. Mark a *neighborhood* of the output on the output ruler,
iv. Draw the *output level band*,
v. Mark which side of the input neighborhood is linked to which side of the output neighborhood ,
vi. The local graph box for the given input - output pair is at the intersection of the corresponding *sides* of the level bands.

DEMO 2.6 Get the sided local graph frame for (-4 , ∞) given that:

- -4^- $\longrightarrow$ $-\infty$
- -4^+ $\longrightarrow$ $+\infty$

i. We mark a *neighborhood of* -4 on the *input ruler*,
ii. We draw the *input level band* through the *neighborhood* of -4 ,
iii. We mark a *neighborhood of* ∞ on the *output ruler*,
iv. We draw the *output level band* through the *neighborhood* of ∞ ,
v. Mark:

- left of -4 $\to$ near $-\infty$
- right of -4 $\to$ near



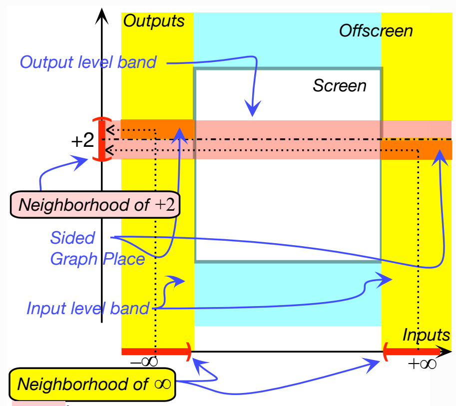
vi. $+\infty$The *sided graph box* for (-4 , ∞) is at the intersection of the corresponding *sides* of the level bands.

DEMO 2.7 Get the sided local graph frame for (∞ , $+2$) given that:

- $-\infty$ $\longrightarrow$ $+2^+$
- $+\infty$ $\longrightarrow$ $+2^-$

i. We mark a *neighborhood of* ∞ on the *input ruler*,
ii. We draw the *input level band* through the *neighborhood* of ∞,
iii. We mark a *neighborhood of* $+2$ on the *output ruler*,
iv. We draw the *output level band* through the *neighborhood* of $+2$,
v. Mark:

- $-\infty \rightarrow +2^{+}$
- $+\infty \rightarrow +2^{-}$



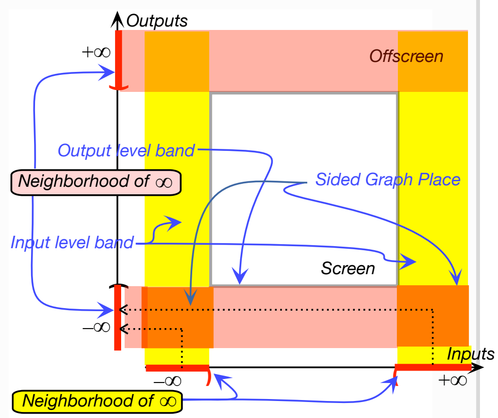
vi. The *sided graph box* for (∞, $+2$) is at the intersection of the corresponding *sides* of the level bands.

DEMO 2.8 Get the sided local graph frame for (∞, ∞) given that:

- $-\infty \longrightarrow -\infty$
- $+\infty \longrightarrow -\infty$

i. We mark a *neighborhood of* ∞ on the *input ruler*,
ii. We draw the *input level band* through the *neighborhood* of ∞,
iii. We mark a *neighborhood of* ∞ on the *output ruler*,
iv. We draw the *output level band* through the *neighborhood* of ∞,
v. Mark:

- $-\infty \rightarrow -\infty$
- $+\infty \rightarrow -\infty$

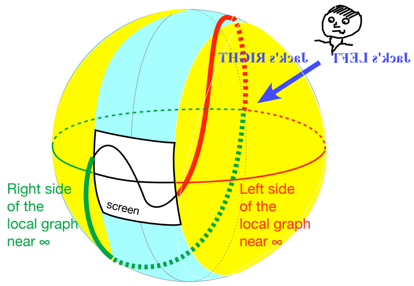


vi. The *sided graph box* for (∞, ∞) is at the intersection of the corresponding *sides* of the level bands.

With a Magellan view of the global graph, we proceed pretty much as in **??** and once we imagine facing ∞, we can *see* which side is which.

==========================

EXAMPLE 2.38.



Jack is facing ∞ so the local graph near $+\infty$ which is to *his left* is *left* of ∞ and the local graph near $-\infty$ which is *to* his right *is right* of ∞.

Height height continuous at x_0

Chapter 3

Global Ways Functions May Behave

Investigating a function consists essentially in finding how the function 'behaves' and, in the case of a function given by a curve, as we will discuss in Chapter 2 - Local Features Functions May Have (Page 133) and Chapter 3 - Global Ways Functions May Behave (Page 165), we can *see* on the curve how the function behaves.

1 Height-Continuity

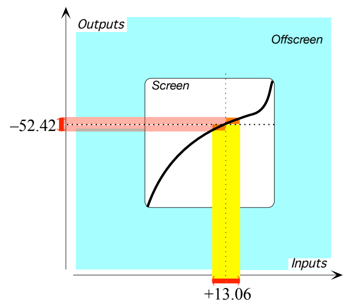
The first kind of abrupt change that can occur is in the size of the outputs for nearby inputs.

1. Height-continuity at x_0. Given a medium-size input x_0, we tend to expect that functions will be **Height height continuous at x_0**, that is that the outputs for nearby inputs will themselves be near $f(x_0)$, the output at x_0.

EXAMPLE 3.1.

height discontinuous
height discontinuous at x_0
jump
hollow dot

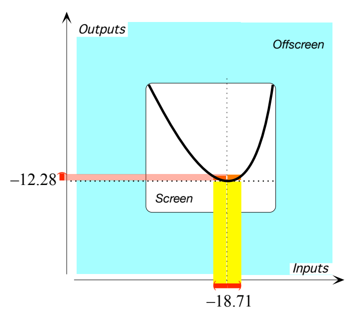
The function



is *height continuous* at $+13.06$ because:

- the output *at* $+13.06$ is -52.42 and
- the outputs for *all* nearby Inputs, both *left* of $+13.06$ and *right* of $+13.06$, are themselves near -52.42.

EXAMPLE 3.2.
The function



is *height continuous* at -18.71 because

- the output *at* -18.71 is -12.28 and
- the outputs for *all* nearby Inputs, both *left* of -18.71 and *right* of -18.71, are themselves *near* -12.28.

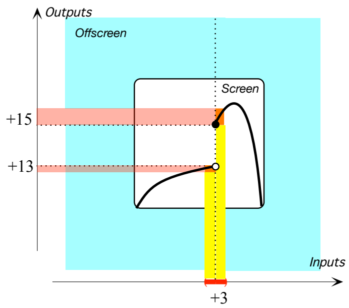
2. Height-discontinuity at x_0. Given a medium-size input x_0, a function is **height discontinuous at x_0** when *not all* the outputs for nearby inputs are near $f(x_0)$, the output at x_0.

- A function can be height discontinuous at x_0 because the function has a **jump** at x_0, that is because the outputs for nearby inputs on one side of x_0 are all near one medium-size output while all the outputs for nearby inputs on the other side of x_0 are near a different medium-size output.

  Since we use solid dots to represent input-output pairs, we will use **hollow dots** for points that *do not* represent input-output pairs.

  EXAMPLE 3.3.

The function



is *height discontinuous* at $+3$ because the function has a *jump* at $+3$ that is:

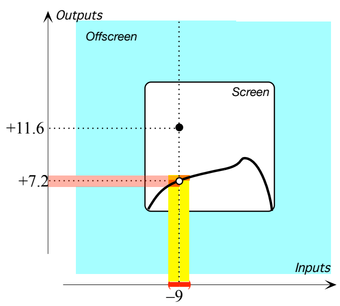
- the outputs for nearby inputs *right* of $+3$ are all near $+15$,

but

- the outputs for nearby Inputs *left* of $+3$ are all near $+13$.

gap

EXAMPLE 3.4.
The function

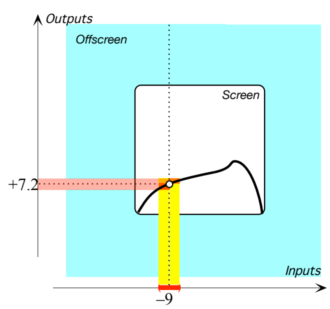


is *height discontinuous* at -9 because the function has a *double jump* at -9 that is:

- even though the outputs for nearby inputs, both inputs *right* of -9 and inputs *left* of -9, are all near $+7.2$,
- the output for -9 itself is $+11.6$.

- A function can be height discontinuous at x_0 because the function has a **gap** at x_0, that is because the function does not return a medium-size output for x_0

EXAMPLE 3.5.
The function

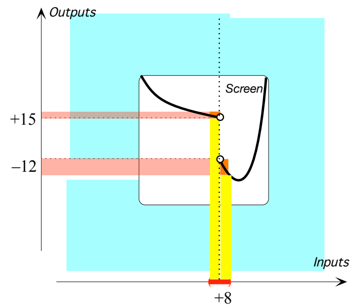


is *height discontinuous* at -9 because the function has a *gap* at -9 that is:

- even though the outputs for nearby inputs, both inputs *right* of -9 and inputs *left* of -9, are all near $+7.2$,
- there is no output for -9 itself.

EXAMPLE 3.6.

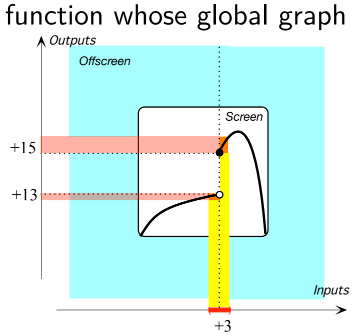
The function



is *height discontinuous* at $+8$ not only because the function has a *jump* at $+8$ but also because the function has a *gap* at $+8$.

EXAMPLE 3.7.

The function whose global graph is



is *height discontinuous* at $+3$ because the global graph has a **jump** at $+3$:

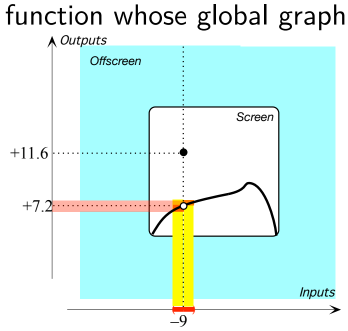
- the outputs for nearby inputs *right* of $+3$ are all near $+15$,

but

- the outputs for nearby Inputs *left* of $+3$ are all near $+13$.

EXAMPLE 3.8.

The function whose global graph is

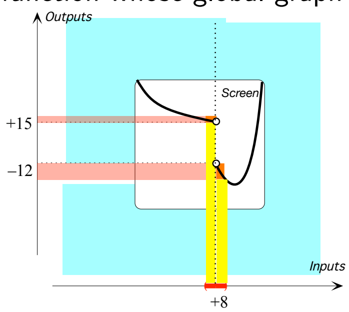


is *height discontinuous* at -9 because the global graph has a **gap** at -9:

- even though the outputs for nearby inputs, both inputs *right* of -9 and inputs *left* of -9, are all near $+7.2$,
- the output for -9 itself is $+11.6$.

EXAMPLE 3.9.

The function whose global graph is

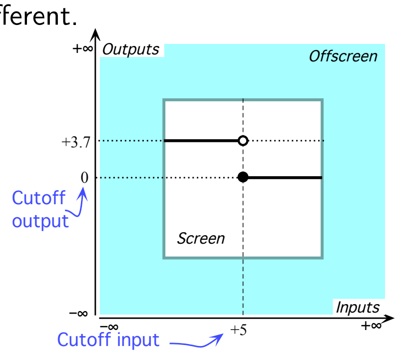


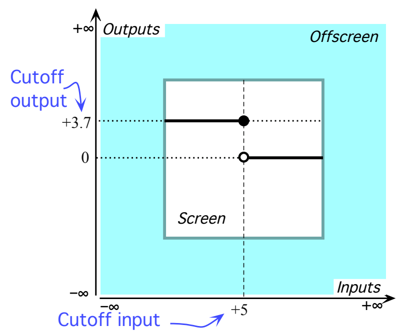
is *height discontinuous* at $+8$ not only because the global graph has a *jump* at $+8$ but also because the global graph has a *gap* at $+8$.

cut-off input
on-off function
transition function
transition

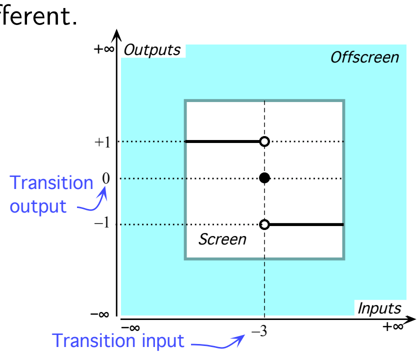
- Actually, height discontinuous functions are quite common in Engineering.

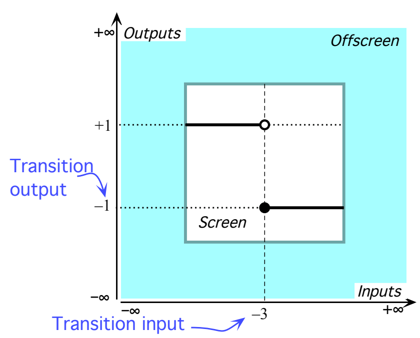
EXAMPLE 3.10. The following **on-off functions** are both *height discontinuous* but are different since the *outputs* for the **cut-off inputs** are different.





EXAMPLE 3.11. The following **transition functions** are both *height discontinuous* but are different since the *outputs* at the **transitions** are different.





- And, finally, there are even functions that are height discontinuous *everywhere*! (https://en.wikipedia.org/wiki/Nowhere_continuous_function)

Magellan height continuous at limit

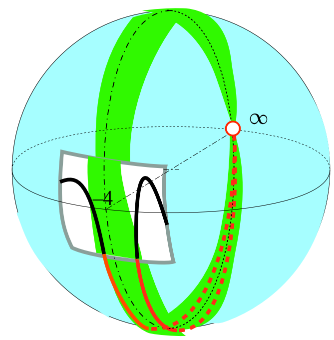
=============OK SO FAR ===============

=======Begin WORK ZONE=======

3. Magellan height-continuity at x_0. A function is **Magellan height continuous at** x_0 when we could remove the height discontinuity at x_0 by overriding or supplementing the global input-output rule with an input-output table involving ∞ as Magellan output.

EXAMPLE 3.12. The function in **??** is *height discontinuous* at -4 because the function has a gap at -4 but *Magellan height continuous* as we could *remove* the gap by *supplementing* the global input-output rule with the input-output table

Input	Output
-4	∞



4. Height-continuity at ∞ The use of nearby inputs instead of the raises a crucial question: Are the outputs for *nearby* inputs *all* near the output *at* the given input?

Any answer, though, will obviously depend on whether or not ∞ is allowed as Magellan input and Magellan output and the reader must be warned that the prevalent stand *in this country* is that ∞ does not exist and that one should use **limits**. (For what limits are, see `https://en.wikipedia.org/wiki/Limit_(mathematics)`.) This for no apparent reason and certainly for none ever given.[1]

As for us, we *will* allow ∞ as Magellan input and Magellan output, an old, tried and true approach. See `https://math.stackexchange.com/questions/354319/can_a_function_be_considered_heightcontinuous_if_it_reaches_infinity_at_one_point` and, more comprehensively, `https://en.wikipedia.org/wiki/Extended_real_number_line`.

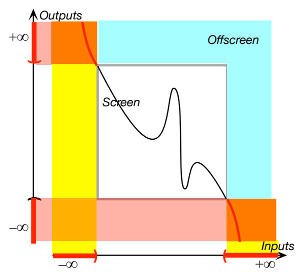
As a backdrop to what we will be doing with Algebraic Functions, we will now show some of the many different possible answers to the above question. For clarity, we will deal with medium-size inputs and medium-size outputs separately from ∞ as Magellan input and Magellan output.

[1]The absolute silence maintained by Educologists in this regard is rather troubling.

Keep in mind that we use solid dots to represent input-output pairs as opposed to hollow dots which do *not* represent input-output pairs.

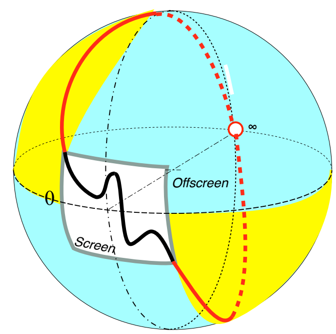
5. Magellan height-continuity at ∞. A function is **Magellan height continuous at ∞** when we could remove the height discontinuity at ∞ by overriding or supplementing the global input-output rule with an input-output table involving ∞ as Magellan input and/or as Magellan output.

EXAMPLE 3.13. The function

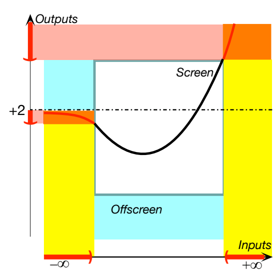


is *height discontinuous* at ∞ but is *Magellan height continuous* since we could remove the height discontinuity with an input-output table involving ∞ as *Magellan input* and *Magellan output*,

Input	Output
$+\infty$	$-\infty$
$-\infty$	$+\infty$

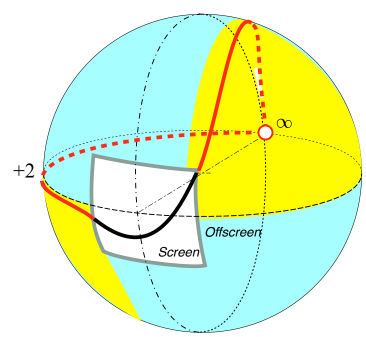


EXAMPLE 3.14. The function



is *height discontinuous* at ∞ but is *Magellan height continuous* since we could remove the height discontinuity with an input-output table involving ∞ as Magellan input and Magellan output

Input	Output
$+\infty$	$+\infty$
$-\infty$	-2^-



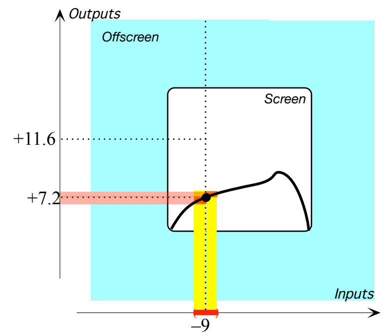
quasi-height continuous at
removable height discontinuity at
remove
override
supplement

6. Quasi height-continuity at x_0. A function is **quasi-height continuous at** x_0 if the height discontinuity could be **removed** by **overriding** or **supplementing** the global input-output rule with an input-output table.

LANGUAGE NOTE 3.1 Removable height discontinuity at x_0 is the standard term but, for the sake of language consistency, rather than saying that a function *has* (or *does not have*) a removable height discontinuity at x_0, *we* will prefer to say that a function *is* (or *is not*) quasi-height continuous at x_0.

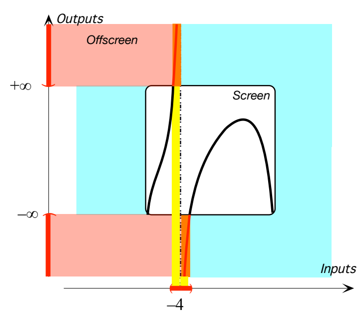
EXAMPLE 3.15. The function in EXAMPLE 3.5 is *height discontinuous* at -9 but the height discontinuity could be *removed* by overriding the input-output pair $(-9, +11.6)$ with the input-output table

Input	Output
-9	$+7.2$



A function can be height discontinuous at x_0 because the function has a pole at x_0.

EXAMPLE 3.16. The function



is *height discontinuous* at -4 because not only does the function have a gap at -4 but the function has a *pole* at -4 that is:

- the outputs for nearby inputs, both inputs *right* of -4 and inputs *left* of -4, are all $large$,

but

- -4 has no medium-size output.

========End WORK ZONE=======

2 Slope-Continuity

1. Tangent. The *first* degree of smoothness is for the *slope* not to have any abrupt change. kink

to be height continuous, that is, to borrow a word from plumbing, we don't want the curve to have any **kink**. More precisely, we don't want any input x_0 for which there is a "jump in slope" from one side of x_0 to the other side of x_0. In other words, we don't want any input x_0 for which the slope on one side differs from the slope on the other side by some medium-size number.

3 Concavity-Continuity

1. Osculating circle. The *second* degree of smoothness is for the *concavity* not to have any abrupt change.

to be height continuous but this is much harder to represent because it is hard to judge by just looking how much a curve is bending.

=======Begin WORK ZONE=======

=======End WORK ZONE========

2. Dealing with poles. The difficulty here stems only from whether or not it is "permisible" to use ∞ as a given input and/or as an output.

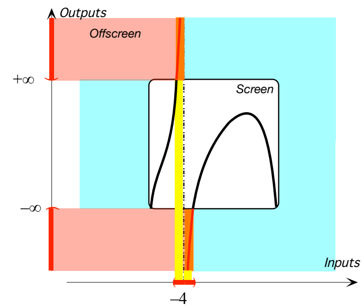
Even though, because **?? ??** - **??** (**??**) (**?? ??** - **??** (**??**)), **?? ??** - **??** (**??**,) *we do* use ∞ as a (Magellan) input and as a (Magellan) output because, as explained in **??** (**??**), we will *only* declare nearby inputs. (Which will shed much light on the local behavior of functions, in particular on the question of height continuity.)

However, the reader ought to be aware that many mathematicians *in this country*, for reasons never stated, flatly refuse to use nearby inputs with their students.

Another reason *we do* is because Magellan views are a very nice basis on which to discuss the local behavior of functions for inputs near ∞ and when outputs are near ∞. In particular, we can see that disheight continuiities caused by poles can be removed using ∞ as a Magellan output.

When ∞ as is not permissible as Magellan input and/or Magellan output, many functions are height discontinuous

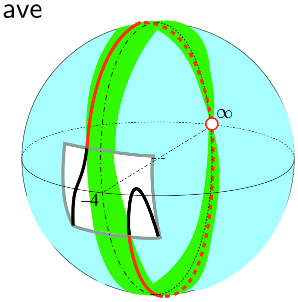
EXAMPLE 3.17. The height discontinuity at -4 of the function in **??** whose Mercator graph is



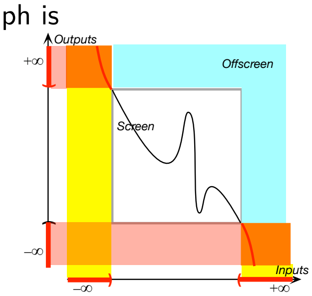
can be *removed* by *supplementing* the global input-output rule with the input-output table:

Input	Output
-4	∞

If we imagine the Mercator graph compactified into a Magellan graph, we have



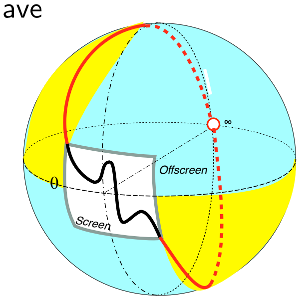
EXAMPLE 3.18. The height discontinuity at ∞ of the function BIB in ?? whose Mercator graph is



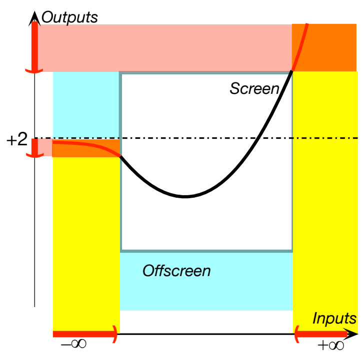
can be *removed* by *supplementing* the global input-output rule with the input-output table:

Input	Output
∞	∞

If we imagine the Mercator graph compactified into a Magellan graph, we have

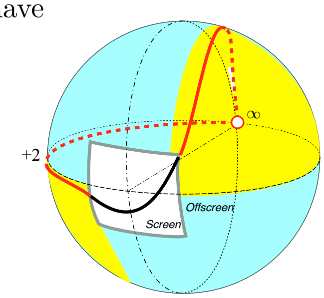


EXAMPLE 3.19. The function whose the global graph in *Mercator view* is



is *height discontinuous* at ∞ not only because the global graph has a *gap* at ∞ since **??** **??** - **??** (**??**) but also because the global graph has a *jump* at ∞.

If we imagine the Mercator view *compactified* into a Magellan view, we have



3. At ∞ The matter here revolves around whether or not ∞ should be allowed as a given input. *We* did but,

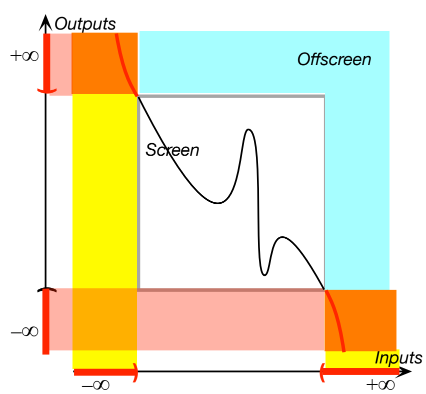
Also, in this section, for a reason which we will explain after we are done, we will have to deal separately with the case when the given input is x_0 and the case when the given input is ∞.

In accordance with **??**, we should say that all functions are height discontinuous at ∞ since the outputs for inputs near ∞ cannot be near the output for ∞ for the very good reason that we cannot use ∞ as input to begin with.

LANGUAGE NOTE 3.2 Continuity at ∞ At ∞, things are a bit sticky:

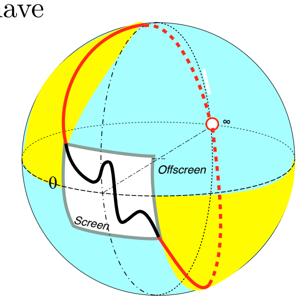
- With a Magellan view, we can *see* if a function is height continuous at ∞ or not.
- Technically, though, to talk of height continuity at ∞ requires being able to take computational precautions not worth taking here but many teachers feel uneasy dealing with height continuity at ∞ without taking these precautions. So, we will not discuss height continuity at ∞ in this text.

EXAMPLE 3.20. The function whose global graph in Mercator view is

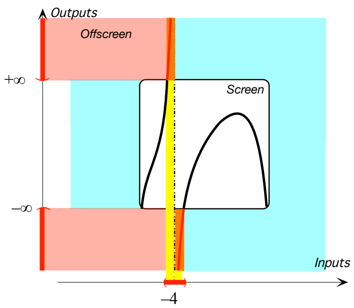


is *height discontinuous* at ∞ because, even though the outputs of inputs near ∞ are all $large$,the global graph has a gap at ∞ since **??**.

If we imagine the Mercator view *compactified* into a Magellan view, we have



EXAMPLE 3.21. The function



is *height discontinuous* at -4 because the global graph has a **pole** at -4:

- the outputs for nearby inputs, both inputs *right* of -4 and inputs *left* of -4, are all $large$,

but, since ??,

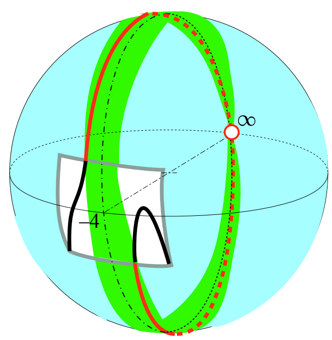
- -4 itself has no output.

Magellan height continuous at

4. Magellan height-continuity at a pole x_0. *We* will say that a function is **Magellan height continuous at** x_0 when we can remove the height discontinuity at x_0 supplementing the offscreen graph with an input-output table involving ∞ as Magellan output.

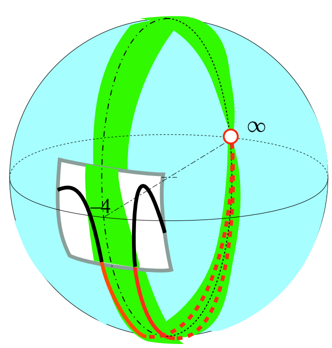
EXAMPLE 3.22. The function in ?? is *height discontinuous* at -4 because the function has a gap at -4 but *Magellan height continuous* as we could *remove* the gap by *supplementing* the global input-output rule with the input-output table

Input	Output
-4	∞



EXAMPLE 3.23. The function in ?? is *height discontinuous* at -4 because the function has a gap at -4 but *Magellan height continuous* as we could *remove* the gap by *supplementing* the global input-output rule with the input-output table

Input	Output
-4	∞



interpolation
smooth interpolation
transition

4 Feature Sign-Change

1.

2.

3.

5 Smooth Interpolations

We now introduce a major tool for extending local graphs and which, starting in **??** **??** - **??** (**??**), we will use to get an approximate global graph for *smooth* functions *not* given by a curve.

1. interpolation

Given two local graphs, **smoothly interpolating** these local graphs consists in drawing inbetween the two given local graphs a curve that:

- is itself smooth

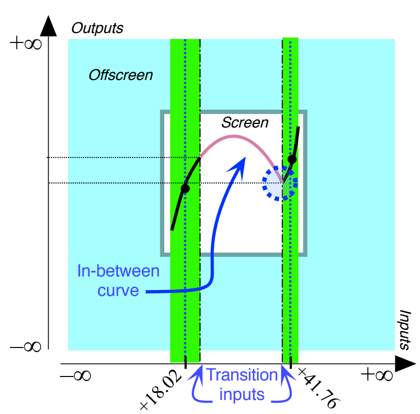
and

- has no jump in height, slope, or concavity at the two **transition** inputs.

Thus, the single curve that results of a smooth interpolation, the single curve that consists of the given local graphs together with the inbetween curve will itself be smooth.

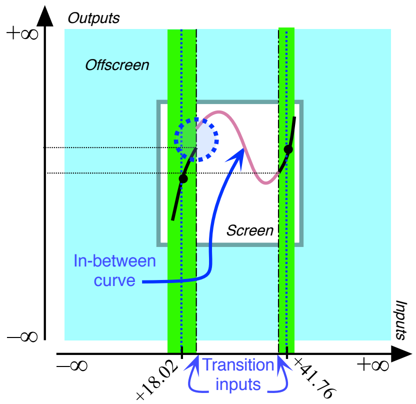
In this chapter, though, since we will only be dealing with given curves we will only be able to "eyeball" compatibility.

EXAMPLE 3.24. The curve inbetween the local graphs



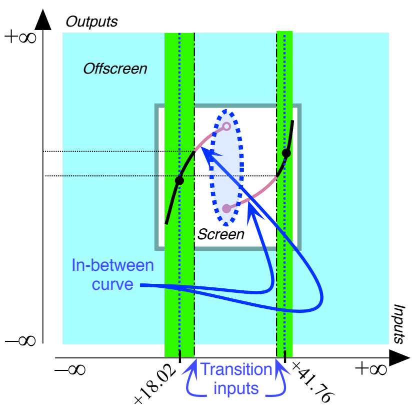
i. is height continuous,
ii. is slope continuous,
iii. is concavity continuous,
but
iv. there is a slope-jump at the transition $+41.76$,
So, this inbetween curve is *not* a smooth interpolation.

EXAMPLE 3.25. The curve inbetween the local graphs



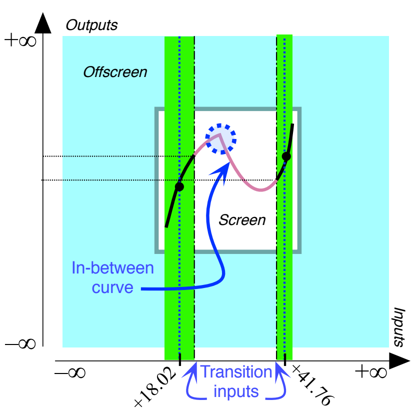
i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
but
iv. there is a height-jump at the transition $+18.02$,
So, this inbetween curve is *not* a smooth interpolation.

EXAMPLE 3.26. The curve inbetween the local graphs



i. is *not* height-continuous,
So, this inbetween curve is *not* a smooth interpolation.

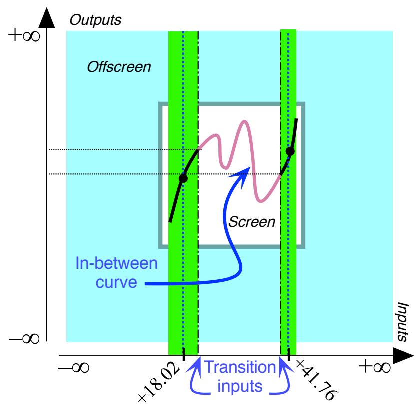
EXAMPLE 3.27. The curve inbetween the local graphs



i. is height-continuous,
ii. is *not* slope-continuous,
So, this inbetween curve is *not* a smooth interpolation.

extraneous
essential
forced

Example 3.28. The curve inbetween the local graphs

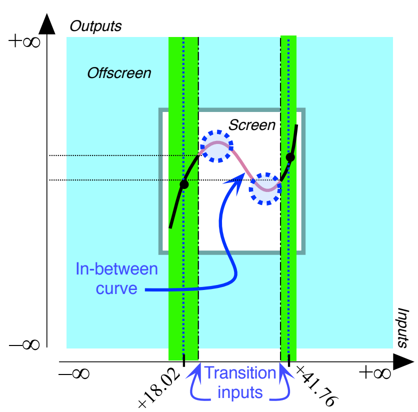


i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
and
iv. there are no jumps at the transitions,
So, this inbetween curve *is* a smooth interpolation

2. However, we will not want the smooth interpolations to introduce **extraneous** features, that is features unwarranted by whichever way the function will be given.

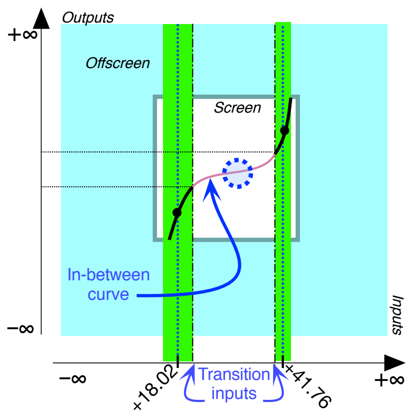
So we will use ***essential*** smooth interpolations in that the feature change inputs for the inbetween curve will all be **forced** by the local graphs being interpolated

Example 3.29. The curve inbetween the local graphs



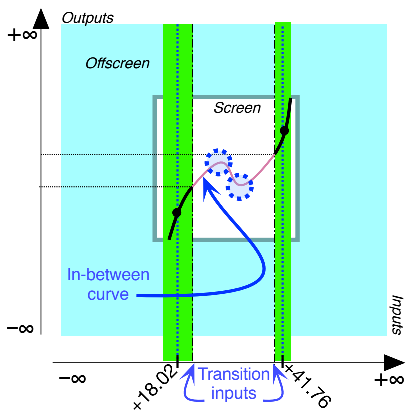
i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
and
iv. there are no jumps at the transitions,
v. the max and the min are *forced* by the given local graphs,
So, this inbetween curve *is* an *essential* smooth interpolation.

Example 3.30. The curve inbetween the local graphs



i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
iv. there are no jumps at the transitions, and
v. the inflection is *forced* by the offscreen graph.
So, this inbetween curve is an *essential* smooth interpolation.

EXAMPLE 3.31. The curve inbetween the local graphs



i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
iv. there are no jumps at the transitions, but
v. the min and the max are *not* forced by the given local graph.
So, this inbetween curve is *not* an *essential* smooth interpolation. (Compare with EXAMPLE 3.52)

==========OK SO FAR ===========

6 Essential Onscreen Graph

The onscreen graph and the offscreen graph are very different in nature and the difference between the offscreen graph and the onscreen graph will be central to the way we will get the global graph of a function given by an 'input-output rule' in Section 1 - Global Input-Output Rules (Page 207).

The essential *difference!*

1. Offscreen vs. onscreen. The main difference between the offscreen graph and the onscreen graph is that:

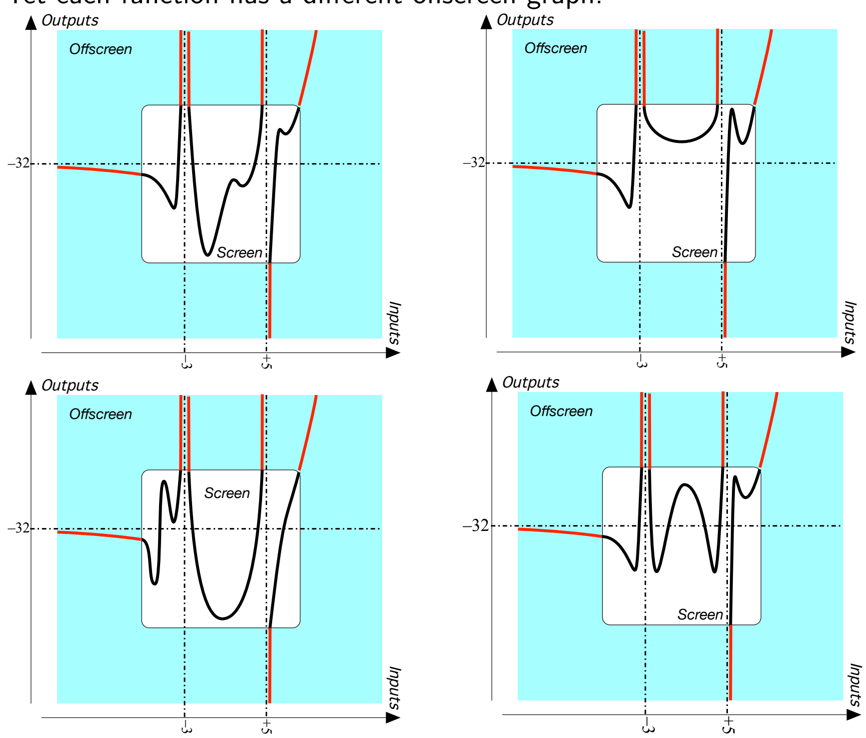
- ► The offscreen graph depends only on:

  - The local graph near ∞
  - The local graph near the pole(s), if any.
- The onscreen graph will depend on local information which will depends very much on the particular function being investigated.

EXAMPLE 3.32. The following functions all have exactly the same offscreen graoh since:

- They all have the same local graphs near ∞
- They all have the same local graphs near the poles -3 and $+5$

Yet each function has a different onscreen graph:

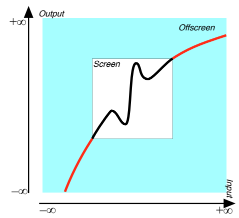


2. Seen from far away. However, the onscreen graphs for a given offscreen graph "*all look the same at sufficiently small scales*" https://www.math.columbia.edu/~abouzaid/ because, seen from further and further away, the features of the onscreen graph which are not

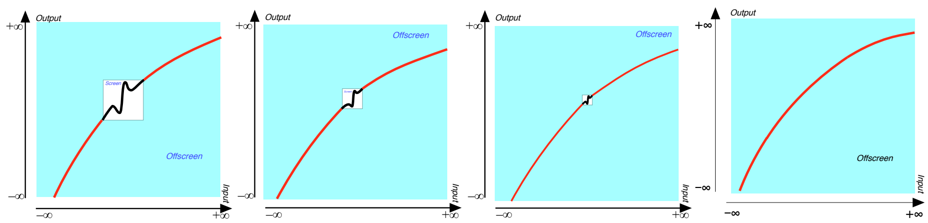
forced

by the offscreen graph become too small to be made out.

EXAMPLE 3.33. Given the global graph,



here is what we see from further and further away:

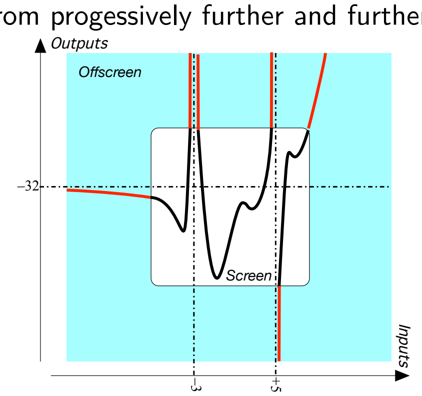


The difference between essential feature changes and non-essential feature changes is that even when seen from very far away, and even if we cannot see the part of the onscreen graph where the feature change actuallly occurs, inasmuch as essentiall feature changes are forced by the offscreen graph, we can infer essential onscreen feature changes from the offscreen graph.

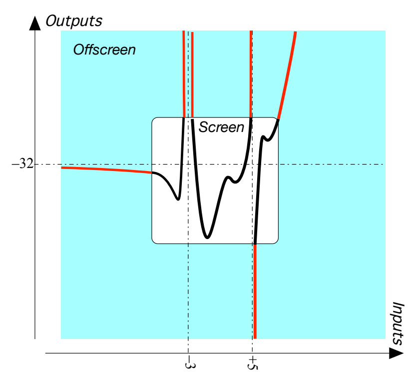
EXAMPLE 3.34.

EXAMPLE 3.35. The first global graph in EXAMPLE 3.40 (Page 188) seen from progessively further and further:

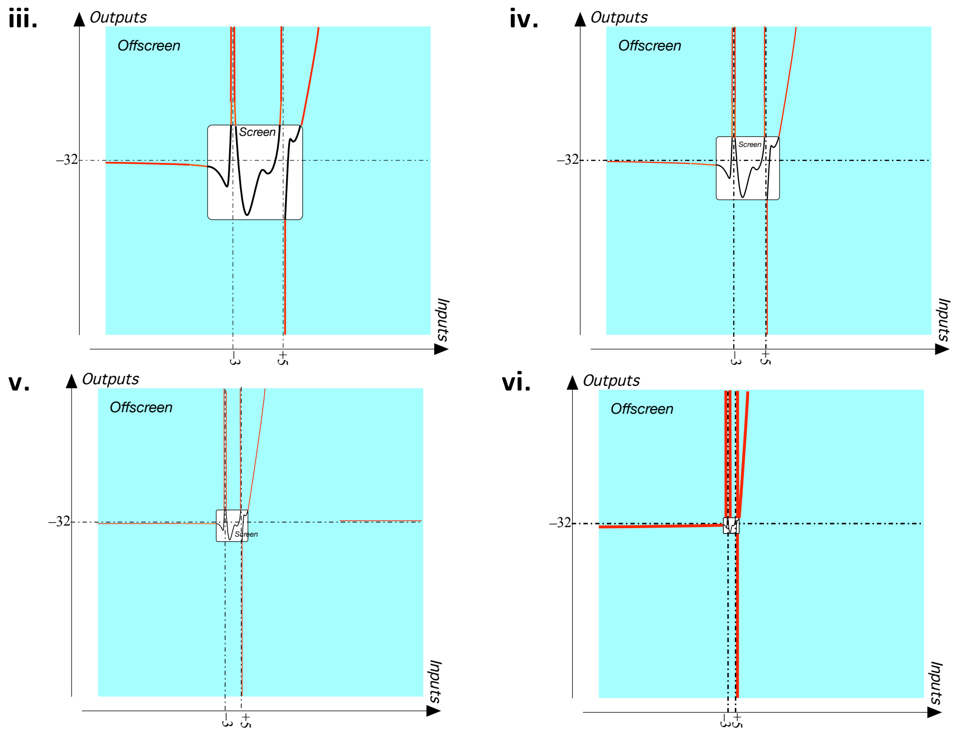
i.



ii.



essential

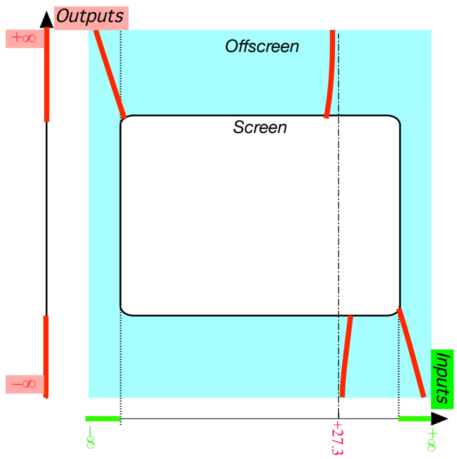


Moreover, since all the global graphs in EXAMPLE 3.40 (Page 188) have the same offscreen graph, we would get the "same" *last* global graph as **vi.** above.

3. Forcing Inasmuch as a function is smooth, the offscreen graph forces the onscreen graph to have local features which we will qualify as **essential**.

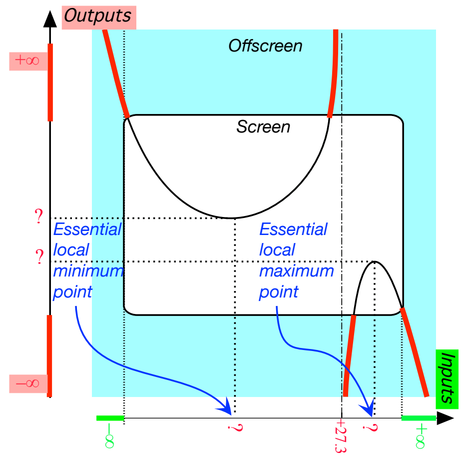
EXAMPLE 3.36.

The offscreen graph (of some smooth function)



forces the onscreen graph to have

(at least) one maximum point and (at least) one minimum point:



but neither *where* the points are located nor *what* the outputs are.

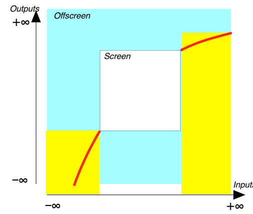
CAUTIONARY NOTE 3.1 *Locating* essential inputs is a totally different question from finding *how many* essential inputs there are. *Locating* essential inputs is usually a much more difficult question which, except in a very few cases, we will not deal with in this text.

Thus, the essential onscreen graph of a smooth function is the global graph with no non-essential feature.

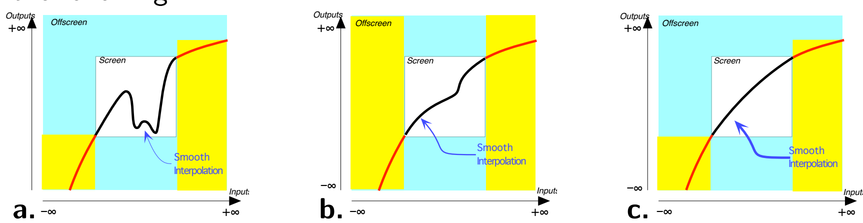
4. Essential interpolation vs Non-essential Interpolation

DEFINITION 3.1 An essential onscreen graphis a **simplest** possible smooth interpolation of the offscreen graph, that is without any *nonessential* feature-sign change inputs and without any *nonessential* features.

EXAMPLE 3.37. Given the offscreen graph,

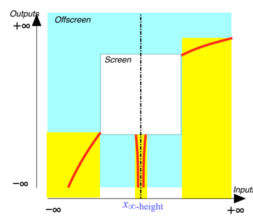


the following

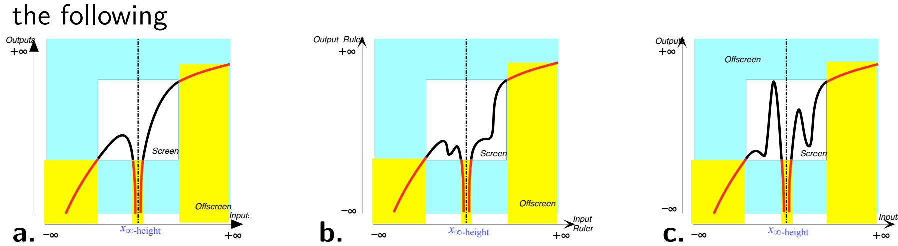


are all smooth interpolations but only **c.** is an *essential* onscreen graph.

EXAMPLE 3.38. Given the offscreen graph,



the following

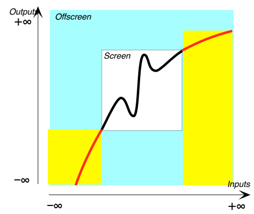


are all smooth interpolations but only **a.** is an *essential* onscreen graph.

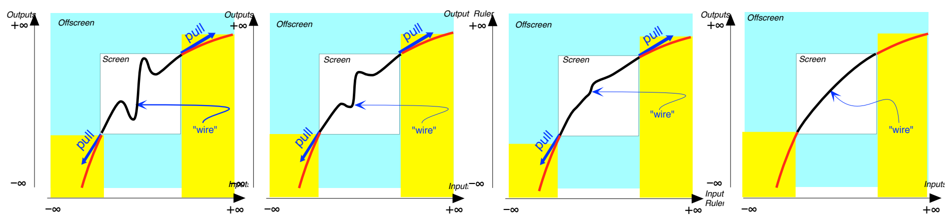
5. Smoothing out non-essential features

- The *essential onscreen graph* is how we see the actual graph from "faraway" inasmuch as nonessential features such as *bumps*, *hiccups* and *fluctuations* are too small to be seen from faraway.
- The *essential onscreen graph* is what we would get if the onscreen graph were a wire being pulled out so as to straighten it.

Example 3.39. Given the global graph,



we can imagine the *non-essential* onscreen graph as a "wire" being pulled by the offscreen graph so as to smooth it out into an *essential* medium-size graph.



=========End WORK ZONE=======

> **Agreement 3.1** From now on we will just say interpolation for essential smooth interpolation.

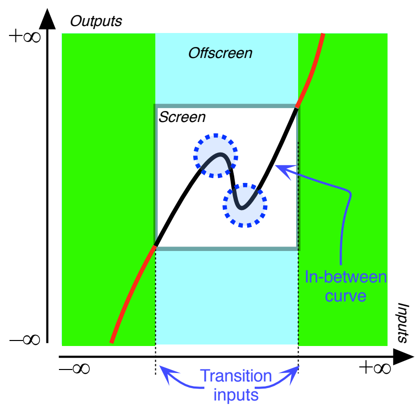
7 Interpolating An Offscreen Graph

Getting the global graph of a function *not* given by a curve is usually not a simple matter (https://en.wikipedia.org/wiki/Interpolation).

As already mentioned in Section 5 - Smooth Interpolations (Page 178), we will use smooth interpolations to get the *global* graph of functions *not* given by a curve which we will do by smoothly interpolating the offscreen graph to get an onscreen graph.

1. Functions without pole. When the function does *not* have a pole, we interpolate smoothly the local graph near ∞ by drawing across the screen an inbetween curve from one transition input to the other transition input.

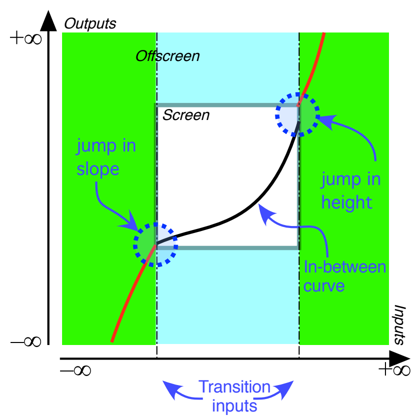
EXAMPLE 3.40. The curve inbetween the offscreen graph



i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
and
iv. there are no jumps at the transitions.
So, this inbetween curve

- gives an onscreen graph,
- does *not* give an *essential* onscreen graph because the min and the max are *not* forced by the offscreen graph

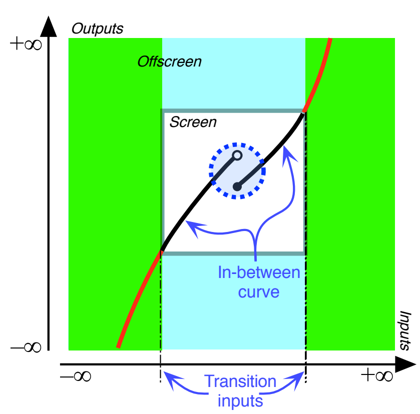
EXAMPLE 3.41. The curve inbetween the offscreen graph



i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
but
iv. there are two jumps at transitions,

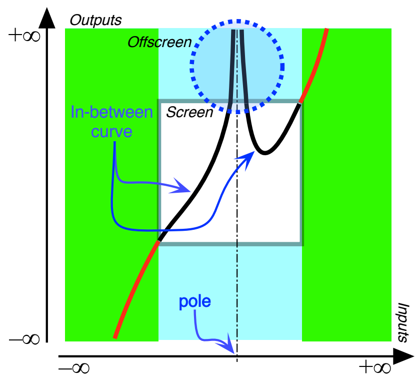
So, this *inbetween curve cannot* the onscreen graph (Even though it *could* be the onscreen graph for a function that is *not* smooth.)

EXAMPLE 3.42. The curve inbetween the offscreen graph



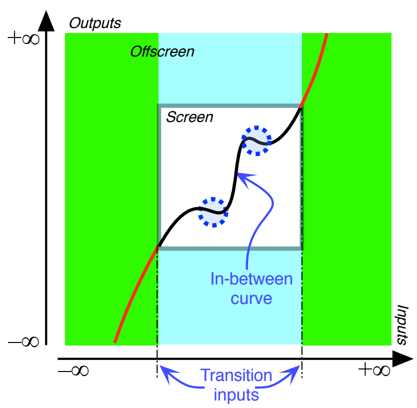
i. is *not* height-continuous,
So: This inbetween curve *cannot* be the onscreen graph.

EXAMPLE 3.43. The curve inbetween the offscreen graph



introduces a **pole**.
So, this inbetween curve *cannot* be the on-screen graph.

EXAMPLE 3.44. The curve inbetween the offscreen graph



i. is height-continuous,
ii. is slope-continuous,
iii. is concavity-continuous,
and
iv. there are no jumps at the transitions

So, this inbetween curve gives an onscreen graph but *not* an *essential* onscreen graph because one of the two **inflections** is not forced by the offscreen graph.

2. Functions with pole(s). When the offscreen graph consists of the local graph near ∞ together with at least the local graph near the pole $x_{\infty\text{-height}}$, we must interpolate with:

- one curve inbetween :
  - ▸ the local graph near $-\infty$
    and
  - ▸ the local graph near the pole $x_{\infty\text{-height}}$,,

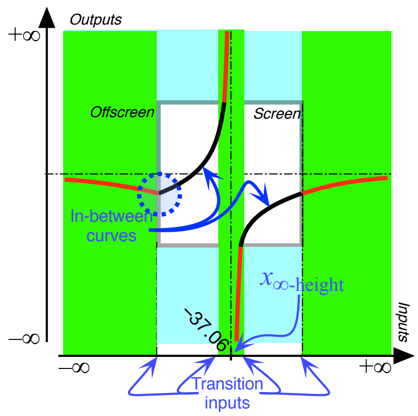
and

- another curve inbetween :
  - ▸ the local graph near the pole $x_{\infty\text{-height}}$,,
    and
  - ▸ the local graph near $+\infty$

In EXAMPLE 3.45 and EXAMPLE 3.46 we will examine whether or not the inbetween curve is an interpolation for the

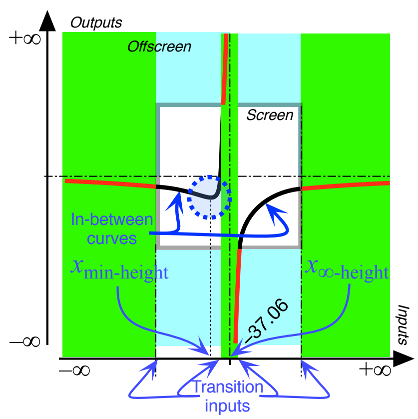
Example 3.45. The curves inbetween the two offcreen local graphs

the *inbetween curve*



i. are both height-continuous,
ii. are both slope-continuous,
iii. are both concavity-continuous,
but
iv. there is a **slope-jump** at one of the transitions.
So, these inbetween curves cannot give an onscreen graph..

Example 3.46. The curves inbetween the two offcreen local graphs



i. are both height continuous,
ii. are both slope continuous,,
iii. are both concavity continuous,
and
iv. there is no jumps at the transitions,.
So, these in-beweern curves give an on-screen graph which is in fact *essential* since the **min** is forced by the offscreen graph.

==========**OK SO FAR** ===========

3. Interpolating an offscreen graph. So, based on the preceding EXAMPLES, to draw an interpolation, we proceed as follows

Procedure 3.1 Interpolate an offscreen graph

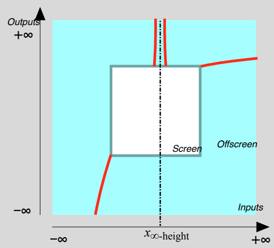
i. Going from left to right, mark the features where the *offscreen graph* **enters the screen** and where the *offscreen graph* **exits the screen**
ii. Draw the inbetween curve(s) from the point(s) where the offscreen graph enters the screen to the point(s) where the offscreen graph exits

essential

the screen making sure that:

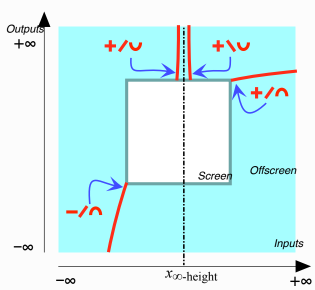
- Each *inbetween curve* is *smooth*,
- Each *transition* between a inbetween curve and the local graph is *smooth*
- The inbetween curves do *not* introduce any *infinite height* input.

DEMO 3.1

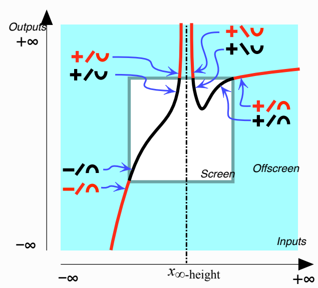


To interpolate,

i. Mark the features where the *offscreen graph* enters and exits the screen:



ii. Draw the inbetween curve(s) *smoothly*



=======End HOLDING =======
=======Begin WORK ZONE=======

8 Essential Feature-Sign Changes Inputs

1. Essential sign-change input A feature sign-change input is **essential** whenever its **existence**is **forced**by the offscreen graph. So, given the offscreen graph of a function, in order

PROCEDURE 3.2 To establish the existence of essential feature sign change inputs in a inbetween curve

i. For each piece of the inbetween curve, check the feature sign at both end of the piece.

- If the feature signs at the two ends of the piece are *opposite*, there *has* to be a feature sign change input for that piece.
- If the feature signs at the two ends of the piece are the *same*, there does *not* have to be a feature sign change input for that piece.

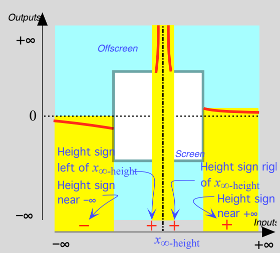
ii. For each ∞ height input, if any, check the feature sign on either side of the ∞ height input:

- If the feature signs on the two sides of the ∞ height input are *opposite*, the ∞ height input *is* a feature sign change input.
- If the feature signs on the two sides of the ∞ height input are the *same*, the ∞ height input *is not* a feature sign change input..

iii. Check the feature sign on the two sides of ∞

- If the feature signs on the two sides of ∞ are *opposite*, ∞ *is* a feature sign change input.
- If the feature signs on the two sides of ∞ are the *same*, ∞ *is not* a feature sign change input..

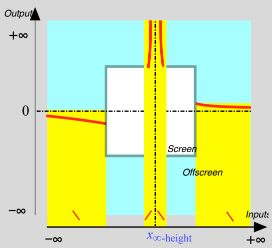
DEMO 3.2a



To establish the existence of Height-sign change inputs

- Since the Height signs near $-\infty$ and left of $x_{\infty\text{-height}}$ are *opposite* there *is* an essential Height sign-change between $-\infty$ and $x_{\infty\text{-height}}$.
- Since the Height signs right of $x_{\infty\text{-height}}$ and near $+\infty$ are *the same* there *is no* essential Height sign-change between $x_{\infty\text{-height}}$ and $+\infty$.

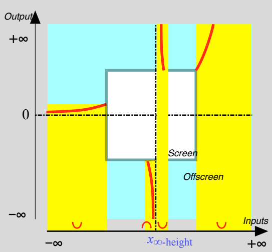
DEMO 3.2b



To establish the existence of Slope-sign change inputs

- Since the Slope signs near $-\infty$ and left of $x_{\infty\text{-height}}$ are *opposite* there *is* an essential Slope sign-change between $-\infty$ and $x_{\infty\text{-height}}$.
- Since the Slope signs right of $x_{\infty\text{-height}}$ and near $+\infty$ are *the same* there *is no* essential Slope sign-change between $x_{\infty\text{-height}}$ and $+\infty$.

DEMO 3.2c

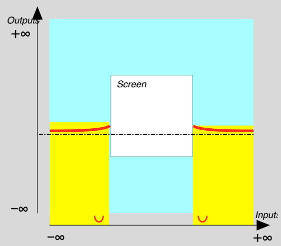


To establish the existence of Concavity-sign change inputs

- Since the Concavity signs near $-\infty$ and left of $x_{\infty\text{-height}}$ are *opposite* there *is* an essential Concavity sign-change between $-\infty$ and $x_{\infty\text{-height}}$.
- Since the Concavity signs right of $x_{\infty\text{-height}}$ and near $-\infty$ are *the same* there *is no* essential Concavity sign-change between $x_{\infty\text{-height}}$ and $+\infty$.

2. more complicated However, things can get a bit more complicated.

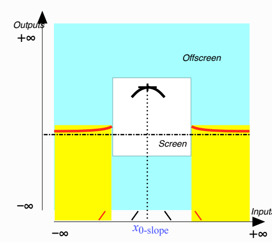
DEMO 3.2d



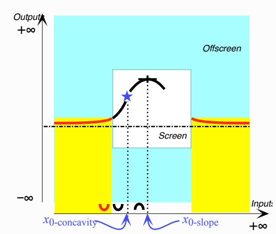
To establish the existence of Concavity-sign change inputs

- Since the concavity-sign at the transitions from $-\infty$ is $\cup$ and the concavity-sign at the transition to $+\infty$ is also $\cup$, one might be tempted to say that there is no essential concavity sign-change input.
- However, attempting a smooth interpolation shows that things are a bit more complicated than would at first appear.

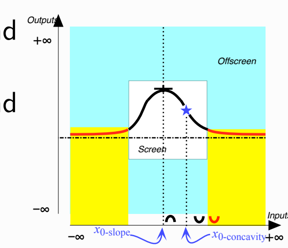
i. Since the slope-signs at the transition *from* $-\infty$ is $\diagup$ and the slope-sign at the transition *to* $+\infty$ is $\diagdown$ there has to be an essential Slope sign-change input near which Concavity sign $= \langle \cap, \cap \rangle$



ii. Since the concavity-signs near $-\infty$ and *left* of $x_{0\text{-slope}}$ are *opposite*, there is an essential Concavity sign-change input between $-\infty$ and $x_{0\text{-slope}}$.



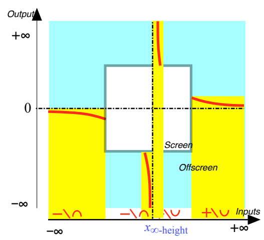
iii. Since the concavity-signs *right* of $x_{0\text{-slope}}$ and near $+\infty$ are *opposite*, there is an essential Concavity sign-change input between $x_{0\text{-slope}}$ and $+\infty$.



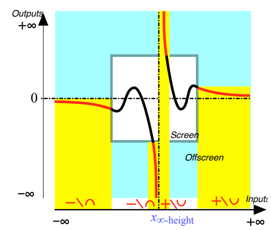
3. non-essential That there is no *essential* feature sign-change input does not mean that there could not actually be a *non-essential* feature sign-change input.

essential␣local␣extreme-height␣input

Example 3.47.
Let f be the function whose offscreen graph is

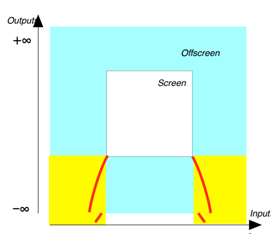


- There is no *essential* Height sign-change input, no *essential* Slope sign-change input, and no *essential* Concavity sign-change input.
- However, the actual medium-size graph could very well be:

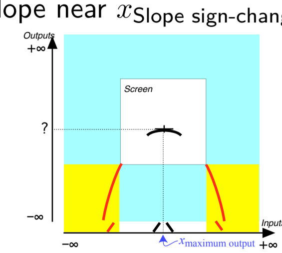


4. Essential Extreme-Height Inputs An extreme-height input is an **essential local extreme-height input** if the existence of the local extreme-height input is forced by the offscreen graph in the sense that *any* smooth interpolation *must* have a local extreme-height input.

Example 3.48.
Let f be a function whose offscreen graph is



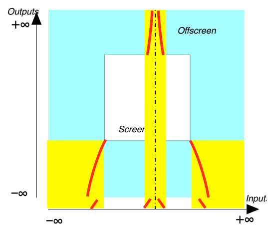
Then,
i. Since the Slope signs near $-\infty$ and $+\infty$ are *opposite* there *is* an essential Slope sign-change between $-\infty$ and $+\infty$.
ii. Since the Height of $x_{\text{Slope sign-change}}$ is not infinite, the slope near $x_{\text{Slope sign-change}}$ must be 0



iii. $x_{\text{0-slope}}$ is a local essential Maximum-Height input.

Example 3.49.

Let f be a function whose offscreen graph is



Then,
i. Since the Slope signs near $-\infty$ and near $+\infty$ are *opposite* there *is* an essential Slope sign-change between $-\infty$ and $+\infty$.
ii. But since there is an ∞-height input, the Height near $x_{slopesign-change}$ is infinite and there is no essential local maximum height input.

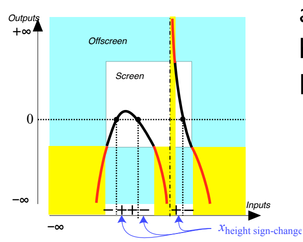
5. Non-essential Features While, as we have just seen, the *offscreen graph* may force the existence of certain feature-sign changes in the *onscreen graph*, there are still many other smooth interpolations of the *offscreen graph* that are not forced by the onscreen graph.

EXAMPLE 3.50. The moon has an influence on what happens on earth—for instance the tides—yet the phases of the moon do not seem to have an influence on the growth of lettuce (see http://www.almanac.com/content/farming-moon) or even on the mood of the math instructor.

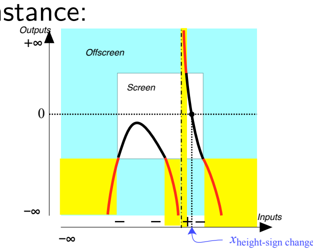
We will say that a global feature is **non-essential**if it is *not* forced by the offscreen graph.

1. As we saw above, feature sign-change inputs can be non-essential.

EXAMPLE 3.51.
Let f be a function whose graph is



Then,
i. The two Height sign-change inputs left of $x_{\infty\text{-height}}$ are non-essential because if the 0-output level line were higher, there would be no Height sign-change input. For instance:



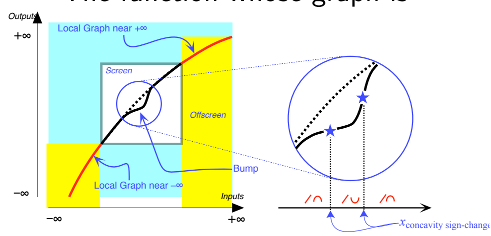
ii. The Height sign-change input right of $x_{\infty\text{-height}}$ is essential because, no matter where the 0-output level line might be, the inbetween curve has to cross it.

bump
wiggle
max-min␣fluctuation
min-max␣fluctuation

2. There other non-essential features:

- A *smooth* function can have a **bump** in which the slope does not change sign but the concavity changes sign twice.

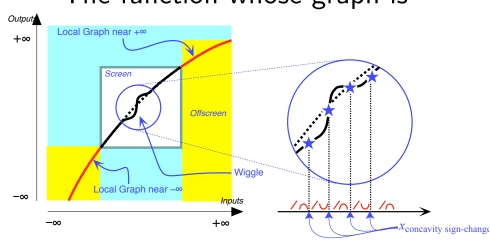
EXAMPLE 3.52. The function whose graph is



has a *bump*.

- A *smooth* function can also have a **wiggle**, that is a pair of bumps in opposite directions with the slope keeping the same sign throughout but with *three* inputs where the concavity changes sign.

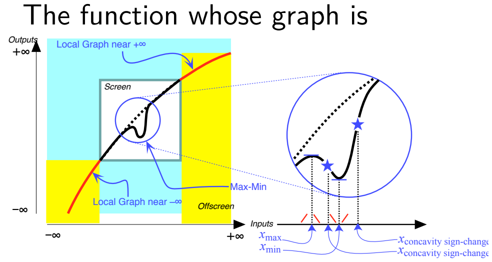
EXAMPLE 3.53. The function whose graph is



has a *wiggle*.

- A *smooth* function can also have a **max-min fluctuation** or a **min-max fluctuation** that is a sort of "extreme wiggle" which consists of a pair of *extremum-heights inputs* in opposite directions. In other words, a fluctuation involves:
  - *two* inputs where the *slope* changes sign
  - *two* inputs where the *concavity* changes sign

EXAMPLE 3.54. The function whose graph is



has a *max-min fluctuation*.

essential onscreen graph
join smoothly
essential graph
join smoothly
essential on-screen graph

However, as we will see in Section 1 - Global Input-Output Rules (Page 207), in Mathematics, functions are not usually given by a curve but are given "mathematically" and the investigation of how a function given "mathematically" behaves *cannot* be based on the function's global graph which, in any case, is usually not necessarily simple to get as we will discuss in Section 4 - Local Input-Output Rule (Page 214).

But, while finding the global graph of a function given "mathematically" is not stricly necessary to *understand* how the given function behaves, the global graph of a function given "mathematically" *can* be a very great help to *see* the way the given function behaves.

So, in order to explain how we will get the global graph of a function given "mathematically" we will have to proceed by stages using functions given by a curve.

We begin by outlining the PROCEDURE which we will follow in Section 1 - Global Input-Output Rules (Page 207).

i. The first step in getting the global graph of a function given "mathematically" will be to get the local graphs near the control points, that is near ∞ and near the poles, if any.

ii. The second step in getting the global graph of a function given "mathematically" will be to get the offscreen graph.

iii. The third step in getting the global graph of a function given "mathematically" will be to get the **essential onscreen graph** by **joining smoothly** the offscreen graph across the screen.

6. The essential onscreen graph. Thus, the first step in getting the global graph of a function given by an I-O rule will be to get the **essential graph**, that is the onscreen graph forced by the offscreen graph, in other words, the onscreen graph as we would see it from very far away.

PROCEDURE 3.3 To get the essential graph of a function given by a global input-output rule

i. Get the offscreen graph, that is,
  a. Get the local graph near ∞,
  b. Get the local graph near the pole(s), if any,
ii. **Join smoothly** the offscreen graph across the screen

Get the offscreen graph from the local graphs near the control inputs namely near ∞ and near the pole(s) if any,

Then get the ***essential*** **on-screen graph** by

And just because something is far away doesn't mean it's of no interest: "Many ancient civilizations collected astronomical information in a systematic manner through observation." *(https://en.wikipedia.org/wiki/History_of_science.)*

existence

proximate on-screen graph

The *essential* on-screen graph will already provide information about the **existence** of *essential* behavior change inputs *on-screen*—but *not* about their location.

However, there might be behavioral changes too small to see from far away, so get the **proximate on-screen graph** by:

a. locating the non-essential behavioral change inputs, if any,

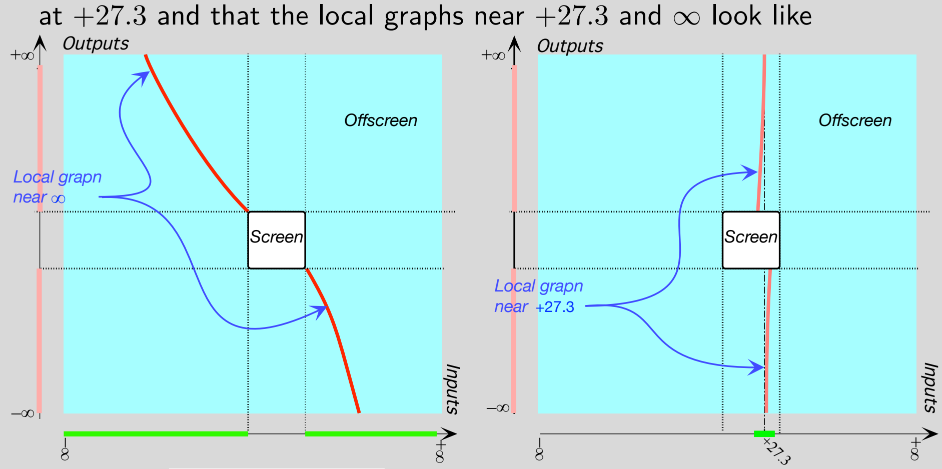
b. getting the local graphs near these non-essential behavioral change inputs

c. Joining smoothly *all* local graphs,

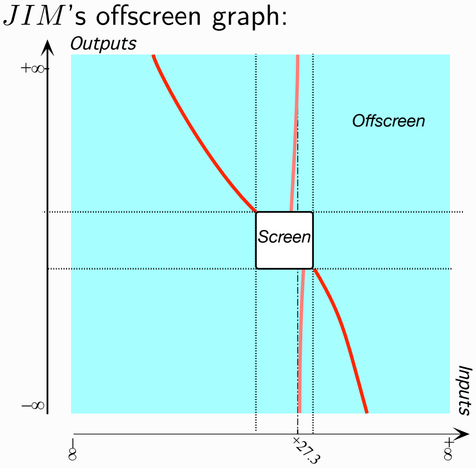
Actually makes sense doesn't it?

and then progressively zero in:

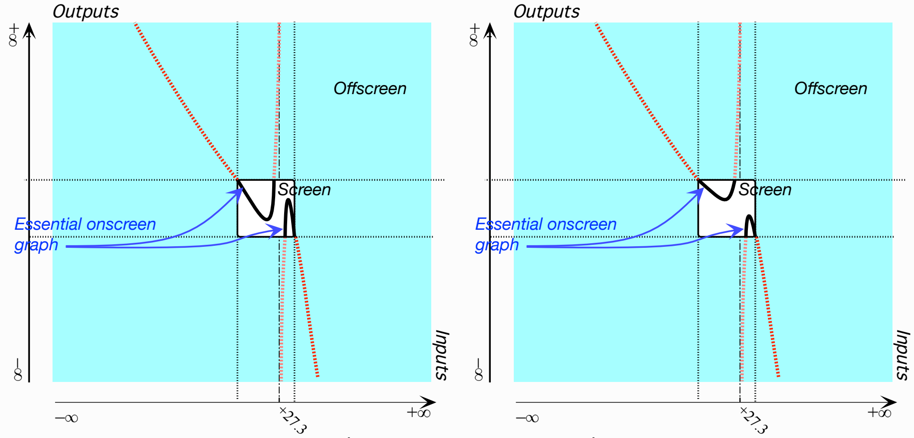
Demo 3.3 Suppose that we found that the function JIM has a pole at $+27.3$ and that the local graphs near $+27.3$ and ∞ look like



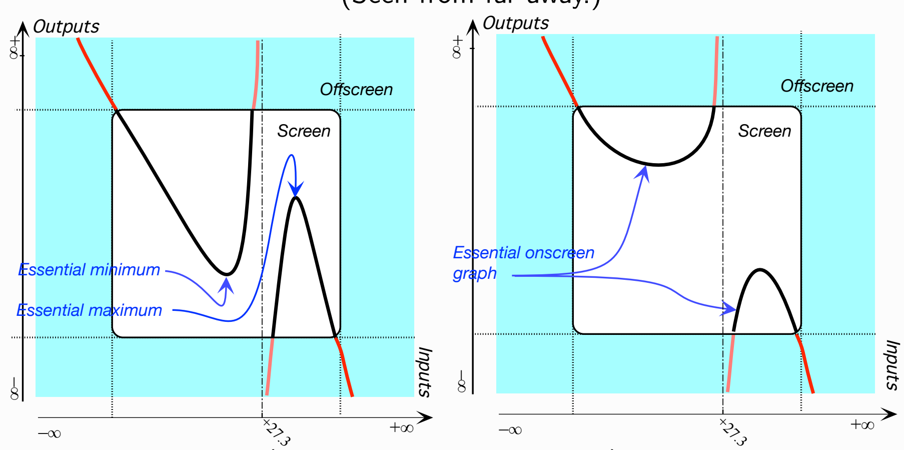
I. We then have JIM's offscreen graph:



II. Then, the essential on-screen graph of JIM would look something like either one of:

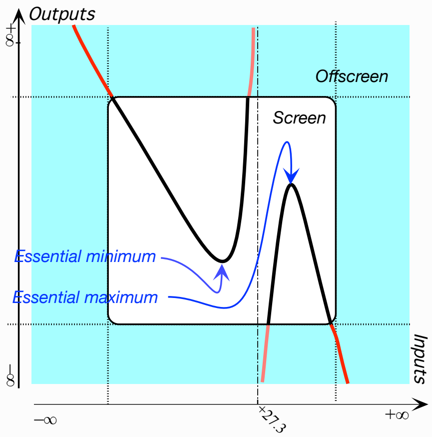


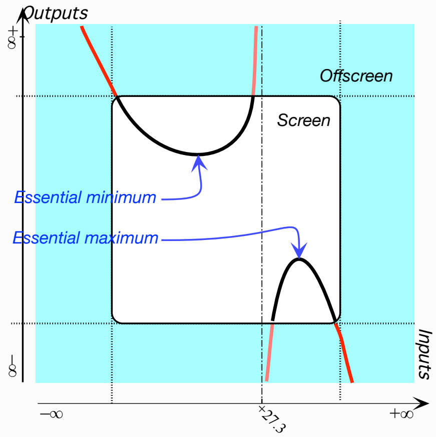
(Seen from far away.)



(Seen from a short distance.)

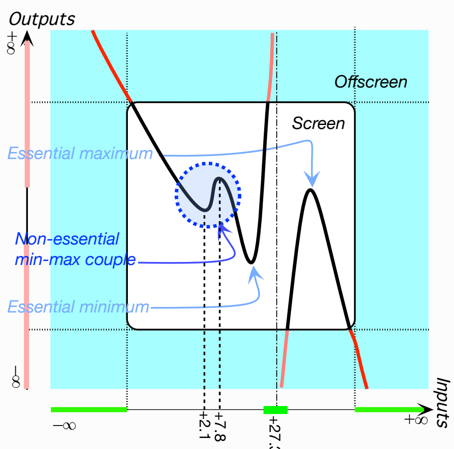
Under the assumption that the function is continuous, the *essential* on-screen graph of JIM already shows, the existence of an *essential* minimum and of an *essential* maximum:

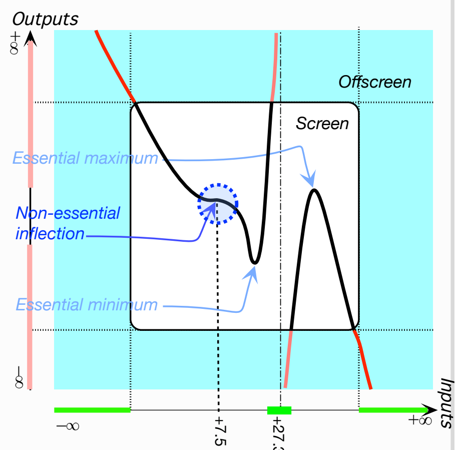




However, there might be behavioral changes that we are too far to see.

III. Supposing, for instance, that we were to locate from the I-O rule a maximum at $+7.8$ coupled with a minimum at $+2.1$, or a non-essential inflection at $+7.5$ then the proximate graph would be:





If we were to locate *no* non-essential behavioral change input, then of course the proximate graph would just be the essential on-screen graph

9 Dilation of Functions

1. Dilating functions given by table.

2. Dilating functions given by curve.

base function
add-on number
add-on function
sum function

10 Addition of Functions

Given a function, to which we will refer as **base function**, one often needs to add a number to each output that the base function returns. Whether or not this **add-on number** remains the same regardless of the input or differs depending on the input, we can look upon the add-on number as being itself the output returned for the same input by some other function to which we will refer as **add-on function**. (If the add-on number is the same regardless of the input this just means that the add-on function is a *constant function*.)

There is then going to be a third function, to be referred as **sum function**, which, for each input, will return the output returned by the base function plus the add-on number returned by the add-on function for that input.
In other words, given the two functions

$$x \xrightarrow{BASE} BASE(x)$$

and

$$x \xrightarrow{ADD\text{-}ON} ADD\text{-}ON(x)$$

there will be a third function given by

$$x \xrightarrow{SUM} SUM(x) = BASE(x) + ADD\text{-}ON(x)$$

1. Adding functions given by table. In sciences such as Biology, Psychology and Economics the three functions are often in *tabular* form.

Example 3.55. When we shop online for, say a textbook, we first see a *price list*—the *base function*. However, a *shipping charge*, which might or might not depend on the textbook, is usually added-on to the *list price* and is given by the *Shipping charge list*—the *add-on function*. The price we end-up having to pay is thus given by the *actual price list*—the *sum function*.

$x \xrightarrow{\mathcal{LIST}}$	$\mathcal{LIST}(x)$
English	140
History	80
Biology	130
Math	10
Poetry	70

$x \xrightarrow{\mathcal{SHIP}}$	$\mathcal{SHIP}(x)$
English	13.15
History	3.45
Biology	7.25
Math	3.75
Poetry	5.32

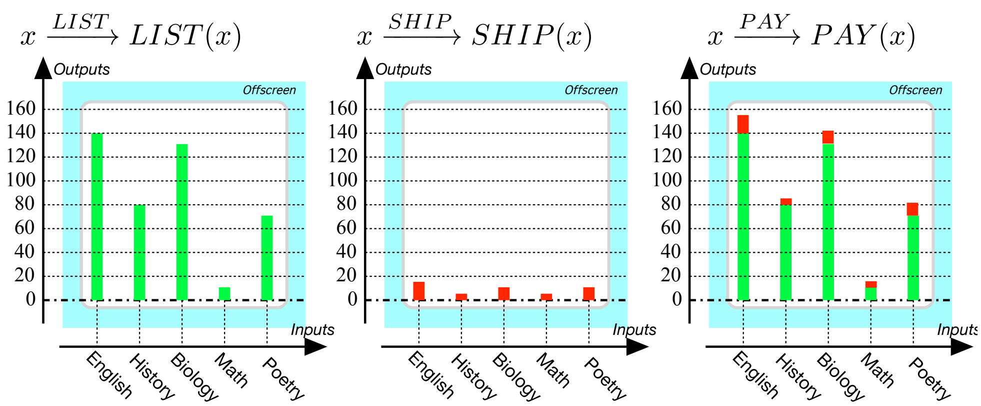
$x \xrightarrow{\mathcal{PAY}}$	$\mathcal{PAY}(x)$
English	140+13.15 = 153.15
History	80+3.45 = 83.45
Biology	130+7.35 = 137.25
Math	10+3.75 = 13.75
Poetry	70+5.32 = 75.32

which says, for instance, that while the *list price* of the English textbook is

\$140 , a *shipping charge* of \$13.15 brings the price to be *paid* to \$140 + \$13.15 = \$153.15.

2. Adding functions given by bar graphs. Instead of giving the functions by *tables*, one might want to given them by *graphs*. Rather than to use *plots*, though, one often uses bar graphs.

EXAMPLE 3.56. The situation in the above example would be represented by the following bar graphs.



11 Linear Combinations of functions

Part II

Functions Given By Rules

While functions given by data, be it tables, I-O plots or curves, are often used by *experimental* scientists, functions given by data do not lend themselves to the calculations necessary to, for instance, assess the precision of the data and, more generally, to a real understanding of what the function does.

So, both in engineering and sciences, functions are mostly given "mathematically", that is by giving a generic expression to *calculate* the output number $f(x)$ to be returned by f for the input number x.

Giving a number (Subsection 4.7, Page 17) makes it likely that there are several ways to give a generic expression for $f(x)$ and this **Part II** deals with the first and simplest way which is of course just to give the generic expression in terms of x for $f(x)$ *itself*.

After introducing global Input-Output rules in Chapter 5 - Straight Functions (Page 227), we will discuss:

1. **Local I-O Rule**
2. **Local Graph**
3. **Local Features**
4. **Control Point(s)**
5. **Global Graph**

Of course, most of **Part II** will belabor the obvious but, this way, the reader will see how matters remain essentially the same even as the functions get more complicated.

globlal I-O rule
pointwise format

Chapter 4

Input-Output Rules

1 Global Input-Output Rules

To give a function, the simplest way after giving the outputs *themselves* by way of data is to give instructions for *computing* the outputs:

This is the "direct" way to give instructions. In Part III - (Laurent) Polynomial Functions (Page 337) we will see the "reverse" way, that is giving the instructions as 'solution of a (function)'equation'.

DEFINITION 4.1 A **global I-O rule** provides a generic expression in terms of x for computing the output $f(x)$ in terms of the input x:

$$\underbrace{x}_{\text{Input}} \xrightarrow{\;f\;} \underbrace{\underbrace{f(x)}_{\text{Output}} = \text{generic expression in terms of } x}_{\text{Global input-output rule}}$$

In order to work with input-output pairs *at* a given number x_0 when the function is given by an Input-Output rule, we will use

DEFINITION 4.2 Depending on whether x_0 is a regular input or a pole we will use the following formats:

- For *graphing*, the pair format,

$\left(x_0 , y_0 \right)$ or $\left(x_0 , \infty \right)$

- For *computing*, the plain function format,

$f\left(x_0\right) = y_0$ or $f\left(x_0\right) = \infty$

- For *seeing*, the arrow function format,

$x_0 \xrightarrow{f} y_0$ or $x_0 \xrightarrow{f} \infty$

- For *anything*, the full function format,

$x_0 \xrightarrow{f} f\left(x_0\right) = y_0$ or $x_0 \xrightarrow{f} f\left(x_0\right) = \infty$

2 Output *at* a given number.

Even though, as we argued in Section 11 - Neighborhoods (Page 40), evaluating a generic expression *at* a given number is to ignore the real world, we *will* occasionally, if only for plotting purposes, want to get the outputs *at* given numbers.

1. Getting input-output pairs

PROCEDURE 4.1 To get the output returned *at* a given *number* x_0 by a function f given by an I-O rule $x \xrightarrow{f} f(x) =$ generic expression in terms of x,

In standard CALCULUS *texts the two steps, declaration and replacement, are often conflated into a single step but we will not do it.*

i. Declare that the output is to be *at* the given x_0 by writing to the right of the global input-output rule: the declaration $\Big|_{x \leftarrow x_0}$:

$$x \xrightarrow{f} f(x) = \text{generic expression in terms of } x \Big|_{x \leftarrow x_0}$$

ii. Replace every occurence of x in the global input-output rule by the given input x_0 to get the individual expression for x_0:

$$x_0 \xrightarrow{f} f(x_0) = \text{Individual expression in terms of } x_0$$

iii. Execute the individual expression in terms of x_0, that is do the operations in the individual expression to get:

$$\text{Individual expression in terms of } x_0 = y_0$$

iv. Format the input-output pair according to DEFINITION 4.2 - Point-wise formats (Page 207)

DEMO 4.1a To get the output returned for the input -5 by the function $\mathcal{JILL}$ given by the global input-output rule $x \xrightarrow{\mathcal{JILL}} \mathcal{JILL}(x) = \frac{(-4 \odot x) \oplus +7}{+2 \odot (x \oplus +7)}$

i. We declare that the output is to be *at* -5 by writing $\Big|_{x \leftarrow -5}$ to the right ot the global input-output rule:

$$x \xrightarrow{\mathcal{JILL}(} \mathcal{JILL}(x) = \frac{(-4 \odot x) \oplus +7}{+2 \odot (x \oplus +7)}\Bigg|_{x \leftarrow -5}$$

ii. We replace every occurence of x in the global input-output rule by the given input -5 to get the individual expression for the output at -5:

$$-5 \xrightarrow{JILL} \mathcal{JILL}\left(-5\right) = \frac{\left(-4 \odot -5\right) \oplus +7}{+2 \odot \left(-5 \oplus +7\right)}$$

iii. We execute the individual expression in terms of -5, that is we do the operations in the individual expression:

$$= \frac{(+20) \oplus +7}{+2 \odot (+2)}$$

$$= \frac{+20 \oplus +7}{+2 \odot +2}$$

$$= \frac{+27}{+4}$$

$$= +6.75$$

So, -5 is a regular input of the function $\mathcal{JILL}$

iv. We format the input-output pair, that is we write,

- For *graphing*, $\left(-5, +6.75\right)$
- For *computing* purposes, we write $\mathcal{JILL}(-5) = +6.75$

OK, don't worry too much about the algebra: the idea for this DEMO *and for the* EXAMPLES *that will follow is only to impress you with the power and the scope of* PROCEDURE *1.8. So, for the time being, the most important for you is to develop an appreciation of jusr the way* PROCEDURE *1.8 works.*

- For *discussing*, $-5 \xrightarrow{\mathcal{JILL}} +6.75$
- For *anything*, $-5 \xrightarrow{\mathcal{JILL}} \mathcal{JILL}(-5) = +6.75$

DEMO 4.1b To get the output returned for the input -7 by the function $\mathcal{JILL}$ given by the global input-output rule $x \xrightarrow{\mathcal{JILL}} \mathcal{JILL}(x) = \dfrac{(-4 \odot x) \oplus +7}{+2 \odot (x \oplus +7)}$

i. We declare that the output is to be *at* -7 by writing $\Big|_{x \leftarrow -7}$ to the right ot the global input-output rule:

$$x \xrightarrow{\mathcal{JILL}(} \mathcal{JILL}(x) = \frac{(-4 \odot x) \oplus +7}{+2 \odot (x \oplus +7)}\Bigg|_{x \leftarrow -5}$$

ii. We replace every occurence of x in the global input-output rule by the given input -7 to get the individual expression for the output at -7:

$$-7 \xrightarrow{JILL} \mathcal{JILL}\big(-7\big) = \frac{\big(-4 \odot -7\big) \oplus +7}{+2 \odot \big(-7 \oplus +7\big)}$$

iii. We execute the individual expression in terms of -7, that is we do the operations in the individual expression:

$$= \frac{(+28) \oplus +7}{+2 \odot (0)}$$

$$= \frac{+28 \oplus +7}{+2 \odot 0}$$

$$= \frac{+35}{0}$$

$$= \infty$$

So, -7 is a pole of the function $\mathcal{JILL}$

iv. We format the input-output pair, that is we write,

- For *graphing*, $\big(-5, \infty\big)$
- For *computing* purposes, we write $\mathcal{JILL}(-5) = \infty$
- For *discussing*, $-5 \xrightarrow{\mathcal{JILL}} \infty$

▸ For *thinking*, $-5 \xrightarrow{\mathcal{JILL}} \mathcal{JILL}(-5) = \infty$

3* Global Graph: Why Not Just Plot & Interpolate?

One of the main goals when a function f is given by a global I-O rule ise to get the global graph of f and the way readers were most probably taught was to:

a. Declare a few (regular) input numbers: x_0, x_1, x_2, etc

b. Compute the outputs $f(x_0)$, $f(x_1)$, $f(x_2)$, etc using PROCEDURE 4.1 - Get the output *at* x_0 from the I-O rule giving f (Page 208),

c. Plot the input-output pairs $\big(x_0, f(x_0)\big)$, $\big((x_1, f(x_1)\big)$, $\big((x_2, f(x_2)\big)$, etc using Picture a few pairs (PROCEDURE 1.1, Page 73)

d. Interpolate these plot dots with a curve.

Nice and easy, eh? In fact, utterly misleading garbage that has to be disposed of properly before any CALCULUS *can be done ...* ACCORDING TO THE REAL WORLD.

Unfortunately, if this is indeed fairly easy, this worls only for Straight Functions (Chapter 5, Page 227) because of a number of issues:

1. How do we get the off-screen graph? Since the only numbers we can tickmark are *mid-size* numbers, there is no way we can plot input-output pairs near ∞ and/or near the pole(s) if any.

Of course. we never asked that *from our teacher.*

2. How do we know *which* numbers to declare?

EXAMPLE 4.1. Let $\mathcal{KEN}$ be given by the I-O rule $x \xrightarrow{\mathcal{KEN}} \mathcal{KEN}(x) = -3x^{+2}$

i. If we use the input numbers -2 and $+2$, using PROCEDURE 4.1 - Get the output *at* x_0 from the I-O rule giving f (Page 208) we get the I-O pairs $\big(-2, -12\big)$ and $\big(+2, -12\big)$ whose plot points we could join smoothly with a *horizontal* straight line.

ii. If, however, we use, say, the input numbers -2 and $+4$, using PROCEDURE 4.1 - Get the output *at* x_0 from the I-O rule giving f (Page 208) we get the I-O pairs $\big(-2, -12\big)$ and $\big(+4, -48\big)$ whose plot points we certainly cannot join smoothly with a *horizontal* straight line.

3. How do we know *how many* numbers to declare?

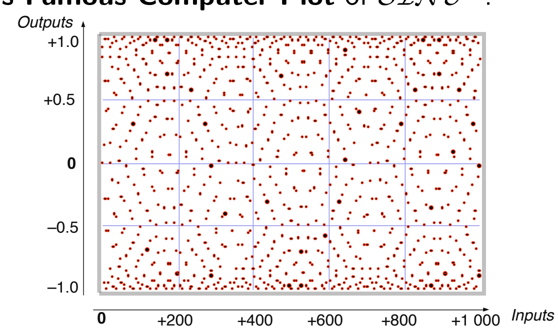
EXAMPLE 4.2. Let $\mathcal{KEN}$ be given by the I-O rule $x \xrightarrow{\mathcal{KEN}} \mathcal{KEN}(x) = -3x^{+2}$

i. If we use two input numbers, say -4 and $+3$, using PROCEDURE 4.1 - Get the output *at* x_0 from the I-O rule giving f (Page 208) we get the I-O pairs $(-4, -48)$ and $(+3, +27)$ whose plot points we can join smoothly with a straight line.

ii. If, however, we use three input numbers, say -4 and $+3$ with some thired input number, the chances are very high that we will not be able to join the plot points smoothly with a straight line.

And in fact, too many plot points can make it impossible to join them smoothly.

EXAMPLE 4.3. The function $\mathcal{SINE}$ belongs to VOLUME II, but the point here is **Strang's Famous Computer Plot** of $\mathcal{SINE}$ [1]:



How are we to "join smoothly"?

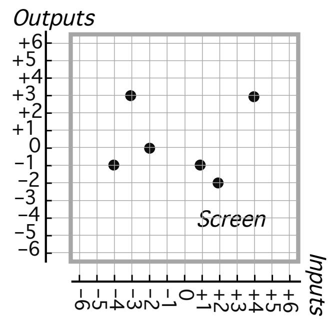
We would sometimes ask but they always said it would be obvious.

4. How do we know with which curve to interpolate?

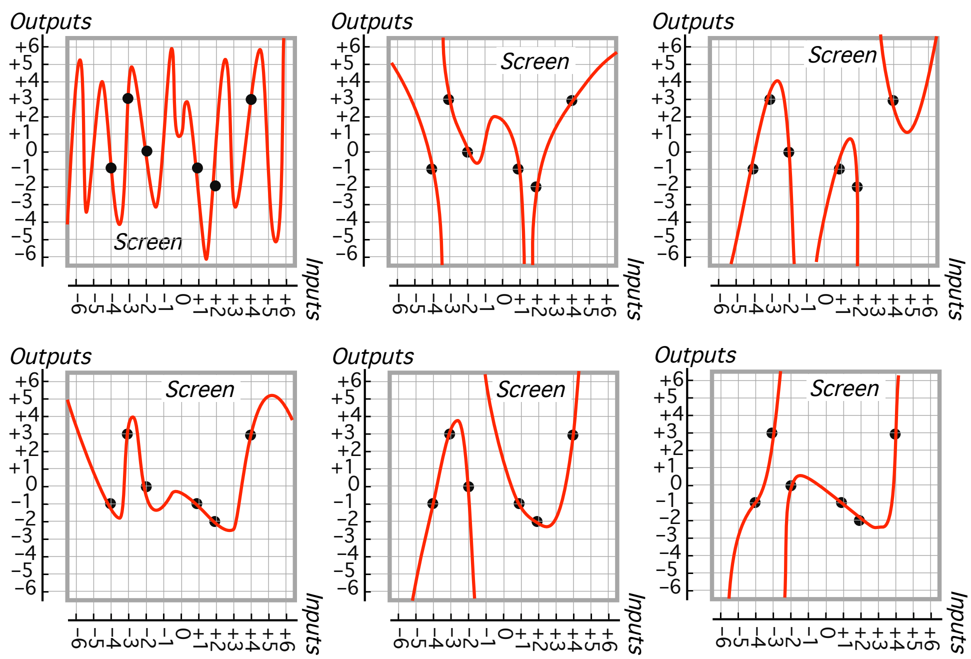
EXAMPLE 4.4. Suppose the function $\mathcal{RINO}$ was given by some *I-O rule* and that we got the following input-output table and therefore the following plot:

[1] The plot appears on the back cover of Strang's *Calculus*, 1991, Wellesley-Cambridge Press, where it is discussed in Section 1.6 A Thousand Points of Light, pages 34-36.

Inputs	-4	-3	-2	$+1$	$+2$	$+4$
Outputs	-1	$+3$	0	-1	-2	$+3$



Now, how would you join these plot dots? For instance:



Answer: Other than making sure not to break the **??** (**??** **??**, **??**) and other than very exceptionally, there is no rule for joining plot dots smoothly.

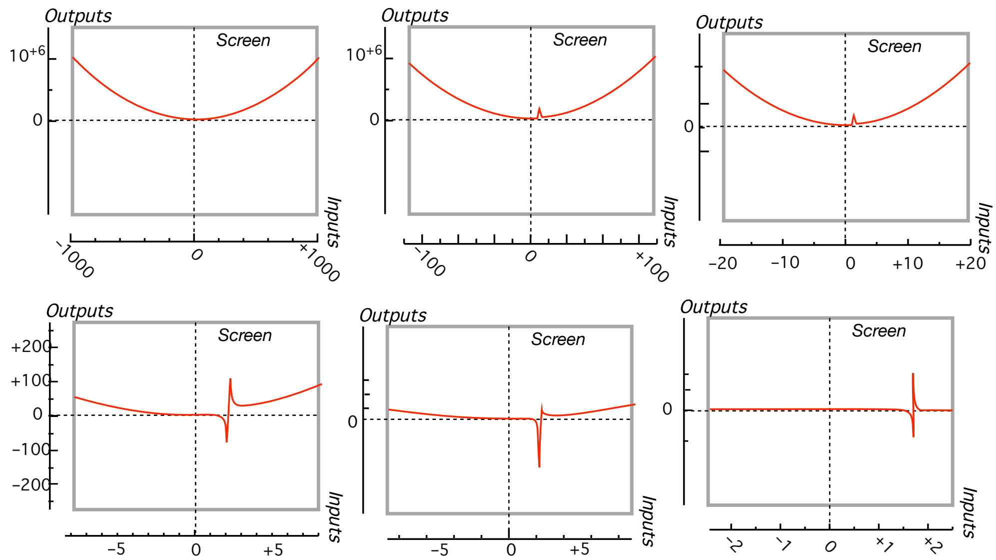
If we asked, they would say "just get more plot points".

5. How do we know the curve we got *is* the graph?

Example 4.5. Let CAT be the function given by the I-O rule

$$x \xrightarrow{\;CAT\;} CAT(x) = \frac{x^3-1}{x-2}$$

Which of the following *computer-generated* "graphs" is the right one?



In case you wonder, they all are. "Just" different scales!

Answer: On the basis of *only* so many plot dots, we can *never* be sure what even the on-screen graph is going to look like.

4 Local Input-Output Rule

OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR
OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR

We already discussed in Numbers In General (Section 3, Page 7) why, in the real world, we cannot use isolated numbers and in Computing with Extended Numbers (Section 10*, Page 39) that we need neighborhoods.

In Local graph near a point (Section 8, Page 119), we saw how to get global graphs from local graphs near control points/

Here, we will see that to get the local graphs we need from ***Local* input-Output rules** to get outputs *near* a given point.

Local input-Output rule
local input-output pair
local input-output rule
local arrow pair

from which we will get local graphs which we will interpolate to get global graphs.

make a diagram here.

alluded to the heart of the matter in Computing with Extended Numbers (Section 10*, Page 39)

As hinted at in Local graph near a point (Section 8, Page 119), the way we will operate is by interpolation of local graph graphs.

The iwuestion then is how to get the local graph near a given point for the global I-O rule, that is how to comput outputs near given numbers.

with computing outputs at given numbers is that:

OKsoFAR **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR**
OKsoFAR **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR**

A major part of our work with functions given by input-output rules will be getting local graphs in order to:

- See Local Features Functions May Have (Chapter 2, Page 133)
- Construct the global graph of the function to see Global Ways Functions May Behave (Chapter 3, Page 165)

The first step towards getting local graphs for functions given by input-output rules will be to compute the output near a given point.

The fact that global input-output rules involve a generic expression in terms of a *number* will *not* prevent us from investigating a function near a given point, be it ∞, 0, or x_0 because,

- near ∞, we will use large-size numbers and therefore the large variable L
- near 0 we will use small-size numbers and therefore the small variable h
- near x_0 we will use nearby mid-size numbers and therefore the near mid-size number variable $x_0 \oplus h$

DEFINITION 4.3 Using the symbol V to stand for the appropriate one of the nearby variables for the given point: large variable L, small variable h, circa variable $x_0 \oplus h$, we have:

- For *graphing*, use the **local input-output pair**

$$\Big(V, \text{executed expression in terms of } V\Big)$$

- For *computing*, use the **local input-output rule**

$$\underbrace{f(V) = \text{executed expression in terms of } V}_{\text{Local input-output rule near given point}}$$

- For *seeing*, use the **local arrow pair**

$$V \xrightarrow{f} \text{executed expression in terms of } V$$

- For *thinking*, use

$$V \xrightarrow{f} \underbrace{f(V) = \text{executed expression in terms of } V}_{\text{Local input-output rule near given point}}$$

1. Near ∞

PROCEDURE 4.2 To get the outputs returned *near* ∞ by a function f given by an I-O rule $x \xrightarrow{f} f(x) =$ generic expression in terms of x,

i. Declare that the input is a large-size indeterminate number by using the large variable L and writing the declaration $\Big|_{x \leftarrow L}$ to the right of the global input-output rule:

$$x \xrightarrow{f} f(x) = \text{generic expression in terms of } x\Big|_{x \leftarrow L}$$

ii. Replace every occurence of x in the global input-output rule by the large variable L to get the **local input-output rule** *near* ∞:

$$L \xrightarrow{f} f(L) = \text{generic expression in terms of } L$$

iii. Execute the generic expression in terms of the relevant variable according to the rules in Section 9 - Computing with Qualitative Sizes (Page 33), that is do the operations in the generic expression to get the **executed expression**

iv. Format according to DEFINITION 4.3 - Local formats (Page 216)

local executed expression
local input-output rule
local input-output pair
local input-output arrow pair

DEMO 4.2 To get the outputs returned for inputs *near* ∞ by the function $\mathcal{ZENA}$ given by the global input-output rule $x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \dfrac{x^{+2} \ominus +1}{x \oplus +3}$

i. We *declare* that the input is a large-size indeterminate number by writing the declaration $\Big|_{x \leftarrow L}$ to the right of the global input-output rule:

$$x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \frac{x^{+2} \ominus +1}{x \oplus +3}\Bigg|_{x \leftarrow L}$$

ii. We *replace* every occurence of x in the generic expression by L to get the individual expression for L:

$$L \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(L) = \frac{L^{+2} \ominus +1}{L \oplus +3}$$

iii. We *execute* the individual expression for L:

$$= \frac{L^{+2} \ominus +1}{L \oplus +3}$$

$$= \frac{L^{+2} \oplus [...]}{L \oplus [...]}$$

$$= L \oplus [...]$$

The last expression above *is* the executed expression.

iv. We *format* according to DEFINITION 4.3 - Local formats (Page 216)

- local Input-output pair $\big(L, L \oplus [...]\big)$
- local input-output rule $\mathcal{ZENA}\big(L\big) = L \oplus [...]$
- local arrow pair $L \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\big(L\big)$

executed expression
local input-output rule
local input-output pair

- local input-output rule $L \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\big(L\big) = L \oplus [...]$

2. Near 0

PROCEDURE 4.3 To get the outputs returned *near* 0 by a function f given by an I-O rule $x \xrightarrow{f} f(x) =$ generic expression in terms of x,

i. Declare that the input is a small-size indeterminate number by using the small variable h and writing the declaration $\big|_{x \leftarrow h}$ to the right of the global input-output rule:

$$x \xrightarrow{f} f(x) = \text{generic expression in terms of } x \Big|_{x \leftarrow h}$$

ii. Replace every occurence of x in the global input-output rule by the small variable h to get the **local input-output rule** *near* 0:

$$h \xrightarrow{f} f\big(h\big) = \text{generic expression in terms of } h$$

iii. Execute the generic expression in terms of the relevant variable according to the rules in Section 9 - Computing with Qualitative Sizes (Page 33), that is do the operations in the generic expression to get the **executed expression**

iv. Format according to DEFINITION 4.3 - Local formats (Page 216)

- For *graphing*, use the input-output pair
$$\Big(h, \text{executed expression in terms of } h\Big)$$
- For *computing*, use the equality
$$\underbrace{f\big(h\big) = \text{executed expression in terms of } h}_{\text{Local input-output rule near } 0}$$
- For *seeing*, use the arrow pair
$$h \xrightarrow{f} \text{executed expression in terms of } h$$
- For *thinking*, use the **local input-output rule**

$$h \xrightarrow{f} \underbrace{f\Big(h\Big) = \text{executed expression in terms of } h}_{\text{Local input-output rule near } 0}$$

executed expression
local input-output rule
local input-output pair

DEMO 4.3 To get the outputs returned for inputs *near* 0 by the function $\mathcal{ZENA}$ given by the global input-output rule $x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \dfrac{x^{+2} \ominus +1}{x \oplus +3}$

i. We *declare* that the input is a small-size indeterminate number by using the small variable h and writing the declaration $\Big|_{x \leftarrow h}$ to the right of the global input-output rule:

$$x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \frac{x^{+2} \ominus +1}{x \oplus +3}\Bigg|_{x \leftarrow h}$$

ii. We *replace* every occurence of x in the generic expression by h to get the individual expression for h:

$$h \xrightarrow{\quad\mathcal{ZENA}\quad} \mathcal{ZENA}(h) = \frac{h^{+2} \ominus +1}{h \oplus +3}$$

iii. We *execute* the individual expression for h:

$$= \frac{-1 \oplus h^2}{+3 \oplus +h}$$

$$= -\tfrac{1}{3} \oplus +\tfrac{1}{3^{+2}}h \oplus +\tfrac{8}{3^{+3}}h^{+2}$$

The last expression above *is* the executed expression.

iv. We *format* according to DEFINITION 4.3 - Local formats (Page 216)

$$\Big(h, -\tfrac{1}{3} \oplus +\tfrac{1}{3^{+2}}h \oplus +\tfrac{8}{3^{+3}}h^{+2}\Big)$$

$$\mathcal{ZENA}\Big(h\Big) = -\tfrac{1}{3} \oplus +\tfrac{1}{3^{+2}}h \oplus +\tfrac{8}{3^{+3}}h^{+2}$$

$$h \xrightarrow{\mathcal{ZENA}} = -\tfrac{1}{3} \oplus +\tfrac{1}{3^{+2}}h \oplus +\tfrac{8}{3^{+3}}h^{+2}$$

$$h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\Big(h\Big) = -\tfrac{1}{3} \oplus +\tfrac{1}{3^{+2}}h \oplus +\tfrac{8}{3^{+3}}h^{+2}$$

3. Near x_0 .

PROCEDURE 4.4 To get the outputs returned *near* x_0 by a function f given by an I-O rule $x \xrightarrow{f} f(x) =$ generic expression in terms of x,

i. Declare that the input is a nearby indeterminate number by using the local variable $x_0 \oplus h$ and writing the declaration $\Big|_{x \leftarrow x_0 \oplus h}$ to the right of the global input-output rule:

$$x \xrightarrow{f} f(x) = \text{generic expression in terms of } x \Big|_{x \leftarrow x_0 \oplus h}$$

ii. Replace every occurence of x in the global input-output rule by the local variable $x_0 \oplus h$ to get the **local input-output rule** *near* x_0:

$$x_0 \oplus h \xrightarrow{f} f\left(x_0 \oplus h\right) = \text{generic expression in terms of } x_0 \oplus h$$

iii. Execute the generic expression in terms of the relevant variable according to the rules in Section 9 - Computing with Qualitative Sizes (Page 33), that is do the operations in the generic expression to get the **executed expression**

iv. Format according to DEFINITION 4.3 - Local formats (Page 216)

- For *graphing*, use the input-output pair

$$\left(x_0 \oplus h, \text{executed expression in terms of } h\right)$$

- For *computing*, use the equality

$$\underbrace{f\left(x_0 \oplus h\right) = \text{executed expression in terms of } h}_{\text{Local input-output rule near } 0}$$

- For *seeing*, use the arrow pair

$$x_0 \oplus h \xrightarrow{f} \text{executed expression in terms of } h$$

- For *thinking*, use the **local input-output rule**

$$x_0 \oplus h \xrightarrow{f} \underbrace{f\left(h\right) = \text{executed expression in terms of } h}_{\text{Local input-output rule near } 0}$$

Note that the executed expression in the output for inputs near x_0, that is in the output for $x_0 \oplus h$, is *not* the same as the executed expression in the output for inputs near 0, that is in the output for h, because the executed expression in the output for $x_0 \oplus h$ "contains" x_0

DEMO 4.4a To get the outputs returned for inputs *near* $+5$ by the function $\mathcal{ZENA}$ given by the global input-output rule $x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \dfrac{x^{+2} \ominus +1}{x \oplus +3}$

i. We *declare* that the input is an indeterminate number *near* $+5$ by writing the declaration $\Big|_{x \leftarrow +5 \oplus h}$ to the right of the global input-output rule:

$$x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \frac{x^{+2} \ominus +1}{x \oplus +3}\Bigg|_{x \leftarrow +5 \oplus h}$$

ii. We *replace* every occurence of x in the generic expression by $+5 \oplus h$ to get the individual expression for $+5 \oplus h$:

$$+5 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(+5 \oplus h) = \frac{{+5 \oplus h}^{+2} \ominus +1}{+5 \oplus h \oplus +3}$$

iii. We *execute* the individual expression for $+5 \oplus h$:

$$= \frac{+25 \oplus +10h \oplus +h^2 \ominus +1}{+5 \oplus +h \oplus +3}$$

$$= \frac{+24 \oplus +10h \oplus +h^2}{+8 \oplus +h}$$

$$= +3 \oplus +\tfrac{7}{8}h \oplus +\tfrac{1}{64}h^{+2} \oplus [...]$$

The last expression above *is* the executed expression.

Ok, so, why stop the dvision here? You will see in Section 6 - Graphing Power Functions (Page 287)

iv. We *format* the input-output pair:

- $\left(+5 \oplus h, +3 \oplus +\tfrac{7}{8}h \oplus +\tfrac{1}{64}h^{+2} \oplus [...]\right)$
- $\mathcal{ZENA}\left(+5 \oplus h\right) = +3 \oplus +\tfrac{7}{8}h \oplus +\tfrac{1}{64}h^{+2} \oplus [...]$
- $+5 \oplus h \xrightarrow{\mathcal{ZENA}} +3 \oplus +\tfrac{7}{8}h \oplus +\tfrac{1}{64}h^{+2} \oplus [...]$
- $+5 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\left(+5 \oplus h\right) = +3 \oplus +\tfrac{7}{8}h \oplus +\tfrac{1}{64}h^{+2} \oplus [...]$

DEMO 4.4b To get the outputs returned for inputs *near* -3 by the function $\mathcal{ZENA}$ given by the global input-output rule $x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \dfrac{x^{+2} \ominus +1}{x \oplus +3}$

i. We declare that the input is an *undeterminate number near* -3 by writing the declaration $\Big|_{x \leftarrow -3 \oplus h}$ to the right of the global input-output rule

$$x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \frac{x^{+2} \ominus +1}{x \oplus +3}\Bigg|_{x \leftarrow -3 \oplus h}$$

ii. We replace every occurence of x in the input-ouput rule by $-3 \oplus h$ to get the generic expression in terms of numbers *near* -3

$$-3 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\left(-3 \oplus h\right) = \frac{-3 \oplus h^{+2} \ominus +1}{-3 \oplus h \oplus +3}$$

iii. We execute the generic expression in terms of $-3 \oplus h$, that is we do the operations in the individual expression:

$$= \frac{+9 \oplus -6h \oplus h^2 \ominus +1}{-3 \oplus +3 \oplus h}$$

$$= \frac{+8 \oplus -6h \oplus h^2}{h}$$

$$= +8h^{-1} \oplus -6 \oplus h$$

$$= +8h^{-1} \oplus [...]$$

The last expression above *is* the executed expression.

iv. We format the input-output pair, that is we write:

- $\left(-3 \oplus h, +8h^{-1} \oplus -6 \oplus h\right)$
- $\mathcal{ZENA}(-3 \oplus h) = +8h^{-1} \oplus -6 \oplus h$
- $-3 \oplus h \xrightarrow{\mathcal{ZENA}} +8h^{-1} \oplus -6 \oplus h$
- $-3 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\left(-3 \oplus h\right) = +8h^{-1} \oplus -6 \oplus h$

DEMO 4.4c To get the outputs returned for inputs *near* $+1$ by the function $\mathcal{ZENA}$ given by the global input-output rule $x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \dfrac{x^{+2} \ominus +1}{x \oplus -3}$

i. We declare that the outputs are to be for *numbers near* $+3$ by writing the declaration $\Big|_{x \leftarrow +1 \oplus h}$ to the right of the global input-output rule:

$$x \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}(x) = \frac{x^{+2} \ominus +9}{x \oplus -3}\Bigg|_{x \leftarrow +1 \oplus h}$$

ii. We replace every occurence of x in the input-ouput rule by $+1 \oplus h$ to get the generic expression in terms of numbers *near* $+1$

$$+1 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\Big(+1 \oplus h\Big) = \frac{{+1 \oplus h}^{+2} \ominus +1}{+1 \oplus h \oplus -3}$$

iii. We execute the generic expression in terms of $+1 \oplus h$, that is we do the operations in the generic expression:

$$= \frac{+1 \oplus +2h \oplus h^2 \ominus +1}{+3 \oplus -3 \oplus h}$$

$$= \frac{+2h \oplus h^2}{h}$$

$$= +2 \oplus h$$

The last expression above *is* the executed expression.

iv. We format the input-output pair, that is we write:

- $\Big(+1 \oplus h, +2 \oplus h\Big)$
- $\mathcal{ZENA}(+1 \oplus h) = +2 \oplus h$
- $+1 \oplus h \xrightarrow{\mathcal{ZENA}} +2 \oplus h$
- $+1 \oplus h \xrightarrow{\mathcal{ZENA}} \mathcal{ZENA}\Big(+1 \oplus h\Big) = +2 \oplus h$

control point

OKsoFAR **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR**
OKsoFAR **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR**

5 Towards Global Graphs.

There is no general way to deal with functions given by I-O rules and how we will deal with functions given by I-O rules will depend entirely on the kind of expression in terms of x that appears in the I-O rule. In particular, there is no general procedure for getting the global graph of functions given by I-O rules. So here we will only be able to say some general things.

1. Direct problems

2. Reverse problems. When a function f is given by an input-output rule

$$x \xrightarrow{f} f(x) = \text{generic expression in terms of } x$$

the reverse problem for a given y_0

$$f(x) = y_0$$

means to solve the *equation*

$$\text{generic expression in terms of } x = y_0$$

However, since the necessary Agebra depends entirely on the kind of generic expression in terms of x that the input-output rule involves, and therefore on what type of function f is, we will only be able to deal with reverse problems as we go along and study each type of functions.

3. Global graph. Altogether, ∞ and poles will be the inputs that we will call the **control points** for that function.

Chapter 2 - Local Features Functions May Have (Page 133) showed how we need local graphs to *see* local function behaviors, but with functions given by an input-output rule we will have to use Procedure 4.2 - Get output *near* ∞ from f given by an I-O rule (Page 216) and then graph the local input-output rule.

$$\underbrace{x}_{\text{Input}} \xrightarrow{\ f\ } \underbrace{\underbrace{f(x)}_{\text{Output}} = \text{generic expression in terms of } x}_{\text{Global input-output rule}}$$

and so, a function being given by an I-O rule, we will proceed in the following three steps:

a. Locate the points near which we will need a local graph, namely:

- The control points, that is
- There will also the poles, if any, that is the input numbers for which the output is ∞
-

As we saw, there will always be ∞ because it is one of the control points, ∞ and at the very least the poles if any, of the given function.

b. We will have to find the local frames in which the local graphs will be.

c. We will have to find the shape of the local graph.

The reason that there is no simple PROCEDURE for getting local graphs is that:

Step **a** is a reverse problem which will require solving equations that will depend on the generic expression in the I-O rule that gives the function under investigation.

Step **b** of course has already been dealt with with **??** however CAUTIONARY NOTE 1.4 will complicate matters.

Step **c** will depend on being able to approximate the given function.

4. Need for Power Functions. ,

So we will need local graphs for two purposes:

i. Get the global graph
ii. Get the local behaviors

So our approach will be:

i. Get the local graphs we will need to get the essential global graph
ii. Get the local graphs we need to get the needed behaviors

because no number of input-output pairs can almost never get us even an idea of the graph.

OKsoFarOKsoFarOKsoFarOKsoFarOKsoFarOKsoFarOKsoFarOKsoFarOKsoFarOKsoFar

==============OK SO FAR ===============

straight function
straight line
ZERO

Chapter 5

Straight Functions

We will call **straight functions** those functions whose global graph is a **straight line**. Straight functions are therefore exceptional in that they lack local concavity but straight functions are *not* exceptional in that they are used extremely often, if only as benchmarks for functions that are *not* straight.

But if we all know how to draw a straight line is (https://duckduckgo.com/?q=straight+edge&t=ffab&iar=images&iax=images&ia=images), defining a straight line is quite another story, better left to GEOMETRY. *(https://en.wikipedia.org/wiki/Line_(geometry)).*

LANGUAGE NOTE 5.1 The name straight function is absolutely *not* standard but. while there is no standard word, in *this* text, everything *has* to have a name.

1 The Function *ZERO*

This is the absolute simplest possible function:

DEFINITION 5.1 The function ***ZERO*** is given by the I-O rule

$$x \xrightarrow{\mathcal{ZERO}} \mathcal{ZERO}(x) = \underbrace{0}_{\text{General expression in terms of } x}$$

While not a very intersting function by itself, the function $\mathcal{ZERO}$ will actually play a central role among functions very much like the role played by 0 among numbers:

1. Local I-O rule.

2. Local graph.

3. Local features. Since the global graph of $\mathcal{ZERO}$ is the 0-output level line, that is a *straight* line,

i. $\mathcal{ZERO}$ has 0-height everywhere,
ii. $\mathcal{ZERO}$ has 0-slope everywhere,
iii. $\mathcal{ZERO}$ has 0-concavity everywhere.

4. Control point(s).

5. Global graph. The graph points are at the intersection of the input level lines and the output level lines, but since no matter what the input, the output of $\mathcal{ZERO}$ is always 0, the output level line is always the 0-output level line. So, of course:

PROCEDURE 5.1 To get the global graphs of the $\mathcal{ZEROI}$ function, that is of the function given by the I-O rule

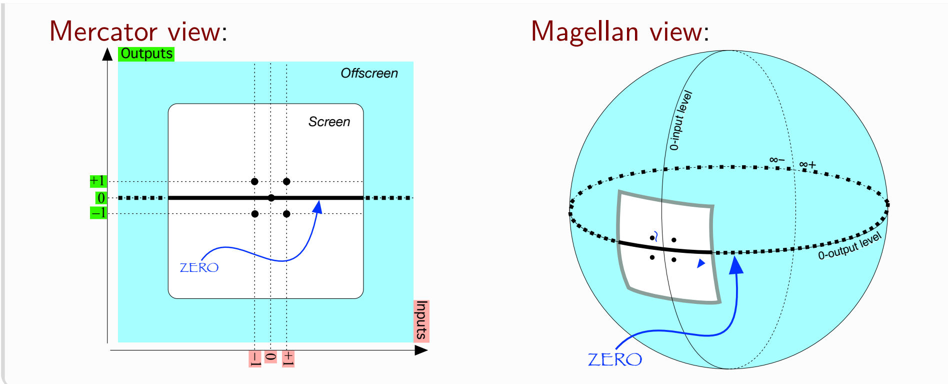
$$x \xrightarrow{\mathcal{ZERO}} \mathcal{ZERO}(x) = 0$$

i. Tickmark the constant coefficient 0 on the output ruler
ii. Draw the output level line through the tickmark

DEMO 5.1
To get the global graph of the function $\mathcal{ZERO}$, that is the function given by the I-O rule

$$x \xrightarrow{\mathcal{ZERO}} \mathcal{ZERO}(x) = 0$$

i. Tickmark 0 on the output ruler
ii. Draw the output level line through the tickmark



constant coefficient
$\mathcal{UNIT}^-$
$\mathcal{UNIT}^+$

2 The Functions $\mathcal{UNIT}^+$ and $\mathcal{UNIT}^-$

These are the next simplest possible functions. While still not very interesting functions by themselves, $\mathcal{UNIT}^+$ and $\mathcal{UNIT}^-$ also play an important role among functions much like the role played by $+1$ and -1 among numbers.

DEFINITION 5.2

- The function $\mathcal{UNIT}^+$ is given by the I-O rule

$$x \xrightarrow{\mathcal{UNIT}^+} \mathcal{UNIT}^+(x) = \underbrace{+1}_{\text{General expression in terms of } x}$$

- The function $\mathcal{UNIT}^-$ is given by the I-O rule

$$x \xrightarrow{\mathcal{UNIT}^-} \mathcal{UNIT}^-(x) = \underbrace{-1}_{\text{General expression in terms of } x}$$

CAUTIONARY NOTE 5.1 The symbols $+$ and $-$ to the upper right of $\mathcal{UNIT}$ are *not* exponents and serve *only* to distinguish the two $\mathcal{UNIT}$ functions.

1. Local I-O rules.

2. Local graphs.

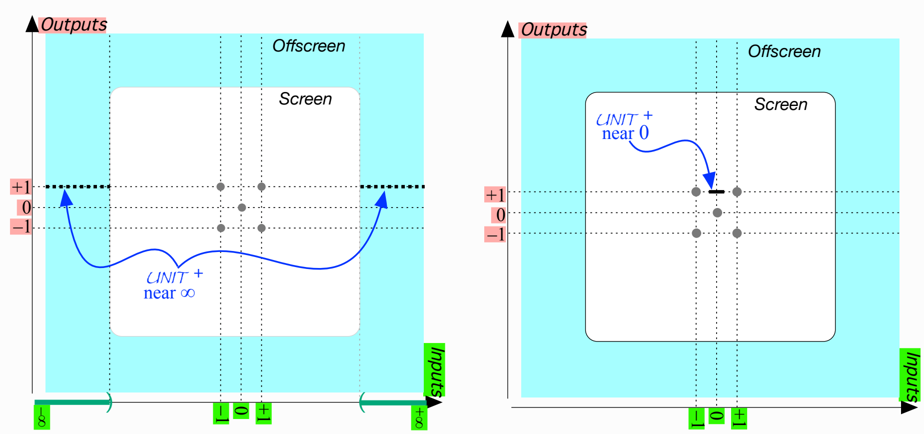
PROCEDURE 5.2 To get the local graph near a point, ∞, 0, or x_0 of a $\mathcal{UNIT}$ function, that is of a function given by the I-O rule .

$$x \xrightarrow{\mathcal{UNIT}} \mathcal{UNIT}(x) = \pm 1$$

Use ?? ?? - ?? (??) to get the local graphs from the global graph given by THEOREM 6.5 - Odd regular power functions are diagonally symmetrical (Page 286) .

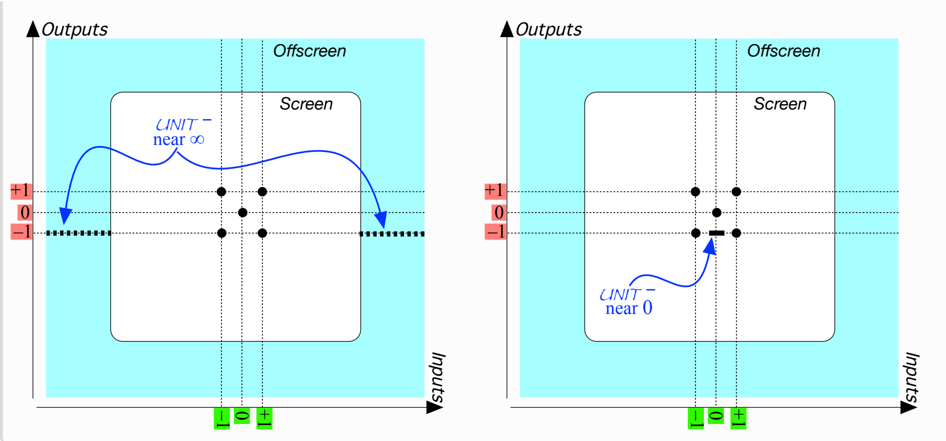
DEMO 5.2a

From DEMO 5.4a - Global graph of the $\mathcal{IDENTITY}$ function (Page 236) we get:



DEMO 5.2b

From ?? ?? - ?? (??)



3. Local features.

4. Control point(s).

5. Global graph. The graph points are at the intersection of the input level lines and the output level lines, but since no matter what the input, the output of $\mathcal{UNIT}^+$ is always $+1$, the output level line is always the $+1$-output level line. So, the global graph of $\mathcal{UNIT}^+$ is

Since, no matter what the input is, the output of a unit function is either always $+1$ or always -1,

PROCEDURE 5.3 To get the global graphs of a $\mathcal{UNIT}$ function, that is of a function given by either one of the I-O rules

$$x \xrightarrow{\mathcal{UNIT}^+} \mathcal{UNIT}^+(x) = +1$$

or

$$x \xrightarrow{\mathcal{UNIT}^-} \mathcal{UNIT}^-(x) = -1$$

i. Mark the constant coefficient $+1$ or -1 on the output ruler
ii. Draw the output level line through the tickmark

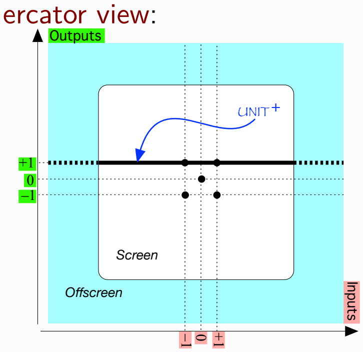
DEMO 5.3a
To get the global graph of the function $\mathcal{UNIT}^+$, that is the function

given by the I-O rule

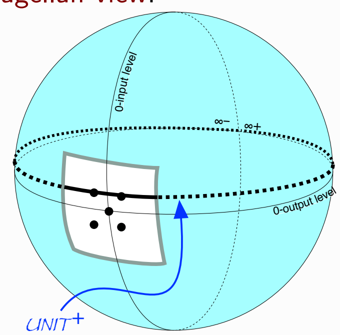
$$x \xrightarrow{\mathcal{UNIT}^+} \mathcal{UNIT}^+(x) = -1$$

i. Mark $+1$ on the output ruler
ii. Draw the output level line through the tickmark

Mercator view:



Magellan view:

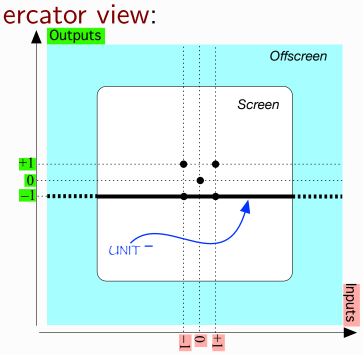


Demo 5.3b
To get the global graph of the function $\mathcal{UNIT}^-$, that is the function given by the I-O rule

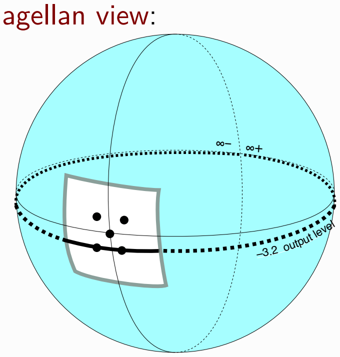
$$x \xrightarrow{\mathcal{UNIT}^-} \mathcal{UNIT}^-(x) = -1$$

i. Mark -1 on the output ruler
ii. Draw the output level line through the tickmark

Mercator view:



Magellan view:



OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR tant function
OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR OKsoFAR bal␣height

3 Constant functions

Constant functions are given by dilating a $\mathcal{UNIT}$ function.

> **LANGUAGE NOTE 5.2** The name *constant functions* is an **abuse of language** because it is not the *function* which is constant but its *output* in the sense that the *output* remains *constantly* equal to the constant coefficient no matter what the input is.

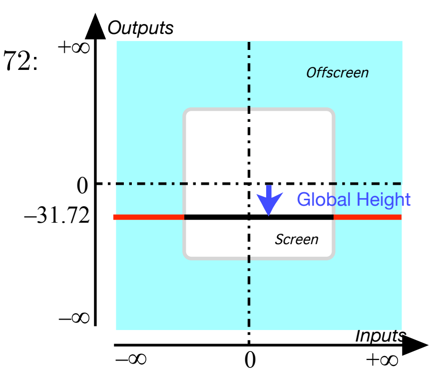
What makes constant functions *exceptional* is that they lack both *local slope* and *local concavity* and have only *local height*.

But then, since for a constant function the *local height* is the same everywhere, we can talk of the **global height** of a *constant function*.

EXAMPLE 5.1. Let f be the function specified by the *global input-output rule*

$$\begin{aligned} x \xrightarrow{\ f\ } f(x) &= (-31.72)x^0 \\ &= -31.72 \end{aligned}$$

the global height of f is -31.72:



Given a *base* function as a monomial function, when we *add-on* a monomial function with the *same* exponent, the *sum* is a monomial function with the same exponent.

Example 5.2. G iven the base function $MINT$ specified by the *global input-output rule*

$$x \xrightarrow{MINT} MINT(x) = -12.82x^{+4}$$

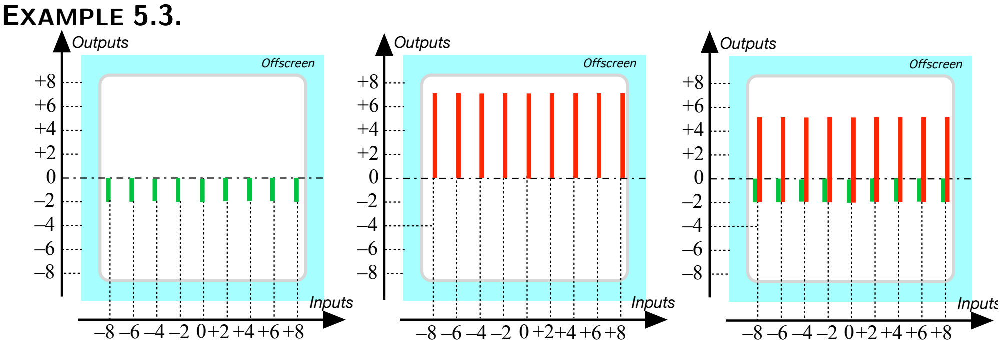
and given the add-on function TEA specified by the *global input-output rule*

$$x \xrightarrow{TEA} TEA(x) = +49.28x^{+4}$$

then the sum function will be specified by the *global input-output rule*

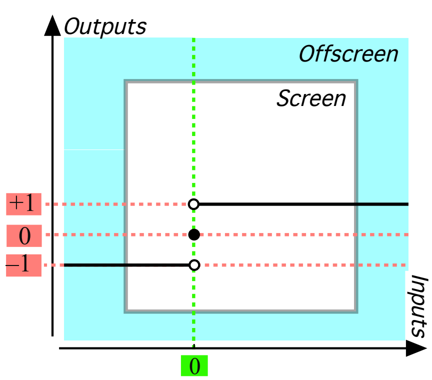
$$\begin{aligned} x \xrightarrow{\quad SUM \quad} SUM(x) &= MINT(x) + TEA(x) \\ &= -12.82x^{+4} \oplus +49.28x^{+4} \\ &= [-12.82 \oplus +49.28]\, x^{+4} \\ &= +36.46x^{+4} \end{aligned}$$

Example 5.3.



4 Piecewise constant functionss

Example 5.4. The function $\mathcal{SIGN}$, aka $\mathcal{HEAVISIDE}$ function, is given by the global graph:

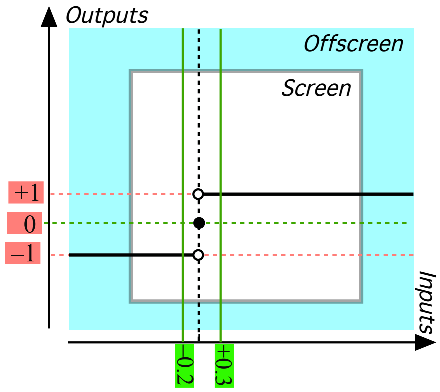


In other words,

- If $x > 0$, then $\mathcal{SIGN}(x) = +1$,
- If $x = 0$, then $\mathcal{SIGN}(x) = 0$,
- If $x < 0$, then $\mathcal{SIGN}(x) = -1$,

diagonal function
linear coefficient

But then, while $SIGN\left(0\right) = 0$



shows that

- $SIGN\left(+0.3\right) = +1$
- $SIGN\left(-0.2\right) = -1$

5 The Diagonal Functions *IDENTITY* and *OPPOSITE*

These functions are the next simplest possible functions given by I-O rule.

1. Global I-O rules.

DEFINITION 5.3

- The **diagonal function** *IDENTITY* is the function given by the I-O rule

$$x \xrightarrow{IDENTITY} IDENTITY(x) = \underbrace{+1 \odot x}_{\text{General expression in terms of } x}$$

- The **diagonal function** *OPPOSITE* is the function given by the I-O rule

$$x \xrightarrow{OPPOSITE} OPPOSITE(x) = \underbrace{-1 \odot x}_{\text{General expression in terms of } x}$$

LANGUAGE NOTE 5.3 The name *diagonal function* is *not* standard but there is no standard word that covers both the *IDENTITY*

function and the $\mathcal{OPPOSITE}$ function in spite of the fact that they really belong together.
The name *diagonal function* alludes to the look of the global graphs relative to the quincunx.

2. Global graphs. The global graph of diagonal functions is the next simplest of the global graphs of the only three kinds of functions whose global graph is a straight line and which, therefore, we can get *directly*.

The global graph of diagonal functions involve plot dots for input-output pairs whose input and output have the same size.

EXAMPLE 5.5. The following I-O pairs give plot dots:

- $\left(0, 0\right)$, for both the Identity function and the Opposite function,
- $\left(+1, +1\right)$, for the Identity function,
- $\left(-73.092, +73.092\right)$, for the Opposite function.

PROCEDURE 5.4 To get the global graph of a function given by the I-O rule

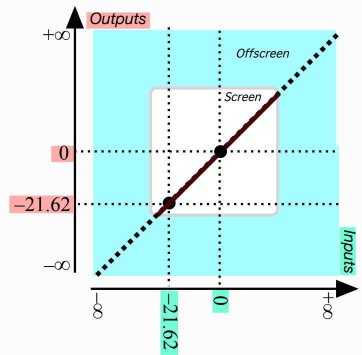
$$x \xrightarrow{\mathcal{DIAGONAL}} \mathcal{DIAGONAL}(x) = \pm 1 \odot x$$

i. Plot any two input-output pairs $\left(\textit{input}, \textit{output}\right)$ with:
For the $\mathcal{IDENTITY}$ function, *output* the *same* as *input*
For the $\mathcal{OPPOSITE}$ function, *output* the *opposite* of *input*
ii. Draw a straight line through the two plot dots.

DEMO 5.4a
To get the global graph of the function given by the I-O rule

$$x \xrightarrow{\mathcal{IDENTITY}} \mathcal{IDENTITY}(x) = +1 \odot x$$

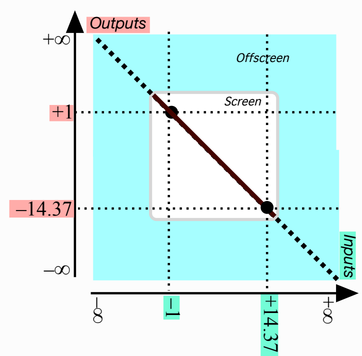
i. Plot, say, the pair $\big(0, 0\big)$ and the pair $\big(-21.62, -21.62\big)$
ii. Draw the straight line through the two plot dots.



DEMO 5.4b
To get the global graph of the function given by the I-O rule

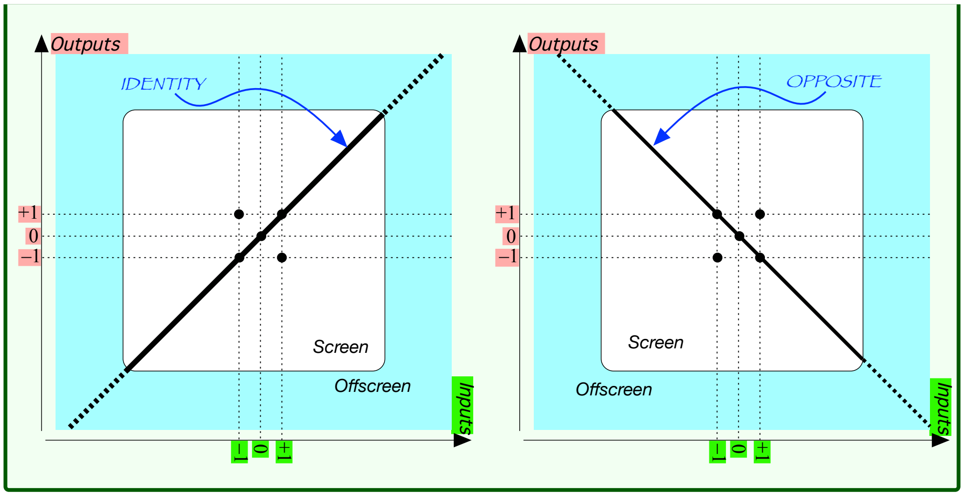
$$x \xrightarrow{\mathcal{OPPOSITE}} \mathcal{OPPOSITE}(x) = -1 \odot x$$

i. Plot, say, the pair $\big(-1, +1\big)$ and the pair $\big(+14.37, -14.37\big)$
ii. Draw the straight line through the two plot dots.



In terms of the quincunx,

THEOREM 5.1 Global graphs of the $\mathcal{DIAGONAL}$ functions. The global graphs of the $\mathcal{DIAGONAL}$ functions are the straight lines that extend the diagonals of the quincunx:



Proof. We cannot *prove* THEOREM 5.1 because we have no *definition* for straight line. □

3. Control point(s).

4. Local graphs.

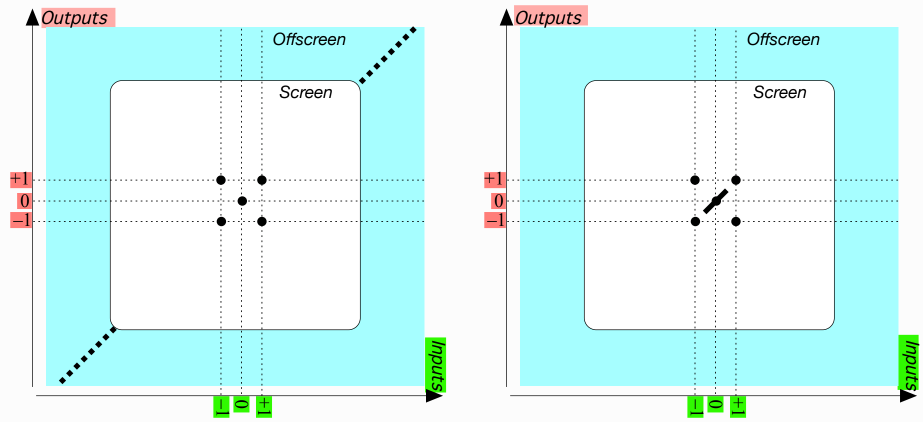
PROCEDURE 5.5 To get the local graph near ∞ or the local graph near 0 of a $\mathcal{DIAGONAL}$ functions given by the I-O rule .

$$x \xrightarrow{\mathcal{DIAGONAL}} \mathcal{DIAGONAL}(x) = \pm 1 \odot x$$

Use ?? ?? - ?? (??) to get the local graphs from the global graph given by THEOREM 5.1 - Global graphs of the $\mathcal{DIAGONAL}$ functions (Page 237).

DEMO 5.5a

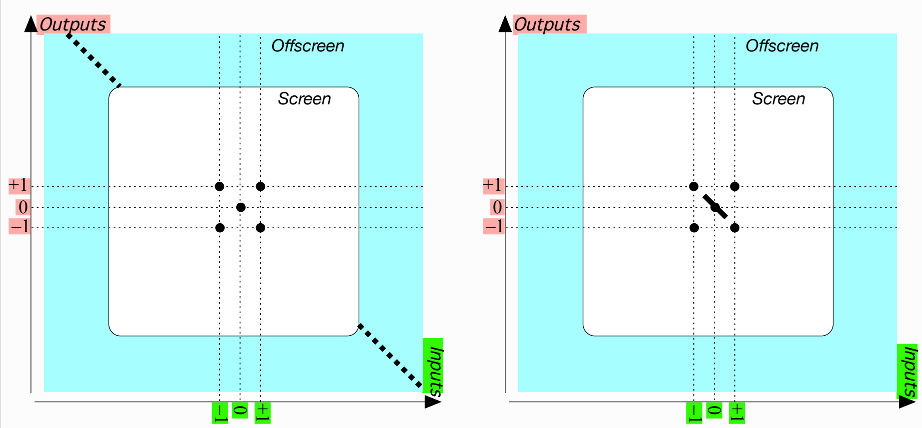
From DEMO 1.9a - Output level band for -7.83 (Page 122) global slope



Local graph near ∞ of *IDENTITY* Local graph near 0 of *IDENTITY*

DEMO 5.5b

From ?? ?? - ?? (??)



Local graph near ∞ of *OPPOSITE* Local graph near 0 of *OPPOSITE*

==========OK SO FAR ===========

5. Local features. What makes diagonal functions different from most other functions is that they lack local concavity and have only local height and local slope.

But then, since for a diagonal function the local slope is the same everywhere, the graph of a diagonal function has a **global slope**,

run
rise

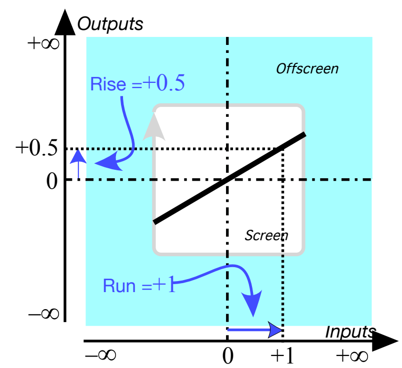
that is the fraction $\frac{\text{Rise}}{\text{Run}}$ where, given two input-output pairs, the **run** is the difference from one input to the other and the **rise** is the corresponding difference from one output to the other.

In fact, the reason we like to use the inputs 0 and 1 is that they make it easy to see that the *global slope* of the *global graph* of a linear function is the *linear coefficient* of the *global input-output rule*.

EXAMPLE 5.6. **L** et f be the function specified by the *global input-output rule*

$$x \xrightarrow{f} f(x) = (+0.5)x^{+1} \\ = +0.5x$$

the global slope of f is $\frac{\text{Rise}}{\text{Run}} = \frac{+0.5}{+1} = +0.5$



6. Local I-O rules.

PROCEDURE 5.6 To get the output *near* a given point for a $\mathcal{DIAGONAL}$ function given by the I-O rule $x \xrightarrow{\mathcal{DIAGONAL}} \mathcal{DIAGONAL}(x) = \pm 1 \odot x$

i. *Declare* that x is to be replaced by L

$$x\Big|_{x \leftarrow L} \xrightarrow{\mathcal{DIAGONAL}} \mathcal{DIAGONAL}(x)\Big|_{x \leftarrow L} = \pm 1 \odot x\Big|_{x \leftarrow L}$$

that is:

$$L \xrightarrow{\mathcal{DIAGONAL}} \mathcal{DIAGONAL}\big(L\big) = \pm 1 \odot L$$

Linear function
dilation function

ii. *Execute* the output-specifying code into a *jet* near ∞

$$= \underbrace{[a]L}_{\text{jet near } \infty}$$

which gives the *local input-output rule* near ∞:

$$L \xrightarrow{DIAGONAL} LINEAR(x) = \underbrace{[a]x}_{\text{jet near } \infty}$$

7. Local features. Output *near* a point.

6 Linear Functions

Linear functions are given by dilating a $\mathcal{DIAGONAL}$ function.

Language Note 5.4 Another name for *linear function* is **dilation function** because it is easy to prove that the distance between any two outputs is obtained by just "dilating" the distance between the two inputs by the coefficient. (See https://en.wikipedia.org/wiki/Dilation_(metric_space).)

Cautionary Note 5.2 dgfdgfddgf (https://en.wikipedia.org/wiki/Linear_function)

Demo 5.6a To get the output near ∞ of the function specified by the global input-output rule

$$x \xrightarrow{BINK} BINK(x) = \underbrace{(-26.18)x^{+1}}_{\text{output-specifying code}}$$

i. We *declare* that x is to be replaced by $\pm L$

$$x\Big|_{x\leftarrow \pm L} \xrightarrow{BINK} BINK(x)\Big|_{x\leftarrow \pm L} = (-26.18)x^{+1}\Big|_{x\leftarrow \pm L}$$

which gives:

$$\pm L \xrightarrow{BINK} BINK(-3) = \underbrace{(-26.18)\cdot(\pm L)^{+1}}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code

$$= -26.18 \cdot \pm L$$

Since 26.18 is *medium-size*, **??** on **??** gives $26.18 \cdot large = large$ and, using **??** on **??**, we get:

$$= \mp large$$

which gives the *input-output pair*

$$(\pm large, \mp large)$$

DEMO 5.6b To get the output near 0 of the function specified by the global input-output rule

$$x \xrightarrow{JINK} JINK(x) = \underbrace{(+45.57)x}_{\text{output-specifying code}}$$

i. We *declare* that x is to be replaced by $\pm small$

$$x \Big|_{x \leftarrow \pm small} \xrightarrow{JINK} JINK(x) \Big|_{x \leftarrow \pm small} = (+45.57)x^{+1} \Big|_{x \leftarrow \pm small}$$

which gives:

$$\pm small \xrightarrow{JINK} JINK(\pm small) = \underbrace{(+45.57) \cdot (\pm small)^{+1}}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code

$$= +45.57 \cdot \pm small$$

Since 45.57 is *medium-size*, **??** on **??** gives $45.57 \cdot small = small$ and, using **??** on **??**, we get:

$$= \pm small$$

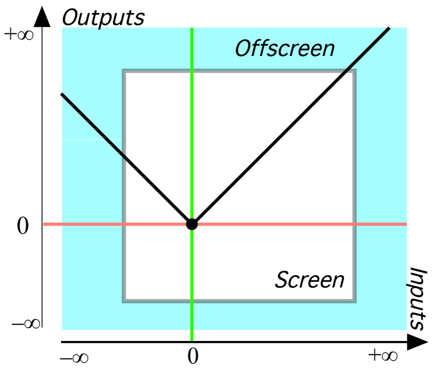
which gives the *input-output pair*

$$(\pm small, \pm small)$$

BeginWORKzone - BeginWORKzone - BeginWORKzone - BeginWORKzone - BeginWORKz

7 Piecewise Linear Functions

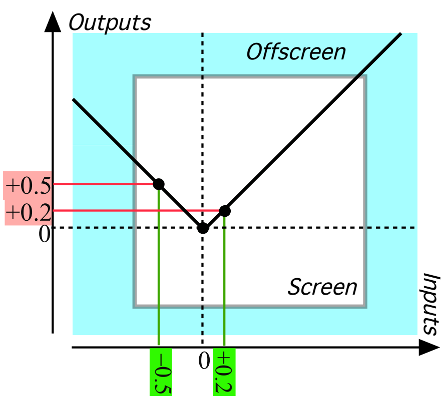
Example 5.7. The function $\mathcal{SIZE}$, aka $\mathcal{ABSOLUTE\,VALUE}$ function, is given by the global graph:



In other words,

- If $x > 0$, then $\mathcal{SIZE}(x) = x$,
- If $x = 0$, then $\mathcal{SIZE}(x) = 0$,
- If $x < 0$, then $\mathcal{SIZE}(x) =$ Opposite x,

But then, while $\mathcal{SIZE}\left(0\right) = 0$



shows that

- $\mathcal{SIZE}\left(+0.2\right) > 0$
- $\mathcal{SIZE}\left(-0.5\right) > 0$

8 Affine Functions

addong

Definition 5.4 An **affine functions** is the sum of a linear function and a constant functon, that is, in other words, a linear combination of $\mathcal{UNIT}^+$ and $\mathcal{IDENTITY}$:

linear part
constant part
affine function
linear term
linear coefficient
constant term
constant coefficient

$$x \xrightarrow{\mathcal{AFFINE}} \mathcal{AFFINE}(x) = a \odot \mathcal{IDENTITY}(x) \oplus b \odot \mathcal{UNIT}^{+}(x)$$
$$= a \odot \mathcal{IDENTITY}(x) \oplus b \odot \mathcal{UNIT}^{+}(x)$$
$$= a \odot x \oplus b \odot +1$$
$$= \underbrace{a \odot x \oplus b}_{\text{Generic expression in terms of } x}$$

in which
$a \odot x$ is the **linear term** and a is the **linear coefficient**
and
b is both the **constant term** and the **constant coefficient**
Moreover,

- The **linear part** of $AFFINE$ is the linear function given by:
$$x \xrightarrow{\mathcal{LIN-AFFINE}} \mathcal{LIN}-\mathcal{AFFINE}(x) = a \odot x$$

and,

- The **constant part** of $AFFINE$ is the constant function given by:
$$x \xrightarrow{\mathcal{CONST-AFFINE}} \mathcal{CONST}-\mathcal{AFFINE}(x) = b$$

Example 5.8. The affine function $\mathcal{NINA}$ given by the linear coefficient -31.39 and the constant coefficient $+5.34$ is the function given by the I-O rule

$$x \xrightarrow{\mathcal{NINA}} \mathcal{NINA}(x) = \underbrace{-31.39 \odot x}_{\text{linear term}} \oplus \underbrace{+5.34}_{\text{constant trem}}$$
$$= \underbrace{-31.39}_{\text{linear coefficient}} x \oplus \underbrace{+5.34}_{\text{constant coefficient}}$$

Moreover,

- The *linear part* of $NINA$ is the linear function
$$x \xrightarrow{\mathcal{LIN-NINA}} \mathcal{LIN}-\mathcal{NINA}(x) = -31.39 \odot x$$

and

- The *constant part* of $NINA$ is the constant function
$$x \xrightarrow{\mathcal{CONST-NINA}} \mathcal{CONSR}-\mathcal{NINA}(x) = +5.34$$

Cautionary note 5.3 Very unfortunately, there is a lot of confusion in the literature between linear functions and affine functions (Including in https://en.wikipedia.org/wiki/Linear_function_(calculus) but see https://en.wikipedia.org/wiki/Linear_function.) This is because, as we will see, like linear functions, affine functions also have a straight line as global graph.
The reason this confusion is most regrettable is that linear functions have a very important property, 'linearity', that no other functions—including affine functions—have: A dilation of a sum of two functions is the sum of the dilations of the two functions. (https://en.wikipedia.org/wiki/Linearity) We will discuss 'linearity' in xxx:
See **??** **??** - **??** (**??**)

OKsoFAR **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR** **OKsoFAR**

BeginWORKzone - BeginWORKzone - BeginWORKzone - BeginWORKzone - BeginWORKzone

1. Output *at* a given number We use Procedure 4.1 - Get the output *at* x_0 from the I-O rule giving f (Page 208)

Example 5.9.

Example 5.10.

Procedure 5.7
i. *Declare* that x is to be replaced by x_0

$$x\Big|_{x\leftarrow x_0} \xrightarrow{AFFINE} AFFINE(x)\Big|_{x\leftarrow x_0} = ax+b\Big|_{x\leftarrow x_0}$$

which gives:

$$x_0 \xrightarrow{AFFINE} AFFINE(x_0) = \underbrace{ax_0 + b}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into an output *number*:

$$= ax_0 + b$$

which gives the input-output pair

$$(x_0, ax_0 + b)$$

Demo 5.7

i. We *declare* that x is to be replaced by -3

$$x\Big|_{x\leftarrow -3} \xrightarrow{ALDA} ALDA(x)\Big|_{x\leftarrow -3} = -32.67x + 71.07\Big|_{x\leftarrow -3}$$

which gives

$$-3 \xrightarrow{ALDA} ALDA(-3) = \underbrace{-32.67(-3) + 71.07}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into an output *number*:

$$= +98.01 + 71.07$$

$$= +169.08$$

which gives the *input-output pair*

$$(-3, +169.08)$$

However, as already discussed in **?? ??** and as has already been the case with *monomial* functions, instead of getting the output of an affine function *at* a given input, be it ∞ or x_0, we will usually get the output of the affine function *near* that given input.

2. Output *near* ∞ In order to get the output *near* ∞, we could proceed as we did in **?? ??** with monomial functions, that is we could *declare* "x is $\pm large$" and replace x everywhere in the output-specifying code by $\pm large$. However, the generic expression for affine functions and all functions thereafter will involve more than just one term and using $\pm large$ would become more and more time consuming.

So, in conformity with universal practice, we will declare "x near ∞" but write just x after that. This, though, is extremely dangerous as it is easy to forget that what we write may be TRUE *only* because x has been declared to be near ∞.

3. We will *execute* the output-specifying code, here $ax+b$, into a **jet** , that is with the terms in *descending order of sizes*, which, because x is *large*, means that the powers of x must be in *descending order of exponents*. We will then have the **local input-output rule near** ∞

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = \underbrace{\text{Powers of } x \text{ in } \textit{descending order of exponents}}_{\text{output jet near } \infty}$$

EXAMPLE 5.11. Given the function given by

$$x \xrightarrow{BIBA} BIBA(x) = -61.03 - 82.47x$$

To get the jet near ∞, we first need to get the *order of sizes*.

i. -61.03 is *bounded*

ii. -82.47 is *bounded* and x is *large*. So, since $bounded \cdot large = large$, $-82.47 \cdot x$ is *large*

Then, in the jet near ∞, $-82.47x$ must come first and -61.03 comes second

So, we get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{BIBA} BIBA(x) = \underbrace{-82.47x - 61.03}_{\text{output jet near } \infty}$$

4. Altogether, then:

PROCEDURE 5.8

i. *Declare* that x is near ∞

$$x\Big|_{x \text{ near } \infty} \xrightarrow{AFFINE} AFFINE(x)\Big|_{x \text{ near } \infty} = ax+b\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = \underbrace{ax+b}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into a *jet* near ∞

$$= \underbrace{[a]\, x \oplus [b]}_{\text{output jet near } \infty}$$

where

- a is the *linear* coefficient in the jet near ∞
- b is the *constant* coefficient in the jet near ∞.

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = \underbrace{[a]\, x \oplus [b]}_{\text{output jet near } \infty}$$

approximate

(*Here* the jet near ∞ looks the same as the given global input-output rule but that is only because the output-specifying code *happened* to be written in *descending* order of exponents.)

DEMO 5.8

i. We *declare* that x is near ∞

$$x\Big|_{x \text{ near } \infty} \xrightarrow{NINA} NINA(x)\Big|_{x \text{ near } \infty} = -61.03 - 82.47x\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{NINA} NINA(x) = \underbrace{-61.03 - 82.47\,x}_{\text{output-specifying code}}$$

ii. We *execute* the output-specifying code into a *jet* near ∞:

$$= \big[-82.47\big]\,x \oplus \big[-61.03\big]$$

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{NINA} NINA(x) = \underbrace{\big[-82.47\big]\,x \oplus \big[-61.03\big]}_{\text{output jet near } \infty}$$

where:

- -82.47 is the *linear* coefficient in the jet near ∞
- -61.03 is the *constant* coefficient in the jet near ∞.

(Here the jet near ∞ does *not* look the same as the *global* input-output rule because the output-specifying code happened *not* to be in descending order of exponents.)

The reason we use *jets* here is that the term *largest in size* is the *first* term so that to **approximate** the output we need only write the *first* term in the jet and just replace the remaining terms by [...] which stands for "something too small to matter here". In other words,

THEOREM 5.2 Approximate output near ∞. For *affine* functions, the term in the jet that contributes most to the output near ∞ is the *highest degree term* in the output jet near ∞:

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = \big[a\big]x + [...]$$

EXAMPLE 5.12. Given the function given by

$$x \xrightarrow{NINA} NINA(x) = -61.03 - 82.47x$$

$$x \text{ near } \infty \xrightarrow{NINA} NINA(x) = \left[-82.47\right] x + \left[-61.03\right]$$

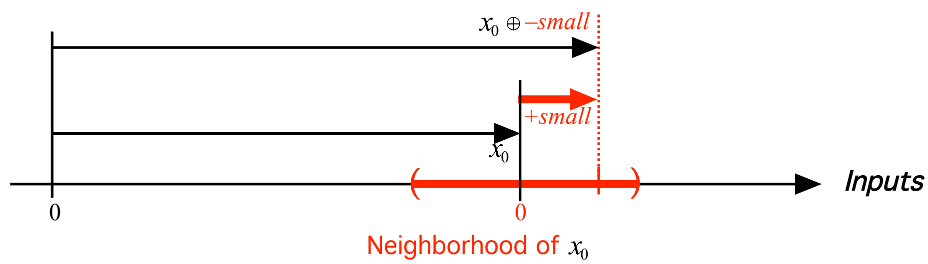
near ∞ we will often just use the *approximation*

$$x \text{ near } \infty \xrightarrow{NINA} NINA(x) = \left[-82.47\right] x + [...]$$

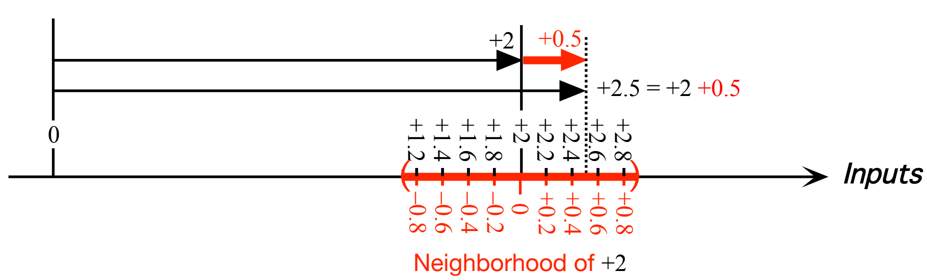
5. Output *near* x_0 https://en.wikipedia.org/wiki/Jet_(mathematics)

While with monomial functions 0 played just as iALERT a role as ∞ (Section 7 Reciprocity), this will not at all be the case with affine functions and all functions thereafter as we *will* very often be interested in the neighborhood of some *given* bounded input(s) *other* than 0. As a matter of fact, the input 0 will usually not be of much more interest than other bounded inputs. (But we will often be concerned with the *output* 0.)

6. In order to "thicken the plot" near a given *bounded* input, we could proceed basically just as we did in **?? ??** with monomial functions, that is *declare* "$x \leftarrow x_0 + small$" or "$x \leftarrow x_0 - small$"and replace x everywhere in the output-specifying code by "$x_0 \oplus +small$"

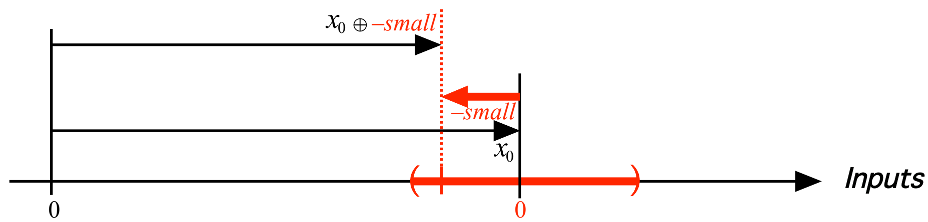


EXAMPLE 5.13. The input $+2.5$ is *near* the given input $+2$:

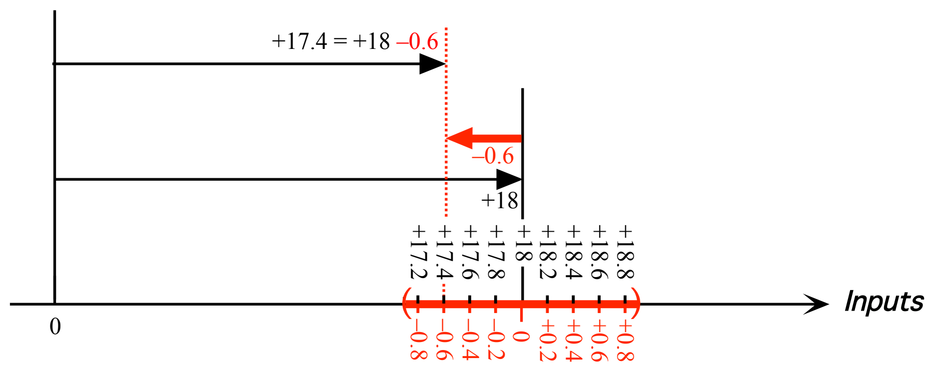


or by "$x \leftarrow x_0 - small$".

h



EXAMPLE 5.14. The input $+17.4$ is *near* the given input $+18$:



However, as already pointed out in subsection 8.2 Output *near* ∞, unlike monomial functions the generic expression for affine functions and all functions thereafter will involves more than just one term. So, using "$x_0 \oplus +small$" or "$x_0 \oplus -small$" would become more and more time consuming and instead we will use "$x_0 + h$" where the letter h is universally accepted as standing for $+small$ or $-small$. In other words, h already includes the *sign*.

Of course, in order to input a neighborhood of 0, we will declare that $x \leftarrow h$, aka $x \leftarrow 0 + h$, in other words that x is to be replaced by h.

7. We can then *execute* the input-output specifying phrase into a *jet* that is with the terms in **descending *order of sizes*** which here, since h is *small*, means that the powers of h will have to be in *ascending* order of exponents. We will then have the local input-output rule near the given input:

$$x_0 \oplus h \xrightarrow{AFFINE} AFFINE(x_0 \oplus h) = \underbrace{\text{Powers of } h \text{ in } \textit{ascending order of exponents}}_{\text{output jet near } \infty}$$

EXAMPLE 5.15. Given the function given by the global input-output rule

$$x \xrightarrow{BIBA} BIBA(x) = -82.47x - 61.03$$

Near $+2$ we do *not* already have the powers of h and we must begin by getting them.

$$\begin{aligned} +2+h \xrightarrow{BIBA} BIBA(+2+h) &= -82.47(+2+h) - 61.03 \\ &= -82.47(+2) - 82.47h - 61.03 \\ &= -164.94 - 82.47h - 61.03 \\ &= -82.47h - 225.97 \end{aligned}$$

Now we need to get the powers of h in *descending order of sizes*: Since -82.47 is $bounded$ and h is $small$ then by **??** on **??**, $-82.47 \cdot h$ is $small$ while -225.97 is $bounded$ so that -225.97 comes first and we get the local input-output rule near $+2$:

$$+2+h \xrightarrow{AFFINE} AFFINE(+2+h) = \underbrace{-225.97 - 82.47h}_{\text{output jet near } +2}$$

8. We will therefore use:

PROCEDURE 5.9

i. *Declare* that x is to be replaced by $x_0 + h$

$$x\Big|_{x \leftarrow x_0+h} \xrightarrow{AFFINE} AFFINE(x)\Big|_{x \leftarrow x_0+h} = ax + b\Big|_{x \leftarrow x_0+h}$$

which gives:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0 + h) = \underbrace{a(x_0 + h) + b}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into a *jet* near x_0:

$$= ax_0 + ah + b$$

iii. *Reorganize* into a **jet near** x_0:

$$= \underbrace{\Big[ax_0 + b\Big] \oplus \Big[a\Big]h}_{\text{output jet near } x_0}$$

which gives the *local input-output rule* near x_0:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0 + h) = \underbrace{\Big[ax_0 + b\Big] \oplus \Big[a\Big]h}_{\text{output jet near } x_0}$$

where:

- $ax_0 + b$ is the **constant coefficient in the jet near** x_0
- a is the **linear coefficient in the jet near** x_0.

DEMO 5.9

i. We declare that x is to be replaced by $-3+h$

$$x\Big|_{x\leftarrow -3+h} \xrightarrow{ALDA} ALDA(x)\Big|_{x\leftarrow -3+h} = -32.67x + 71.07\Big|_{x\leftarrow -3+h}$$

which gives

$$-3+h \xrightarrow{ALDA} ALDA(-3+h) = \underbrace{-32.67(-3+h) + 71.07}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into a *jet* near -3:

$$\begin{aligned} &= -32.67(-3) - 32.67h + 71.07 \\ &= +98.01 - 32.67h + 71.07 \\ &= +98.01 + 71.07 - 32.67h \\ &= \underbrace{\Big[+169.08\Big] \oplus \Big[-32.67\Big]h}_{\text{output jet near } -3} \end{aligned}$$

which gives the *local input-output rule* near -3:

$$-3+h \xrightarrow{ALDA} ALDA(-3+h) = \underbrace{\Big[+169.08\Big] \oplus \Big[-32.67\Big]h}_{\text{output jet near } -3}$$

9. Near x_0, just as we saw was the case near ∞ (see THEOREM 7.10 on page 330), we will often *approximate* the jet to the term(s) that is(are) *largest in size*, which near x_0 is(are) the power(s) of h with the *lowest* exponent(s), and we will just replace the remaining terms by [...] to stand for "something too small to matter here".

In fact, for *affine* functions, we will often use:

THEOREM 5.3 Approximate output near x_0 **.** For *affine* functions, near x_0 the term in the jet that contributes the most to the output is *ordinarily* the *constant term*:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0 + h) = \Big[ax_0 + b\Big] + [...]$$

The exception is of course when the constant term $ax_0 + b = 0$.

10. When all we want is a feature-sign, though, the above procedure is inefficient and we will then use the following procedure based directly on the fact that an *affine function* is the addition of:

- a *cube function*, (See DEFINITION 6.5 on page 264)
- a *square function*, (See DEFINITION 6.2 on page 262)
- a *linear function*, (See **??** on **??**.)
- a *constant function*. (See **??** on **??**.)

that is:

$$x \xrightarrow{AFFINE} AFFINE(x) = \underbrace{ax^3}_{\text{cube}} \oplus \underbrace{bx^2}_{\text{square}} \oplus \underbrace{cx}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

We declare that x is near x_0 that is that x must be replaced by $x_0 + h$:

$$x \xrightarrow{AFFINE} AFFINE(x) = \underbrace{a\,(x_0+h)^3}_{\text{cube}} \oplus \underbrace{b\,(x_0+h)^2}_{\text{square}} \oplus \underbrace{c\,(x_0+h)}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

The output of the local input-output rule near x_0 will have to be a *jet*:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = \left[\quad\right] \oplus \left[\quad\right]h \oplus \left[\quad\right]h^2 \oplus \left[\quad\right]h^3$$

and we want to be able to get any one of the coefficients of the output jet without having to compute any of the other coefficients. So, what we will do is to get the contribution of each monomial function to the term we want.

This requires us to have the *addition formula* at our finger tips:

a.

$$(x_0+h)^2 = x_0^2 + 2x_0h + h^2$$ (See **??** on page 519)

b.

$$(x_0+h)^3 = x_0^3 + 3x_0^2h + 3x_0h^2 + h^3$$ (See **??** on **??**)

More precisely,

i. If we want the *coefficient* of h^0 in the output jet:

- The cube function contributes ax_0^3
- The square function contributes bx_0^2
- The linear function contributes cx_0
- The constant function contributes d

so we have:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = \left[ax_0^3 + bx_0^2 + cx_0 + d\right] \oplus \left[\quad\right]h \oplus \left[\quad\right]h^2 \oplus \left[\quad\right]h^3$$

ii. If we want the *coefficient* of h^1 in the output jet:

- The cube monomial function contributes $3bx_0^2$

- The square monomial function contributes $2bx_0$
- The linear monomial function contributes c
- The constant monomial function contributes nothing

so we have:
$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = [\quad] \oplus [3bx_0^2 + 2bx_0 + c]h \oplus [\quad]h^2 \oplus [\quad]h^3$
If we want the *coefficient* of h^2 in the output jet:

- The cube monomial function contributes $3bx_0$
- The square monomial function contributes c
- The linear monomial function contributes nothing
- The constant monomial function contributes nothing

so we have:
$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = [\quad] \oplus [\quad]h \oplus [c]h^2 \oplus [\quad]h^3$

11. If we want the *coefficient* of h^3 in the output jet:

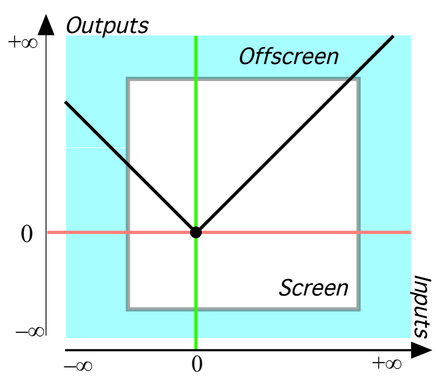
- The square monomial function contributes nothing
- The linear monomial function contributes nothing
- The constant monomial function contributes nothing

so we have:
$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = [\quad] \oplus [\quad]h \oplus [\quad]h^2 \oplus [a]h^3$

EndWORKzone - EndWORKzone - EndWORKzone - EndWORKzone - EndWORKzone

9 Piecewise Affine Functions

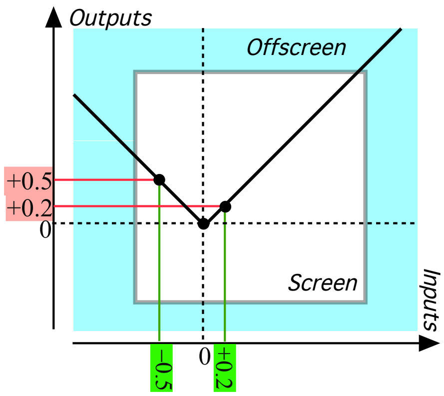
EXAMPLE 5.16. The function $\mathcal{SIZE}$, aka $\mathcal{ABSOLUTE\ VALUE}$ function, is given by the global graph:



In other words,

- If $x > 0$, then $\mathcal{SIZE}(x) = x$,
- If $x = 0$, then $\mathcal{SIZE}(x) = 0$,
- If $x < 0$, then $\mathcal{SIZE}(x) =$ Opposite x,

But then, while $\mathcal{SIZE}\left(0\right) = 0$



shows that

- $\mathcal{SIZE}\left(+0.2\right) > 0$
- $\mathcal{SIZE}\left(-0.5\right) > 0$

power function
coefficient
exponent

Chapter 6

Regular Power Functions

=======Begin WORK ZONE=======

We now come to the functions that will be at the very core of CALCULUS ACCORDING TO THE REAL WORLD inasmuch as, being the simplest possible functions we can give with an I-O rule, they will, as mentioned in Chapter 0, be to (*Laurent*) *polynomial approximations* of functions (https://en.wikipedia.org/wiki/Asymptotic_expansion) what powers of TEN,

About time too!

$$\ldots 0.000\,1,\ 0.001,\ 0.01,\ 0.1,\ 1.0,\ 10.0,\ 100.0,\ 1\,000.0,\ 10\,000.0\ \ldots$$

are to *decimal approximations* of real numbers.

1. I-O rule.

DEFINITION 6.1
A **power function** $\mathcal{P}$ is a function given by an I-O rule of the form

$$x \xrightarrow{\ \mathcal{P}\ } \mathcal{P}(x) = \underbrace{\textbf{coefficient}\ x^{\textbf{exponent}}}_{\textbf{General expression in terms of } x}$$

where:
- the given coefficient is either $+1.$ or $-1.$,
- the given exponent is a signed *counting* number $\pm n$

In other words, the I-O rule of a power function is:

$$x \xrightarrow{f} f(x) = \underbrace{\pm 1\, x^{\pm n}}_{\textbf{General expression in terms of } x}$$

in which

- Size exponent says *how many copies* of x are to be made and multiplied if there is more than one copy. (If exponent is 0, then *no* copy of x is to be made.)
- Sign exponent. $\pm$. says what to do to the coefficient with the product of the copies of x:
  - If Sign exponent is $+$ then the coefficient, namely ± 1, is to be *multiplied* by the product of the copies of x
  - If Sign exponent is $-$, then the coefficient, namely ± 1, is to be *divided* by the product of the copies of x,
  - If exponent is 0, then the coefficient, namely ± 1, is to be *left alone*

Example 6.1. Let f be the power function given by the I-O rule

$$x \xrightarrow{f} f(x) = -1\, x^{+7}$$

then, Definition 6.1 - Power Functions (Page 257) gives

$$= -1 \text{ multiplied by } x \odot x \odot x \odot x \odot x \odot x \odot x$$

$$= -1 \odot x \odot x \odot x \odot x \odot x \odot x \odot x$$

Example 6.2. Let f be the power function given by the I-O rule

$$x \xrightarrow{f} f(x) = -1\, x^{-6}$$

then, Definition 6.1 - Power Functions (Page 257) gives

$$= -1 \text{ dovided by } x \odot x \odot x \odot x \odot x \odot x$$

$$= \frac{-1}{\text{divided by}\; x \odot x \odot x \odot x \odot x \odot x}$$

$$= \frac{-1}{x \odot x \odot x \odot x \odot x \odot x}$$

EXAMPLE 6.3. Let f be the power function given by the I-O rule

$$x \xrightarrow{f} f(x) = -1\, x^{0}$$

Then, DEFINITION 6.1 - Power Functions (Page 257) gives

$$= -1 \text{ left alone}$$

$$= -1$$

> **CAUTIONARY NOTE 6.1**
> - The $+$ in the coefficient goes very often without saying,
> - The $+1$ in the exponent goes *entirely* without saying,
>
> However, even though the above "shortcuts" are completely standard, since nothing goes without saying in *this* text, these shortcuts will *not* be used.

In other words, the idea is for you to go only by what you see.

EXAMPLE 6.4. The I-O rule

$$x \xrightarrow{f} f(x) = -1\, x^{+7}$$

would usually be "shortened" to:

$$x \xrightarrow{f} f(x) = -\, x^{7}$$

But not in this text!

EXAMPLE 6.5. The I-O rule

$$x \xrightarrow{f} f(x) = +1\, x^{+1}$$

would usually be "shortened" to:

$$x \xrightarrow{f} f(x) = x$$

EXAMPLE 6.6. The I-O rule

$$x \xrightarrow{f} f(x) = +1\, x^{-4}$$

would usually be "shortened" to:

$$x \xrightarrow{f} f(x) = x^{-4}$$

But then where would the number the 4 copies of x are to divide come from?

Aside from being used to approximate functions, the power functions will also be:

- The functions whose local graphs near 0 and near ∞ we will use to get the local graphs of all the functions we will discuss in this Volume I. **??** **??** - **??** (??)

exceptional power function

- The simplest functions to exhibit the local behaviors we described in Chapter 2
- The functions which we will use to 'gauge' all other functions.

LANGUAGE NOTE 6.1 **Power Functions** is the name that is normally used when the coefficient is $+1$ or -1 but, unfortunately, the name power function is also used when the coefficient is *any* number, something, however, we will *not* do in this text.

2. Exceptional power functions. We will call those power functions for which exponent is *either* 0 or $+1$ **exceptional power functions** because:

THEOREM 6.1 Exceptional power functions lack at least one feature:

- Exceptional power functions for which exponent is 0 lack slope and concavity,
- Exceptional power functions for which exponent is $+1$ lack concavity,

Proof.
- When exponent is 0 we have $x \xrightarrow{f} f(x) = +1\,x^{0}$

$$= +1$$
$$= +1$$
$$= \mathcal{UNIT}^{+}(x)$$

and, similarly,

$$x \xrightarrow{f} f(x) = -1\,x^{0}$$
$$= -1$$
$$= -1$$
$$= \mathcal{UNIT}^{-}(x)$$

- When exponent is $+1$ we have $x \xrightarrow{f} f(x) = +1\,x^{+1}$

$$= +1 \odot x$$
$$= x$$
$$= \mathcal{IDENTITY}(x)$$

regular power function
characteristic
parity
even power function
odd power function
type

and, similarly,

$$x \xrightarrow{f} f(x) = -1\, x^{+1}$$
$$= -1 \odot x$$
$$= \text{opp } x$$
$$= \mathcal{OPPOSITE}(x)$$

□

3. Regular power functions. In contrast with the exceptional power functions, we will call **regular power functions** those power functions for which exponent is *any* signed counting number *other* than 0 and +1

a. Along with Sign Coefficient and Sign exponent, the third **characteristic** of a regular power function, will be **parity** exponent:

- An **even power function** is a *regular* power function whose exponent is an *even* (whole) number,
- An **odd power function** is a *regular* power function whose exponent is an *odd* (whole) number.

b. From the point of view of these three characteristics, there will therefore be *eight* **types** of regular power functions but the order of the characteristics in the type

Sign exponent Parity exponent Sign Coefficient

is not quite the order in the output but, as we will see, is the order of importance in getting the global graph.

Type	Sign exponent	Parity exponent	Sign Coefficient	Output
+ even +	+	even	+	$+1\, x^{+\text{even}}$
+ even −			−	$-1\, x^{+\text{even}}$
+ odd +		odd	+	$+1\, x^{+\text{odd}}$

primary power function

+ odd −			−	$-1\ x^{+\text{odd}}$
− even +	−	even	+	$+1\ x^{-\text{even}}$
− even −			−	$-1\ x^{-\text{even}}$
− odd +		odd	+	$+1\ x^{-\text{odd}}$
− odd −			−	$-1\ x^{-\text{odd}}$

c. Low degree power functions We will now discuss the eight **primary power functions**, that is the *regular* power function with lowest size exponent for each type.

d. Secondary power functions
https://en.wikipedia.org/wiki/Exponentiation#Power_functions

1 Functions $\mathcal{SQUARING}^{+}$ and $\mathcal{SQUARING}^{-}$

DEFINITION 6.2 The **squaring functions** are the power functions with *exponent* $+2$ namely:

$$x \xrightarrow{\mathcal{SQUARING}^{+}} \mathcal{SQUARING}^{+}(x) = +1\ x^{+2}$$

and

$$x \xrightarrow{\mathcal{SQUARING}^{-}} \mathcal{SQUARING}^{-}(x) = -1\ x^{+2}$$

1. Control point(s).

2. Global graphs.

3. Local graphs.

4. Local I-O rules.

5. Local features.

2 Functions $\mathcal{CUBING}^+$ and $\mathcal{CUBING}^-$

Cubing function
reciprocating function

1. Global I-O rules.

DEFINITION 6.3are the power functions with *exponent* $+3$, namely:

$$x \xrightarrow{\mathcal{CUBING}^+} \mathcal{CUBING}^+(x) = +1\, x^{+3}$$

and

$$x \xrightarrow{\mathcal{CUBING}^-} \mathcal{CUBING}^-(x) = -1\, x^{+3}$$

2. Global graphs.

3. Control point(s).

4. Local graphs.

5. Local I-O rules.

6. Local features.

3 Functions $\mathcal{RECIPROCAL}^+$ and $\mathcal{RECIPROCAL}^-$

1. Global I-O rules.

DEFINITION 6.4 Reciprocating Functions are the power functions with *exponent* -1, namely:

$$x \xrightarrow{\mathcal{RECIP}^+} \mathcal{RECIP}^+(x) = +1\, x^{-1}$$

and

$$x \xrightarrow{\mathcal{RECIP}^-} \mathcal{RECIP}^-(x) = -1\, x^{-1}$$

square-reciprocating functions

2. Global graphs.

3. Control point(s).

4. Local graphs.

5. Local I-O rules.

6. Local features. SQUARERECIP

4 The Functions $\mathcal{SQUARE\text{–}RECIP}^{+}$ and $\mathcal{SQUARE\text{–}RECIP}^{-}$

1. Global I-O rules.

Definition 6.5 Square-reciprocating Functionsare the power functions with *exponent* -2, namely:

$$x \xrightarrow{\mathcal{SQUARE-RECIP}^{+}} \mathcal{SQUARE-RECIP}^{+}(x) = +1\,x^{-2}$$

and

$$x \xrightarrow{\mathcal{SQUARE-RECIP}^{-}} \mathcal{SQUARE-RECIP}^{-}(x) = -1\,x^{-2}$$

2. Control point(s).

3. Global graphs.

4. Local graphs.

5. Local I-O rules.

6. Local features.

5 Secondary Regular Power Functions

1. Global I-O rules.

2. Global graphs.

3. Local graphs.

4. Local I-O rules.

5. Local features.
===========OK SO FAR ============

6. Output *at* x_0. With ??, PROCEDURE 5.3 becomes:

PROCEDURE 6.1
To get the output *at* x_0 for the power function given by the I-O rule

$$x \xrightarrow{f} f(x) = \pm 1\, x^{\pm n}$$

i. Declare x_0:

$$x\Big|_{x \leftarrow x_0} \xrightarrow{f} f(x)\Big|_{x \leftarrow x_0} = \pm 1\, x^{\pm n}\Big|_{x \leftarrow x_0}$$

ii. Write the expression particular to x_0:

$$x_0 \xrightarrow{f} f(x_0) = \pm 1\, x_0^{\pm n}$$

iii. Execute the particular expression according to $\pm n$:

- $+$ says to *multiply* ± 1 by the n copies of x_0:

$$x_0 \xrightarrow{f} f(x_0) = \pm 1 \odot \underbrace{x_0 \odot \ldots \odot x_0}_{n \text{ copies of } x_0}$$

- 0 says to leave ± 1 alone:

$$x_0 \xrightarrow{f} f(x_0) = \pm 1$$

- $-$ says to *divide* ± 1 by the n copies of x_0:

$$x_0 \xrightarrow{f} f(x_0) =$$

$$\underbrace{\frac{\pm 1}{x_0 \odot \ldots \odot x_0}}_{n \text{ copies of } x_0}$$

DEMO 6.1a
To get the output *at* -3 of the power function whose I-O rule is

$$x \xrightarrow{DIP} DIP(x) = +1\, x^{+6} \; :$$

i. Declare -3:

$$x\Big|_{x \leftarrow -3} \xrightarrow{DIP} DIP(x)\Big|_{x \leftarrow -3} = +1\, x^{+6}\Big|_{x \leftarrow -3}$$

ii. Write the expression particular to -3:

$$-3 \xrightarrow{DIP} DIP\left(-3\right) = +1\, -3^{+6}$$

iii. Execute the particular expression according to what $+6$ says:

- $+$ says to *multiply* $+1$ by the 6 copies of -3:

$$-3 \xrightarrow{DIP} DIP\left(-3\right) = +1 \odot \underbrace{-3 \odot -3 \odot -3 \odot -3 \odot -3 \odot -3}_{6 \text{ copies of } -3}$$

$$= +729$$

DEMO 6.1b
To get the output *at* -3 of the power function whose I-O rule is

$$x \xrightarrow{FIP} FIP(x) = -1\, x^{0} \; :$$

i. Declare -3:

$$x\Big|_{x \leftarrow -3} \xrightarrow{FIP} FIP(x)\Big|_{x \leftarrow -3} = -1\, x^{0}\Big|_{x \leftarrow -3}$$

ii. Write the expression particular to -3:

$$-3 \xrightarrow{FIP} FIP(-3) = -1 \; -3^{\,0}$$

iii. Execute the particular expression according to ^{0}:

- ^{0} says to leave -1 alone:

$$-3 \xrightarrow{FIP} FIP(-3) = -1$$

DEMO 6.1c
To get the output *at* -3 of the power function whose I-O rule is
$x \xrightarrow{GIP} GIP(x) = +1 \; x^{-5}$:

i. Declare -3:

$$x\Big|_{x\leftarrow -3} \xrightarrow{GIP} GIP(x)\Big|_{x\leftarrow -3} = +1 \; x^{-5}\Big|_{x\leftarrow -3}$$

ii. Write the expression particular to -3:

$$-3 \xrightarrow{GIP} GIP(-3) = +1 \; x_0^{-5}$$

iii. Execute the particular expression according to $^{-5}$:

- $^{-}$ says to *divide* $+1$ by the ^{5} copies of -3:

$$-3 \xrightarrow{GIP} GIP(-3) = \underbrace{\frac{+1}{-3 \odot -3 \odot -3 \odot -3 \odot -3}}_{5 \text{ copies of } -3}$$

$$= -0.004\,115\,226\,337\,449 + [\ldots]$$

Only just in case you wanted to see lots of decimals.

Just in case you forgot the input could be a decimal number.

DEMO 6.1d
To get the output *at* -0.2 of the power function whose I-O rule is
$x \xrightarrow{GIP} GIP(x) = +1 \; x^{-5}$:

i. Declare -0.2:

$$x\Big|_{x\leftarrow -0.2} \xrightarrow{GIP} GIP(x)\Big|_{x\leftarrow -0.2} = +1\ x^{-5}\Big|_{x\leftarrow -0.2}$$

ii. Write the expression particular to -0.2:

$$-0.2 \xrightarrow{GIP} GIP(-0.2) = +1\ x_0^{-5}$$

iii. Execute the particular expression according to $^{-5}$:

- $-$ says to *divide* $+1$ by the 5 copies of -0.2:

$$-0.2 \xrightarrow{GIP} GIP(-0.2) = \underbrace{\frac{+1}{-0.2 \odot -0.2 \odot -0.2 \odot -0.2 \odot -0.2}}_{5 \text{ copies of } -0.2}$$
$$= -3\,125$$

=======Begin HOLDING=======

7. Output *near* x_0.

8. Output *near* ∞.

i. When we want to thicken only one side of ∞, we proceed as follows:

PROCEDURE 6.2

1. *Normalize* the global input-input rule using **??** on **??**
2. *Declare* that x is to be replaced by $+large$ or $-large$
3. *Execute* the output-specifying code that is:
 a. *Decode* the output-specifying code, that is write out the computations to be performed according to the output-specifying code.
 b. *Perform* the computations given by the code using **??** on **??** and **??** on **??** or **??** on **??**

DEMO 6.2

i. We *normalize* $NADE$:

$$x \xrightarrow{NADE} NADE(x) = (-mediumsize)\ x^{-odd}$$

ii. We *declare* that x is to be replaced by $+large$

$$x\Big|_{x\leftarrow +large} \xrightarrow{NADE} NADE(x)\Big|_{x\leftarrow +large} = (-mediumsize)x^{-odd}\Big|_{x\leftarrow +large}$$

which, once carried out, gives:

$$+large \xrightarrow{NADE} NADE(+large) = \underbrace{(-mediumsize)(+large)^{-\,odd}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $negative$, we get the output $NADE(+large)$ by *dividing* the coefficient $-mediumsize$ by an odd number of copies of the given input $+large$:

$$= \frac{-mediumsize}{\underbrace{(+large)\cdot\ \ldots\ \cdot(+large)}_{odd \text{ number of copies of } +large}}$$

b. We *perform* the computations given by the code. Dealing separately with the $signs$ and the $sizes$, we have

$$= \frac{-mediumsize}{\underbrace{(+)\cdot\ \ldots\ \cdot(+)}_{odd\ \text{number of copies of}\ +} \cdot \underbrace{(\,large\,)\cdot\ \ldots\ \cdot(\,large\,)}_{odd\ \text{number of copies of}\ large}}$$

and since,

- by **??** on **??**, *any* number of copies of $+$ multiply to $+$,
- by the **DDEFINITION** of $large$, *any* number of copies of $large$ multiply to $large$

$$= \frac{-mediumsize}{+\cdot large}$$

and by **??** on **??** and **??** on **??** we get

$$= -small$$

iv. The input-output pairs are $(+large, -small)$

DEMO 6.3 Let $RADE$ be the function given by the global input-output rule

$$x \xrightarrow{RADE} RADE(x) = (+45.67)x^{-4}$$

To get the input-output pairs near $-\infty$ for $RADE$:

i. We *normalize* $RADE$:

$$x \xrightarrow{RADE} RADE(x) = (+mediumsize)\, x^{-even}$$

ii. We *declare* that x is to be replaced by $-large$

$$x\Big|_{x \leftarrow -large} \xrightarrow{RADE} RADE(x)\Big|_{x \leftarrow -large} = (+mediumsize)x^{-even}\Big|_{x \leftarrow -large}$$

which, once carried out, gives:

$$-large \xrightarrow{RADE} RADE(-large) = \underbrace{(+mediumsize)(-large)^{-even}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $negative$, we get the output $RADE(-large)$ by *dividing* the coefficient $+mediumsize$ by an $even$ number of copies of the given input $-large$:

$$= \frac{+mediumsize}{\underbrace{(-large) \cdot \ \ldots \ \cdot (-large)}_{even \text{ number of copies of } -large}}$$

b. We *perform* the computations given by the code:

Dealing separately with the $signs$ and the $sizes$, we have

$$= \frac{+mediumsize}{\underbrace{(-) \cdot \ \ldots \ \cdot (-)}_{even \text{ number of copies of } -} \cdot \underbrace{(large) \cdot \ \ldots \ \cdot (l}_{even \text{ number of copies }}}$$

and since,

- by the **Sign Multiplication Rule**, any *even* number of copies of $-$ multiply to $+$
- by the **DDEFINITION** of $large$, *any* number of copies of $large$ multiply to $arge$

$$= \frac{+mediumsize}{+ \cdot large}$$

and by the **Sign Division Rule** and the **Size Division Theorem**

$$= +small$$

iv. The input-output pairs are $(-large, +small)$

ii. When we want to thicken both sides of ∞, we declare that x is to be replaced by $\pm large$ and keep track of the signs as we *perform the computations* given by the output-specifying code.

DEMO 6.4 Let $DADE$ be the function given by the global input-output rule

$$x \xrightarrow{DADE} DADE(x) = (-83.91)x^{+5}$$

To get the input-output pairs *near* ∞ for $DADE$:

i. We *normalize* $DADE$:

$$x \xrightarrow{DADE} DADE(x) = (-mediumsize)\, x^{+odd}$$

ii. We *declare* that x is to be replaced by $\pm large$

$$x\Big|_{x\leftarrow \pm large} \xrightarrow{DADE} DADE(x)\Big|_{x\leftarrow \pm large} = (-mediumsize)x^{+odd}\Big|_{x\leftarrow \pm large}$$

which, once carried out, gives:

$$\pm large \xrightarrow{DADE} DADE(\pm large) = \underbrace{(-mediumsize)(\pm large)^{+\,odd}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $positive$, we get that the output $DADE(\pm large)$ is obtained by *multiplying* the coefficient $-mediumsize$ by an odd number of copies of the given input $\pm large$:

$$= (-mediumsize) \cdot \underbrace{(\pm large) \cdot \ \ldots \ \cdot (\pm large)}_{odd \text{ number of copies of } \pm large}$$

b. We *perform* the computations given by the code. Dealing separately with the $signs$ and the $sizes$, we have

$$= (-mediumsize) \cdot \underbrace{(\pm) \cdot \ \ldots \ \cdot (\pm)}_{odd \text{ number of copies of } \pm} \cdot \underbrace{(\,large\,) \cdot \ \ldots \ \cdot (\,l}_{odd \text{ number of copies o}}$$

and since,

- by the **Sign Multiplication Rule**, an *odd* number of copies of $+$ multiply to $+$ and an *odd* number of copies of $-$ multiply to $-$
- by the **DDEFINITION** of $large$, *any* number of copies of $large$ multiply to $large$

$$= (-mediumsize) \cdot \pm \cdot large$$

and by the **Sign Multiplication Rule** and the **Size Multiplication Theorem**

$$= \mp large$$

iv. The input-output pairs are $(\pm large, \mp large)$

DEMO 6.5 Let $PADE$ be the function given by the global input-output rule

$$x \xrightarrow{PADE} PADE(x) = (-65.18)x^{+6}$$

To get the input-output pairs *near* ∞ for $PADE$

i. We *normalize* $PADE$.

$$x \xrightarrow{PADE} PADE(x) = (-mediumsize)\, x^{+even}$$

ii. We *declare* that x is to be replaced by $\pm large$

$$x \Big|_{x \leftarrow \pm large} \xrightarrow{PADE} PADE(x) \Big|_{x \leftarrow \pm large} = (-mediumsize)x^{+even} \Big|_{x \leftarrow \pm large}$$

which, once carried out, gives:

$$\pm large \xrightarrow{PADE} PADE(\pm large) = \underbrace{(-mediumsize)(\pm large)^{+\,even}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $positive$, we get the output $PADE(\pm large)$ by *multiplying* the coefficient $-mediumsize$ by an $even$ number of copies of the given input $\pm large$:

$$= (-mediumsize) \cdot \underbrace{(\pm large) \cdot \ldots \cdot (\pm large)}_{even \text{ number of copies of } \pm large}$$

b. We *perform* the computations given by the code. Dealing separately with the *signs* and the *sizes*, we have

$$= (-mediumsize) \cdot \underbrace{(\pm) \cdot \ \dots \ \cdot (\pm)}_{even \text{ number of copies of } \pm} \cdot \underbrace{(\ large\ ) \cdot \ \dots \ \cdot (}_{even \text{ number of copies}}$$

and since,

- by the **Sign Multiplication Rule**, an *even* number of copies of $+$ multiply to $+$ and an *even* number of copies of $-$ multiply to $+$
- by the **DDEFINITION** of $large$, *any* number of copies of $large$ multiply to $large$

$$= (-mediumsize) \cdot + \cdot large$$

and by the **Sign Division Rule** and the **Size Division Theorem**

$$= -large$$

iv. The input-output pairs are $(\pm large, -large)$

9. Output *near* 0.

i. When we want to thicken only one side of 0, we proceed as follows:

Procedure 6.3

1. *Normalize* the global input-input rule using **??** on **??**
2. *Declare* that x is to be replaced by $+small$ or $-small$
3. *Execute* the output-specifying code that is:

a. *Decode* the output-specifying code, that is write out the computations to be performed according to the output-specifying code.

b. *Perform* the computations given by the code using **??** on **??** and **??** on **??** or **??** on **??**

Demo 6.6

i. We *normalize* $MADE$:

$$x \xrightarrow{MADE} MADE(x) = (+mediumsize)\, x^{+odd}$$

ii. We *declare* that x is to be replaced by $+small$

$$x \Big|_{x \leftarrow +small} \xrightarrow{MADE} MADE(x) \Big|_{x \leftarrow +small} = (+mediumsize) x^{+odd} \Big|_{x \leftarrow +small}$$

which, once carried out, gives:

$$+small \xrightarrow{MADE} MADE(+small) = \underbrace{(-mediumsize)(+small)^{+\,odd}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $positive$, we get that the output $MADE(+small)$ is obtained by *multiplying* the coefficient $+mediumsize$ by an odd number of copies of the given input $+small$:

$$= (+mediumsize) \cdot \underbrace{(+small) \cdot \ \ldots \ \cdot (+sm}_{odd \text{ number of copies of } +sm}$$

b. We *perform* the computations given by the code. Dealing separately with the $signs$ and the $sizes$, we have

$$= (+mediumsize) \cdot \underbrace{(+) \cdot \ \ldots \ \cdot (+)}_{odd \text{ number of copies of } +} \cdot \underbrace{(small}_{odd \text{ numb}}$$

and since,

- by the **Sign Multiplication Rule**, *any* number of copies of $+$ multiply to $+$
- by the **DDEFINITION** of $small$, *any* number of copies of $small$ multiply to $small$

$$= (+mediumsize) \cdot + \cdot small$$

and by the **Sign Multiplication Rule** and the **Size Multiplication Theorem**

$$= +small$$

iv. The input-output pairs are $(+small, -small)$

DEMO 6.7 Let $WADE$ be the function given by the global input-output rule

$$x \xrightarrow{WADE} WADE(x) = (-28.34)x^{-3}$$

To get the output of $WADE$ near 0^+

i. We *normalize* $WADE$:

$$x \xrightarrow{WADE} WADE(x) = (-mediumsize)\, x^{-even}$$

ii. We *declare* that x is to be replaced by $+small$

$$x\Big|_{x\leftarrow +small} \xrightarrow{WADE} WADE(x)\Big|_{x\leftarrow +small} = (-mediumsize)x^{-even}\Big|_{x\leftarrow +small}$$

which, once carried out, gives:

$$+small \xrightarrow{WADE} WADE(+small) = \underbrace{(-mediumsize)(+small)^{-even}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $negative$, we get the output $WADE(+small)$ by *dividing* the coefficient $-mediumsize$ by an $even$ number of copies of the given input $+small$:

$$= \frac{-mediumsize}{\underbrace{(+small)\cdot \ldots \cdot (+small)}_{even \text{ number of copies of } +small}}$$

iv. **b.** We *perform* the computations given by the code. Dealing separately with the *signs* and the *sizes*, we have

$$= \frac{-mediumsize}{\underbrace{(+)\cdot \ldots \cdot (+)}_{even \text{ number of copies of } +} \cdot \underbrace{(small)\cdot \ldots \cdot (small)}_{even \text{ number of copies of } small}}$$

and since,

- by the **Sign Multiplication Rule**, *any* number of copies of $+$ multiply to $+$
- by the **DDEFINITION** of $small$, *any* number of copies of $small$ multiply to $small$

$$= \frac{-mediumsize}{+\cdot small}$$

and by the **Sign Division Rule** and the **Size Division Theorem**

$$= -large$$

iv. The input-output pairs are $(+small, -large)$

ii. When we want to thicken both sides, we will declare that x is to be replaced by $\pm small$ and keep track of the signs as we *perform the computations* given by the output-specifying code.

DEMO 6.8 Let $JADE$ be the function given by the global input-output rule

$$x \xrightarrow{JADE} JADE(x) = (-65.71)x^{-5}$$

To get the output of $JADE$ near 0,

i. We *normalize* $JADE$:

$$x \xrightarrow{JADE} JADE(x) = (-mediumsize)\ x^{-odd}$$

ii. We *declare* that x is to be replaced by $\pm small$

$$x\Big|_{x\leftarrow \pm small} \xrightarrow{JADE} JADE(x)\Big|_{x\leftarrow \pm small} = (-mediumsize)x^{-odd}\Big|_{x\leftarrow \pm small}$$

which, once carried out, gives:

$$\pm small \xrightarrow{JADE} JADE(\pm small) = \underbrace{(-mediumsize)(\pm small)^{-odd}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $negative$, we get the output $JADE(\pm small)$ by *dividing* the coefficient $-mediumsize$ by an odd number of copies of the given input $\pm small$:

$$= \frac{-mediumsize}{\underbrace{(\pm small)\cdot \ \ldots\ \cdot(\pm small)}_{odd \text{ number of copies of } \pm small}}$$

b. We *perform* the computations given by the code. Dealing separately with the $signs$ and the $sizes$, we have

$$= \frac{-mediumsize}{\underbrace{(\pm)\cdot\ \ldots\ \cdot(\pm)}_{odd\text{ number of copies of }\pm} \cdot \underbrace{(small)\cdot\ \ldots\ \cdot(s}_{odd\text{ number of copies of}}}$$

and since,

- by the **Sign Multiplication Rule**, an *odd* number of copies of $+$ multiply to $+$ and an *odd* number of copies of $-$ multiply to $-$
- by the **DDEFINITION** of $small$, *any* number of copies of $small$ multiply to $small$

$$= \frac{-mediumsize}{\pm \cdot small}$$

and by the **Sign Division Rule** and the **Size Division Theorem**

$$= \mp large$$

iv. The input-output pairs are $(\pm small, \mp large)$

DEMO 6.9 Let $FADE$ be the function given by the global input-output rule

$$x \xrightarrow{FADE} FADE(x) = (-65.18)x^{+6}$$

To get the input-output pairs *near* 0 for $FADE$:

i. We *normalize* $FADE$.

$$x \xrightarrow{FADE} FADE(x) = (-mediumsize)\, x^{+even}$$

ii. We *declare* that x is to be replaced by $\pm small$

$$x \Big|_{x \leftarrow \pm small} \xrightarrow{FADE} FADE(x) \Big|_{x \leftarrow \pm small} = (-mediumsize)x^{+even} \Big|_{x \leftarrow \pm small}$$

which, once carried out, gives:

$$\pm small \xrightarrow{FADE} FADE(\pm small) = \underbrace{(-mediumsize)(\pm small)^{+\,even}}_{\text{output-specifying code}}$$

iii. We *execute* the output-specifying code that is:

a. We *decode* the output-specifying code: since the exponent is $positive$, we get the output $FADE(-small)$ by *multiplying* the coefficient $-mediumsize$ by an $even$ number of copies of the given input $\pm small$:

$$= (-mediumsize) \cdot \underbrace{(\pm small) \cdot \ \ldots \ \cdot (\pm small)}_{even \text{ number of copies of } \pm small}$$

b. We *perform* the computations given by the code. Dealing separately with the $signs$ and the $sizes$, we have

$$= (-mediumsize) \cdot \underbrace{(\pm) \cdot \ \ldots \ \cdot (\pm)}_{even \text{ number of copies of } \pm} \cdot \underbrace{(small) \cdot \ \ldots \ \cdot}_{even \text{ number of copie}}$$

Input Sign
Input Size
Output Sign
Output Size

and since,

- by the **Sign Multiplication Rule**, an *even* number of copies of $+$ multiply to $+$ and an *even* number of copies of $-$ multiply to $+$
- by the **DDEFINITION** of $small$, *any* number of copies of $small$ multiply to $small$

$$= (-mediumsize) \cdot + \cdot small$$

and by the **Sign Multiplication Rule** and the **Size Multiplication Theorem**

$$= -small$$

iv. The input-output pairs are $(\pm small, -small)$

=======End HOLDING =======

Really! In spite of appearances!

10. Output Sign and Output Size The following will make our life a bit easier:

AGREEMENT 6.1

- Instead of saying "the sign of an input", we will use the entity **Input Sign**
- Instead of saying "the size of an input", we will use the entity **Input Size**

and of course, similarly,

- Instead of saying "the sign of an output", we will use the entity **Output Sign**
- Instead of saying "the size of an output", we willuse the entity **Output Size**

However, in all other cases, we will continue to say "the sign of . . . " or just "Sign . . . " and "the size of . . . " or just "Size . . . " and "the parity of . . . " or just "Parity . . . ".

i. Output Sign Output Sign will play a central role in discussing the behavior of functions and, in the case of regular power functions,t we have:

But it's not going to be alway that *easy to get!*

THEOREM 6.2 Output Sign for *regular* power functions

- If Input Sign is $+$, then Output Sign will be the *same* as Sign Coefficient

- If Input Sign is $-$, then Output Sign will depend on Parity *exponent*:
  - If *exponent* is *even*, then Output Sign will be the *same* as Sign Coefficient,
  - If *exponent* is *odd*, then Output Sign will be the *opposite* of Sign Coefficient.

EXAMPLE 6.7. Let $\mathcal{KIP}$ be the regular power function given by the I-O rule $x \xrightarrow{\mathcal{KIP}} \mathcal{KIP}(x) = +1\,x^{+3}$

Then THEOREM 6.2 - Output Sign for *regular* power functions (Page 278)), gives that, since $+3$ is *odd*,

- If Input Sign is $+$, Output Sign is $+$ (The *same* as Sign $+1$),
- If Input Sign is $-$, Output Sign is $-$ (The *opposite* of Sign $+1$).

And indeed, we get directly from DEFINITION 6.1 - Power Functions (Page 257)) that:

$$+ \xrightarrow{\mathcal{KIP}} \mathcal{KIP}(+) = + \odot + \odot + \odot +$$
$$= + \odot +$$
$$= +$$

and

$$- \xrightarrow{\mathcal{KIP}} \mathcal{KIP}(-) = + \odot - \odot - \odot -$$
$$= + \odot -$$
$$= -$$

EXAMPLE 6.8. Let $\mathcal{KIT}$ be the regular power function given by the I-O rule $x \xrightarrow{\mathcal{KIT}} \mathcal{KIT}(x) = +1\,x^{-4}$

Then THEOREM 6.2 - Output Sign for *regular* power functions (Page 278), gives, since -4 is *even*, that:

- If Input Sign is $+$, Output Sign is $+$ (The same as Sign $+1$),
- If Input Sign is $-$, Output Sign is $+$ (The *same* as Sign $+1$).

And indeed, we get directly from DEFINITION 6.1 - Power Functions (Page 257) that:

$$+ \xrightarrow{\mathcal{KIT}} \mathcal{KIT}(+) = \frac{+}{+ \odot + \odot + \odot +} = \frac{+}{+} = +$$

and

$$- \xrightarrow{\mathcal{KIT}} \mathcal{KIT}(-) = \frac{+}{- \odot - \odot - \odot -} = \frac{+}{+} = +$$

ii. Output Size For the purposes of CALCULUS ACCORDING TO THE REAL WORLD, we will mostly need qualitative sizes, particularly near ∞ and near 0. In the case of regular power functions we have:

But it's not going to be alway that easy to get!

THEOREM 6.3 Output Size for *regular* power functions

Output Size depends *always* on Sign ^{exponent}:

- If Sign ^{exponent} is $+$, then:
  Output Size will be the *same* as Input Size.
- If Sign ^{exponent} is $-$, then:
  Output Size will be the *reciprocal* of Input Size.

Proof. The proof goes along the lines of the following two EXAMPLES, is left to the reader. ☐

EXAMPLE 6.9. Let $\mathcal{KIN}$ be the regular power function given by the I-O rule $x \xrightarrow{\mathcal{KIN}} \mathcal{KIN}(x) = -1\, x^{+5}$

Then THEOREM 6.3 - Output Size for *regular* power functions (Page 280) gives, since Sign $^{+5}$ is $+$, that Output Size will be the *same* as Input Size. So:

- When Input Size will be *large*, Output Size will be *large*

- When Input Size will be *small*, Output Size will be *small* .

And indeed, directly from DEFINITION 6.1 - Power Functions (Page 257) we get:

$$large \xrightarrow{\mathcal{KIN}} \mathcal{KIN}(large) = 1\, large^{+5}$$
$$= 1 \odot large \odot large \odot large \odot large \odot large$$
$$= 1 \odot large$$
$$= large$$

and

$$small \xrightarrow{\mathcal{KIN}} \mathcal{KIN}(small) = 1\, small^{+5}$$
$$= 1 \odot small \odot small \odot small \odot small \odot small$$
$$= 1 \odot small$$
$$= small$$

EXAMPLE 6.10. Let $\mathcal{KIM}$ be the regular power function given by the I-O rule $x \xrightarrow{\mathcal{KIM}} \mathcal{KIM}(x) = +1\, x^{-4}$

Then THEOREM 6.3 - Output Size for *regular* power functions (Page 280) gives, since Sign $^{-4}$ is $^{-}$, that Output Size will be the *reciprocal* of Input Size. So:

- When Input Size will be *large*, Output Size will be *small*
- When Input Size will be *small*, Output Size will be *large* .

DEFINITION 6.1 - Power Functions (Page 257) we get:

$$large \xrightarrow{\mathcal{KIM}} \mathcal{KIM}\left(large\right) = 1\, large^{-4}$$
$$= \frac{1}{large \odot large \odot large \odot large}$$
$$= \frac{1}{large}$$

$$= small$$

and

$$small \xrightarrow{\mathcal{KIM}} \mathcal{KIM}\left(small\right) = 1\ small^{-4}$$
$$= \frac{1}{small \odot small \odot small \odot small}$$
$$= \frac{1}{small}$$
$$= large$$

iii. Quincunx In particular, using the fact that large implies larger than 1 and small implies smaller than 1, we have, relative to the quincunx:

> **THEOREM 6.3 (Restated) Output Size for *regular* power functions**
> - If Input Size is *larger* than 1, then:
>   - If Sign ^{exponent} is $+$, then Output Size will be *larger* than 1
>   - If Sign ^{exponent} is $-$, then Output Size will be *smaller* than 1
> - If Input Size *is* 1, then: Output Size will be also 1
> - If Input Size is *smaller* than 1, then:
>   - If Sign ^{exponent} is $+$, then Output Size will be *smaller* than 1
>   - If Sign ^{exponent} is $-$, then Output Size will be *larger* than 1

Proof. The proof is of course just a tiny little bit more complicated but still goes along the lines of the following two EXAMPLES and ... is left to the reader. ☐

EXAMPLE 6.11. Let $\mathcal{KIS}$ be the regular power function given by the I-O rule

$$x \xrightarrow{\mathcal{KIS}} \mathcal{KIS}\,(x) = -1\ x^{+3}$$

Then, directly from DEFINITION 6.1 - Power Functions (Page 257) we get:

$$1 < large \xrightarrow{\mathcal{KIS}} \mathcal{KIS}\left(large\right) = -1\ large^{+3}$$
$$= 1 \ \odot\ large \odot large \odot large$$

$$= 1 \odot large$$
$$= large > 1$$

Just in case: "Jill is older than Jack" *and* "Jack is younger than Jill" *say exactly the same thing.*

and

$$1 > small \xrightarrow{\mathcal{KIS}} \mathcal{KIS}\left(small\right) = -1\ small^{+3}$$
$$= 1 \odot small \odot small \odot small$$
$$= 1 \odot small$$
$$= small < 1$$

EXAMPLE 6.12. Let $\mathcal{KIT}$ be the regular power function given by the I-O rule
$x \xrightarrow{\mathcal{KIT}} \mathcal{KIT}(x) = +1\ x^{-4}$

DEFINITION 6.1 - Power Functions (Page 257) we get:

$$1 < large \xrightarrow{\mathcal{KIT}} \mathcal{KIT}\left(large\right) = 1\ large^{-4}$$
$$= \frac{1}{large \odot large \odot large \odot large}$$
$$= \frac{1}{large}$$
$$= small < 1$$

Again, "Jill is older than Jack" *and* "Jack is younger than Jill" *say exactly the same thing.*

and

$$1 > small \xrightarrow{\mathcal{KIM}} \mathcal{KIM}\left(small\right) = 1\ small^{-4}$$
$$= \frac{1}{small \odot small \odot small \odot small}$$
$$= \frac{1}{small}$$
$$= large > 1$$

==========OK SO FAR ===========

=======Begin WORK ZONE=======

horizontal flip

11. Symmetries.

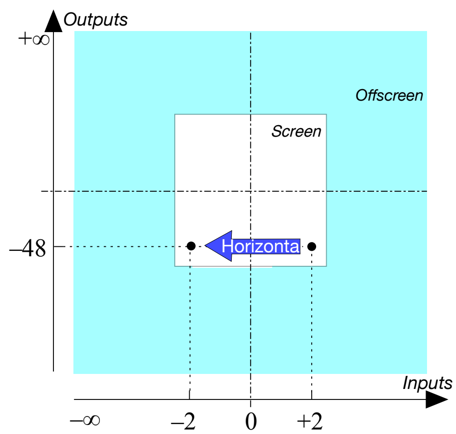
In order to halve the work in graphing regular power functions more efficiently, we need to invest a little bit on a couple of graphic maneuvers:

i. Horizontal Flip

If we do a **horizontal flip** on a first *plot dot* we get a second *plot dot* and

- The *input* of the second plot dot will be the *opposite* of the input of the first plot dot
- The *output* of the second plot dot will be the *same* as the output of the first plot dot

EXAMPLE 6.13. If we do a *horizontal flip* on a the plot dot $(+2, -48)$ we will get a second plot dot and:



- the *input* of the second plot dot will be -2
- the *output* of the second plot dot will be -48

EXAMPLE 6.14. Given the function specified by the global input-ouput rule

$$x \xrightarrow{KAT} KAT(x) = (-3) \cdot x^{+4}$$

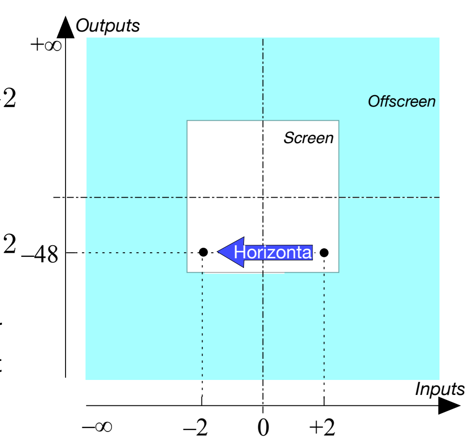
i. For instance

$$+2 \xrightarrow{KAT} KAT(+2) = -3 \bullet +2 \bullet +2 \bullet +2 \bullet +2$$
$$= -48$$

and

$$-2 \xrightarrow{KAT} KAT(-2) = -3 \bullet -2 \bullet -2 \bullet -2 \bullet -2$$
$$= -48$$

ii. We see that we can get the plot dot for input -2 by a *horizontal* flip of the plot dot for input $+2$:



THEOREM 6.4 Even regular power functions are horizontally symmetrical

vertical flip
diagonal flip

Proof. □

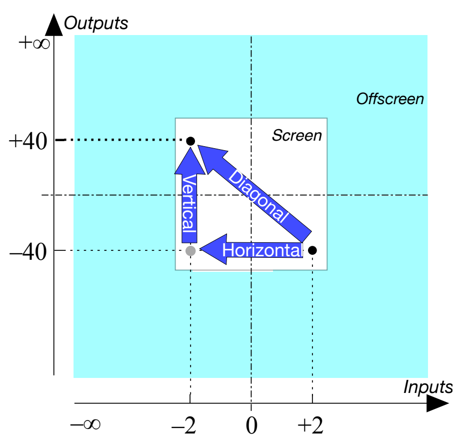
ii. Vertical Flip

iii. Diagonal Flip If we follow the horizontal flip on the *first plot dot* by a **vertical flip** on the *second plot dot*, we will get a *third plot dot* and:

- the *input* of the third plot dot will be the *same* as the *input* of the second plot dot, that is the *opposite* of the input of the first plot dot
- the *output* of the third plot dot will be the *opposite* of the *output* of the second plot dot, that is the *opposite* of the output of the first plot dot

In other words, we can get the third plot dot by a **diagonal flip** on the first plot dot.

EXAMPLE 6.15. If we do a *horizontal flip* on the plot dot $(+2, -48)$ we get a second plot dot and if we follow by a vertical flip on the second plot dot, we get a third plot dot and:



- the *input* of the second plot dot will be -2
- the *output* of the second plot dot will be -40

and then

- the *input* of the third plot dot will be -2
- the *output* of the third plot dot will be $+40$

In other words, both the *input* and the *output* of the third plot dot are *opposite* of the input and output of the first plot dot and so to get the third plot dot directly from the first plot dot we can just use a *diagonal flip* instead of a horizontal flip followed by a vertical flip.

EXAMPLE 6.16. Given the function specified by the global input-ouput rule

$$x \xrightarrow{KAT} KAT(x) = (+5) \cdot x^{+3}$$

opposite input

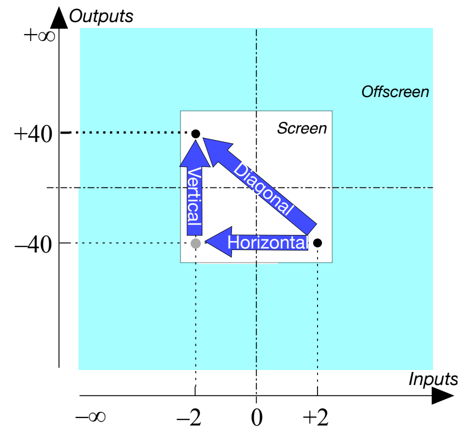
a. For instance

$$+2 \xrightarrow{KAT} KAT(+2) = +5 \bullet +2 \bullet +2 \bullet +2$$
$$= +40$$

and

$$-2 \xrightarrow{KAT} KAT(-2) = +5 \bullet -2 \bullet -2 \bullet -2$$
$$= -40$$

b. We see that we can get the plot dot for input -2 by a *diagonal flip* of the plot dot for input $+2$:



> **THEOREM 6.5 Odd regular power functions are diagonally symmetrical**

Proof. □

So a consequence of **??** on **??** is that once we have the plot dot for an input, we can get the plot dot for the **opposite input**, that is for the input with the *same* size and *opposite* sign with just one flip:

> **THEOREM 6.6 Symmetry** (For Regular Monomial Functions.) Given the plot dot for an input, we get the plot dot for the *opposite input* with:
> - A *horizontal*-flip if Exponent Parity = *even*,
> - A *diagonal*-flip if Exponent Parity = *odd*.

While, as we saw in Section 1 - Height-Continuity (Page 165), getting the output for a given number is not very useful, it does allow to prove the very useful

> **THEOREM 6.7** The global graphs of *all* power functions go through two of the corner plot dots of the quincunx.

Proof. There are four cases:

▸ For the power functions of type

$$x \xrightarrow{f} f(x) = +1\ +1^{+5}$$

we have

$$x \xrightarrow{f} f(x) = +1\ {+1}^{+5}$$
$$= +1 \odot +1 \odot +1 \odot +1 \odot +1 \odot +1$$

- The other three cases are left to the reader

□

========End WORK ZONE=======

6 Graphing Power Functions

=============OK SO FAR ===============

BEGIN WORK========BEGIN WORK========BEGIN WORK

While, as we saw in Section 1 - Height-Continuity (Page 165), getting the output for a given number is not very useful, it does allow to prove the very useful

THEOREM 6.8 The global graphs of *all* power functions go through two of the following four plot dots: $(+1, -1$, $(+1, +1$, $(-1, -1$, $(-1, +1$,

Proof. There are four cases:

- For the power functions of type

$$x \xrightarrow{f} f(x) = +1\ {+1}^{+5}$$

we have

$$x \xrightarrow{f} f(x) = +1\ {+1}^{+5}$$
$$= +1 \odot +1 \odot +1 \odot +1 \odot +1 \odot +1$$

- The other three cases are left to the reader

□

1. Plot dot. Let f be the *regular* power function given by the global input-output rule

$$\underbrace{x}_{\text{input}} \xrightarrow{f} f(x) = \underbrace{ax^{\pm n}}_{\text{output-specifying code}}$$

where n is the number of copies used by f, and let x_0 be the *given input*. To plot the input-output pair for the given input x_0, we use **??** on **??** which, in the case of regular power functions, becomes

PROCEDURE 6.4

1. *To get* the output at the given input using **??** on **??** to get the input-output pair,
2. *Locate* the plot dot with **??** on **??**.

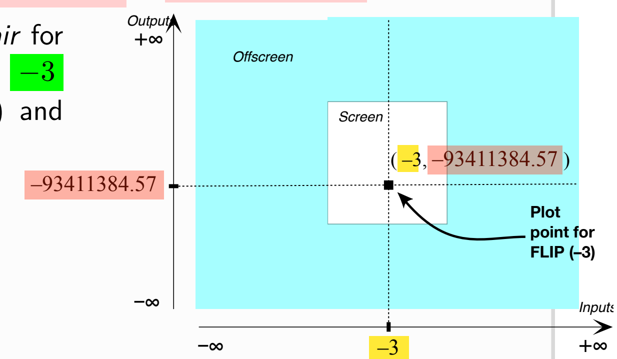
DEMO 6.10 Let $FLIP$ be the function given by the global input-output rule

$$x \xrightarrow{FLIP} FLIP(x) = (+527.31)x^{+11}$$

To *plot* the input-output pair for the input -3:

1. We get the output of the function $FLIP$ at -3. We found in **EXAMPLE 5.1** above that $FLIP(-3) = -93\,411\,384.57$

2. Thus, the *input-output pair* for the plot dot of $FLIP$ at -3 is $(-3, -93\,411\,384.57)$ and the plot dot is:



DEMO 6.11 Let $FLOP$ be the function given by the global input-output rule

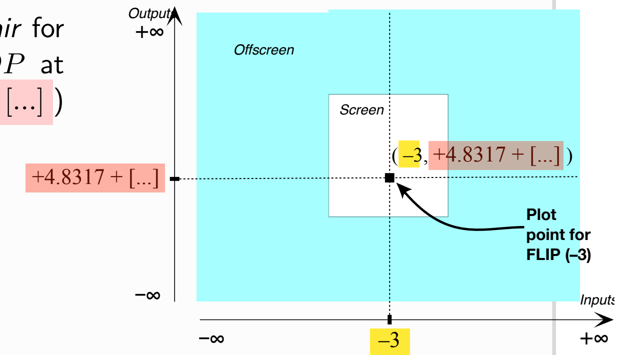
$$x \xrightarrow{FLOP} FLOP(x) = (+3\,522.38)x^{-6}$$

To *plot* the input-output pair for the input -3:

code

1. We get the output of the function $FLOP$ at -3. We found in **??** on ?? that $FLOP(-3) = +4.8317 + [...]$

2. Thus, the *input-output pair* for the graph point of $FLOP$ at -3 is $(-3, +4.8317 + [...])$ and the plot dot is:



2. Thickening the plot dot. As mentioned in **??** on **??**, instead of using *single inputs* to get *single plot dots*, we will "thicken the plot" that is we will use *neighborhoods* of given inputs to get *graph places*. But to use *neighborhoods* with *global input-output rules*, we will first have to introduce **code** to be able to *declare* by what to replace x. And, since this at the very core of what we will be doing in the rest of this text, we want to proceed with the utmost caution.

1. Since we are dealing here with *regular power functions* we will only be interested in inputs *near* ∞ and/or inputs *near* 0 and so here all we will need is the *sign-size*.

In order to declare by what we want to replace x, we will use the following code:

Near	Side				Code
Infinity	Left	0 ------ ∞	*positive*	$+\infty$	$+large$
	Right	∞ ------ 0	*negative*	$-\infty$	$-large$
Zero	Left	∞ ------ 0	*negative*	0^-	$-small$
	Right	0 ------ ∞	*positive*	0^+	$+small$

2. For the input-output pairs on *one* side, we will basically use **??** on **??** but *declare* that x is to be replaced using the above code for the given input.

3. For the input-output pairs of *both* sides, we will use the as follows:

Instead of	We can just write	and the I-O pair
$+ \longrightarrow +$ and $- \longrightarrow +$	$\pm \longrightarrow +$	$(\pm, +)$
$+ \longrightarrow -$ and $- \longrightarrow -$	$\pm \longrightarrow -$	$(\pm, -)$
$+ \longrightarrow +$ and $- \longrightarrow -$	$\pm \longrightarrow \pm$	$(\pm, \pm)$
$+ \longrightarrow -$ and $- \longrightarrow +$	$\pm \longrightarrow \mp$	$(\pm, \mp)$

3. Graph box near ∞ and near 0.

Once we have the input-output pairs near ∞ and near 0, we get the graph places as in **??** **??** on **??**. Here again,

i. In the first four demos, DEMO 6.12 on page 290, DEMO 6.13 on page 291, DEMO 6.14 on page 291, DEMO 6.15 on page 292, we will deal with only one side or the other.

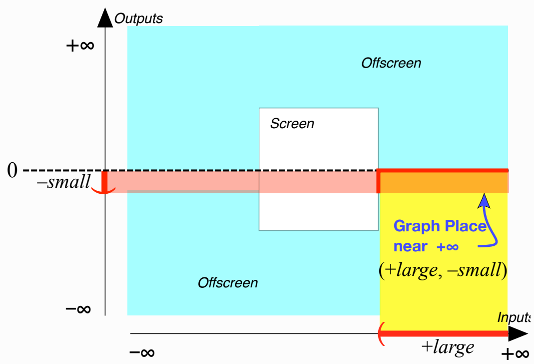
ii. In the next four demos, DEMO 6.16 on page 292, DEMO 6.17 on page 293, DEMO 6.18 on page 294, DEMO 6.19 on page 294, we will deal with both sides at the same time.

PROCEDURE 6.5

1. Get the input-output pairs using **??** **??** on **??** or **??** **??** on **??**.
2. *Locate* the *graph place* using **??** **??** on **??**.

DEMO 6.12

1. We get that the *input-output pairs* for $NADE$ near $+\infty$ are $(+large, -small)$ (See DEMO 6.2 on page 268)
2. The *graph place* of $NADE$ near $+\infty$ then is:



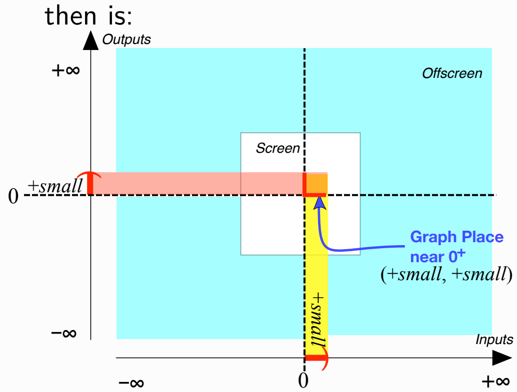
DEMO 6.13 Let $MADE$ be the function given by the global input-output rule

$$x \xrightarrow{MADE} MADE(x) = (+27.61)x^{+5}$$

To *locate* the *graph place* of $MADE$ near 0^+:

1. We get that the *input-output pairs* for $MADE$ near 0^+ are [$+small$, $+small$](See DEMO 6.6 on page 273)

2. The *graph place* of $MADE$ near 0^+ then is:



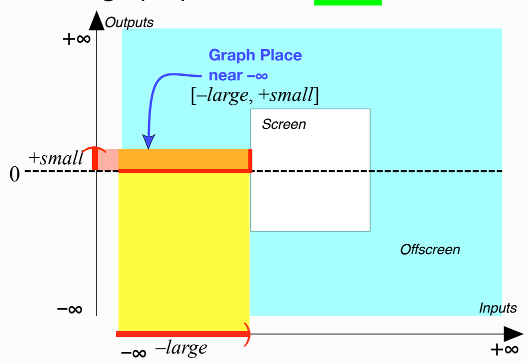
DEMO 6.14 Let $RADE$ be the function given by the global input-output rule

$$x \xrightarrow{RADE} RADE(x) = (+45.67)x^{-4}$$

To *locate* the *graph place* of $RADE$ near $-\infty$:

1. We get that the *input-output pairs* for $RADE$ near $-\infty$ are $[-large, +small]$(See DEMO 6.3 on page 269)

2. The *graph place* near $-\infty$ then is:



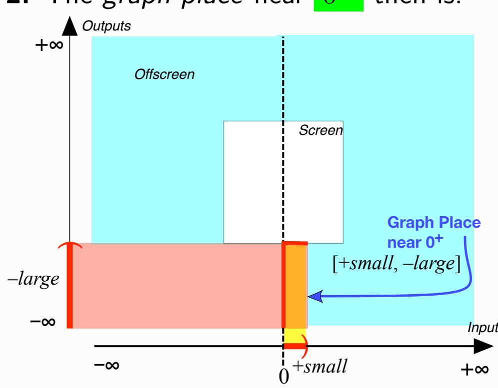
DEMO 6.15 Let $WADE$ be the function given by the global input-output rule

$$x \xrightarrow{WADE} WADE(x) = (-28.34)x^{-3}$$

To *locate* the *graph place* of $WADE$ near 0^+:

1. We get that the *input-output pairs* for $WADE$ near 0^+ are $[+small, -large]$(See DEMO 6.15 on page 292)

2. The *graph place* near 0^+ then is:



DEMO 6.16 Let $PADE$ be the function given by the global input-

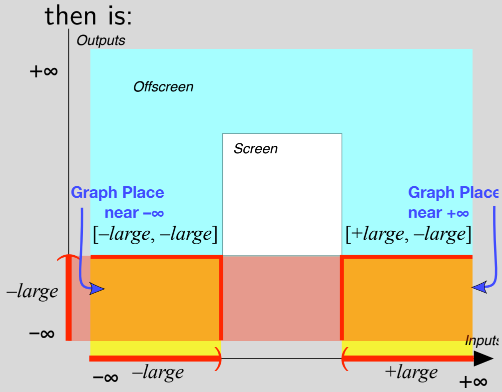
output rule

$$x \xrightarrow{PADE} PADE(x) = (-65.18)x^{+6}$$

To *locate* the *graph place* of $PADE$ near ∞.

1. We get that the *input-output pairs* for $PADE$ near ∞ are [$\pm large$, $-large$](See DEMO 6.5 on page 272)

2. The *graph place* of $PADE$ near ∞
then is:



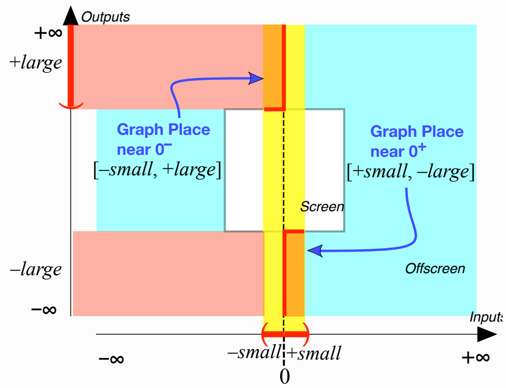
DEMO 6.17 Let $JADE$ be the function given by the global input-output rule

$$x \xrightarrow{JADE} JADE(x) = (-65.71)x^{-5}$$

To *locate* the *graph place* of $JADE$ near 0:

1. We get that the *input-output pairs* for $JADE$ near 0 are [$\pm small$, $\mp large$] (See DEMO 6.8 on page 276)

2. The *graph place* of $JADE$ near 0
then is:



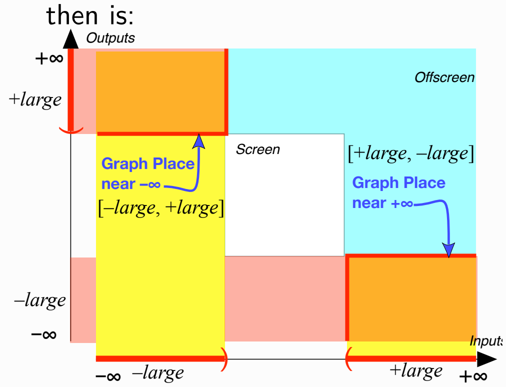
Demo 6.18 Let $DADE$ be the function given by the global input-output rule

$$x \xrightarrow{DADE} DADE(x) = (-83.91)x^{+5}$$

To *locate* the *graph place* of $DADE$ near ∞:

1. We get that the *input-output pairs* for $DADE$ near ∞ are $[\pm large, \mp large]$ (See DEMO 6.4 on page 271)

2. The *graph place* of $DADE$ near ∞ then is:



Demo 6.19 Let $FADE$ be the function given by the global input-output rule

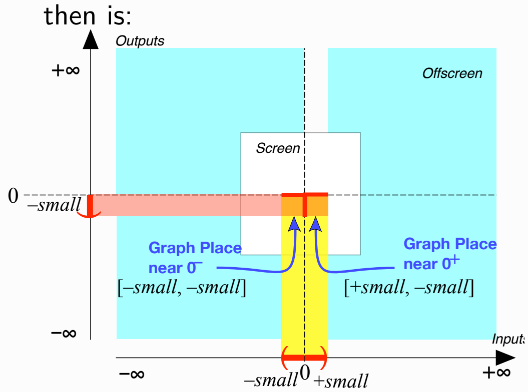
$$x \xrightarrow{FADE} FADE(x) = (-65.18)x^{+6}$$

To *locate* the *graph place* of $FADE$ near 0^-.

shape forced

1. We get that the *input-output pairs* for $FADE$ near 0 are [$\pm small$, $-small$] (See DEMO 6.9 on page 277)

2. The *graph place* of $FADE$ near 0 then is:



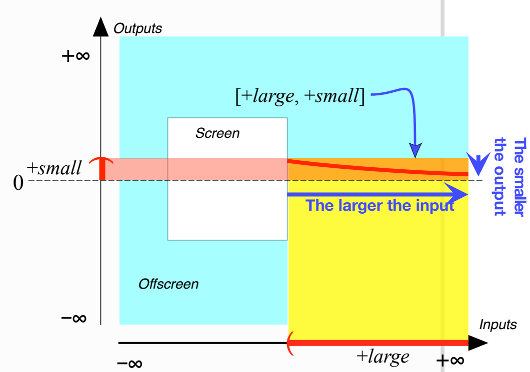
4. Local Graph Near ∞ and Near 0. Regular *power functions* are very nice in that the **shapes** of the local graphs near ∞ and near 0 are **forced** by the *graph place*. In other words, once we know the *graph place*, there is only one way we can draw the *local graph* because:

i. The smaller or the larger the input is, the smaller or the larger the output will be,

ii. The local graph cannot escape from the place.

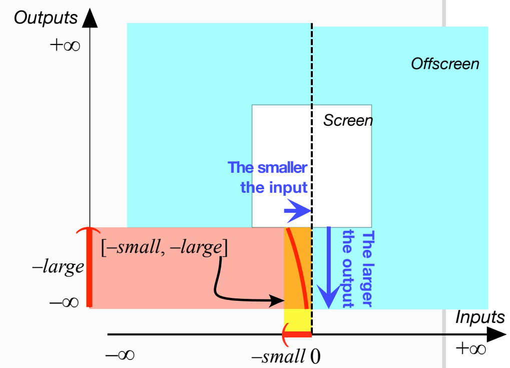
DEMO 6.20 Given a power function for which the place of a local graph is [$+large$, $+small$], we get the *shape* of the local graph as follows

i. The *slope* is forced by the fact that the *larger* the input is, the *smaller* the output will be.
ii. The *concavity* is forced by the fact that the local graph cannot cross the 0-output level line.



Demo 6.21 Given a power function for which the place of a local graph is [$-small$, $-large$], we get the *shape* of the local graph as follows

i. The *slope* is forced by the fact that
the *smaller* the input is,
the *larger* the output will be.
ii. The *concavity* is forced by the fact that the local graph cannot cross the 0-input level line.



5. Local Features Near ∞ and Near 0.

1. Given a regular power function being given by a global input-output rule, to get the *Height sign* near ∞ or near 0, we need only compute the sign of the outputs for nearby inputs with the global input-output rule.

Demo 6.22 Let JOE be the function given by the global input-output rule

$$x \xrightarrow{JOE} JOE(x) = (-65.18)x^{+6}$$

To get the *Height sign* of JOE near 0^+

We ignore the *size* and just look at the *sign*:

$$\begin{aligned} + \xrightarrow{JOE} JOE(+) &= (-)(+)^{+6} \\ &= (-)\cdot(+) \\ &= - \end{aligned}$$

and

$$\begin{aligned} - \xrightarrow{JOE} JOE(-) &= (-)(-)^{+6} \\ &= (-)\cdot(+) \\ &= - \end{aligned}$$

So, Height sign JOE near 0 is $\langle -,- \rangle$

2. Given a regular power function being given by a global input-output rule, to get the *Slope sign* or the *Concavity sign* near ∞ or near 0, we need the *local graph* near ∞ or near 0.

DEMO 6.23 Let $JILL$ be the function given by the global input-output rule

$$x \xrightarrow{JILL} JILL(\pm) = (+32.06)(\pm)^{+6}$$

To *get* the *Slope sign* of $JILL$ near 0

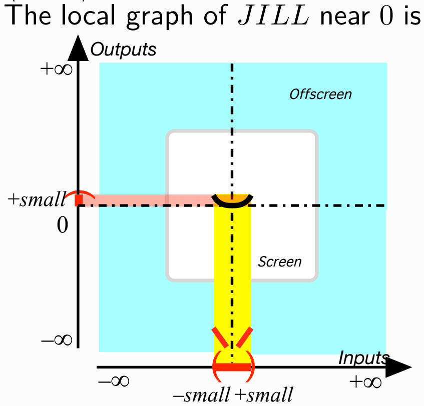
We need the *local graph* of $JILL$ near 0.

i. We get the output for $JILL$ near 0

$$\begin{aligned}\pm small \xrightarrow{JILL} JILL(\pm small) &= (+mediumsize)(\pm small)^{+even} \\ &= (+mediumsize)(\pm)^{even}(small)^{+} \\ &= (+mediumsize)(+)\cdot(small) \\ &= +small\end{aligned}$$

iii. Slope sign $JILL$ near $0 = \langle \diagdown, \diagup \rangle$

ii. The local graph of $JILL$ near 0 is



DEMO 6.24 Let JIM be the function given by the global input-output rule

$$x \xrightarrow{JIM} JIM(x) = (-72.49)x^{-5}$$

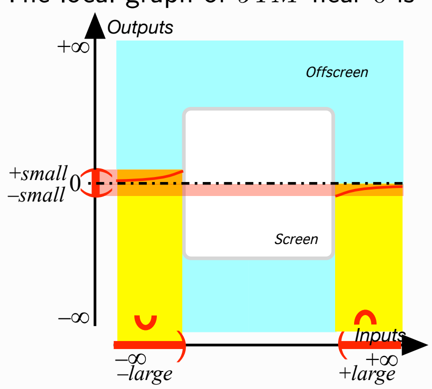
To *get* the *Concavity sign* of JIM near ∞

We need the *local graph* of JIM near ∞.
i. We get the output for JIM near ∞

$$\pm large \xrightarrow{JIM} JIM(\pm large)$$
$$= (-mediumsize)(\pm large)^{-odd}$$
$$= \frac{-mediumsize}{\underbrace{(\pm large)...(\pm large)}_{\text{odd number of copies}}}$$
$$= \frac{-mediumsize}{\pm large}$$
$$= -mediumsize \cdot \pm small$$
$$= \mp small$$

iii. Concavity sign JIM near ∞ = $\langle \cap, \cup \rangle$

ii. The local graph of JIM near 0 is



The Global Analysis of regular monomial functions is very *systematic* because the global input-output rule is very simple.

7 Reciprocity

1. Another way to look at **??** on **??** is to realize that, for a monomial function,

- If Output Size = Input Size, this can only be because Exponent Sign = +,

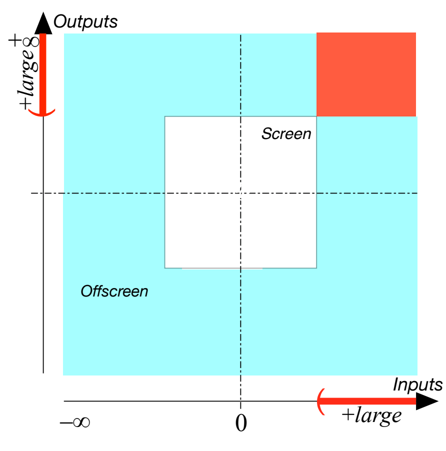
- If Output Size = *Reciprocal* Input Size, this can only be because Exponent Sign = −.

Which gives us the following which we will use to graph regular monomial functions efficiently:

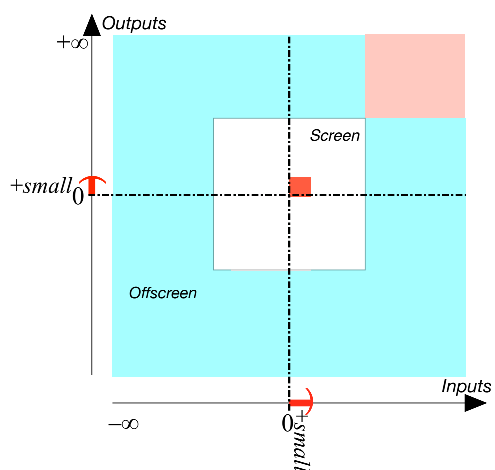
THEOREM 6.9 Reciprocity (For Regular Monomial Functions.)
- If *large* → *large*, then *small* → *small* (And vice versa.)
- If *large* → *small*, then *small* → *large* (And vice versa.)

EXAMPLE 6.17.

After we have found, for instance,

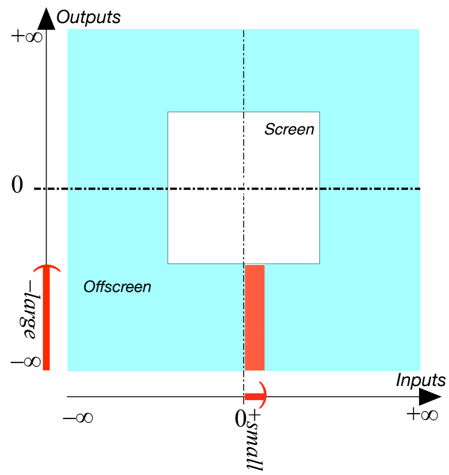


We get from THEOREM 6.9

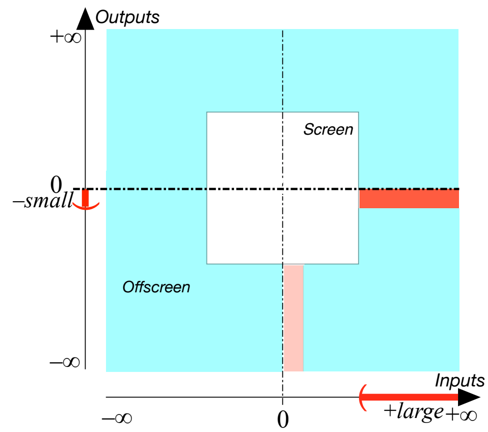


EXAMPLE 6.18.

After we have found, for instance,

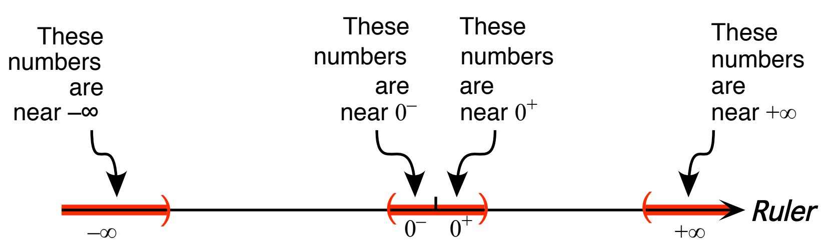


We get from THEOREM 6.9



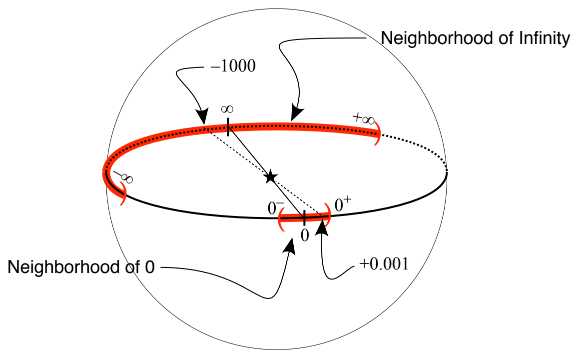
2. The relationship between ∞ and 0 is not only important but also fascinating.

a. Even though, as an input, 0 is usually not particularly important, there is an intriguing "symmetry" between ∞ and 0 namely:



More precisley, *small numbers* are some sort of inverted image of *large numbers* since the *reciprocal* of a *large number* is a *small number* and vice versa.

EXAMPLE 6.19. **T** he opposite of the reciprocal of -0.001 is $+1\,000$. In a Magellan view, we have

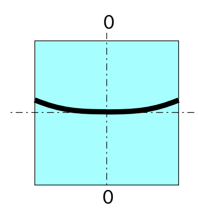


b. Here is yet another way to look at reciprocity. We start with the graph of a monomial function and we "turn" it so as to see it while facing ∞ and we then compare it with the graph near 0 of the reciprocal function.

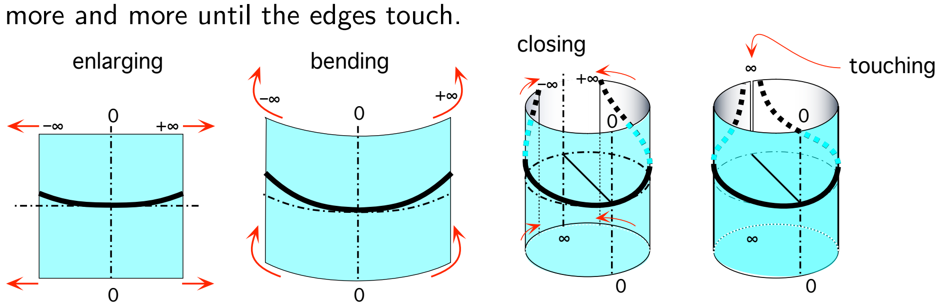
EXAMPLE 6.20. L et the monomial function specified by the global input-output rule

$$x \xrightarrow{RAIN} RAIN(x) = (+1)x^{+4}$$

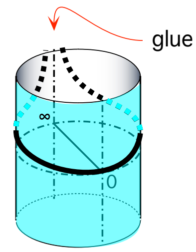
the local graph near 0 of $RAIN$ is: ¥



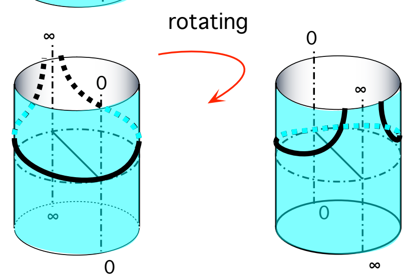
We enlarge the *extent* of the input ruler more and more while shrinking the *scale* by the edges more and more and, as we do so, we bend the screen backward more and more until the edges touch.



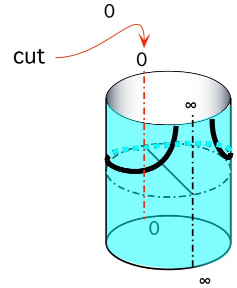
We then glue shut the edges of the screen at ∞ to get a cylinder. ¥



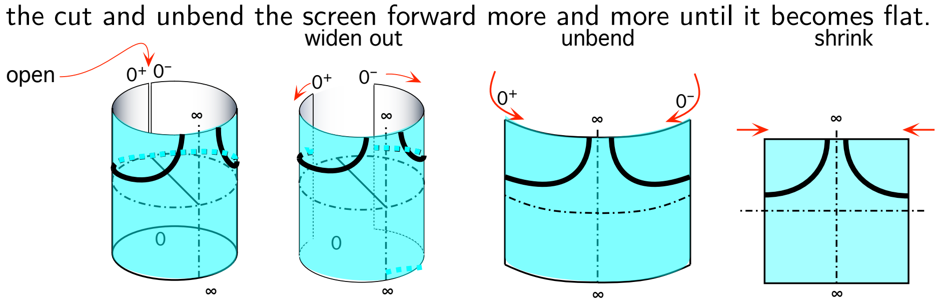
Then we turn the cylinder half a turn so that ∞ gets to be in front of us: ¥



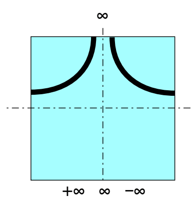
Now we cut open the cylinder along the input level line for 0 ¥



Finally we widen the cut and unbend the screen forward more and more until it becomes flat.

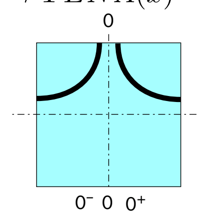


The local graph near ∞ that we got for $RAIN$ is:



¥

It is the same as the local graph near 0 of the reciprocal function specified by the global input-output rule $x \xrightarrow{TENA} TENA(x) = (+1)x^{-4}$



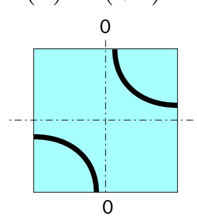
¥

(Keep in mind that the left side of ∞ is the positive side of ∞ and the right side of ∞ is the negative side of ∞. So the graphs on the positive sides are the same and the local graphs on the negative sides are also the same.)

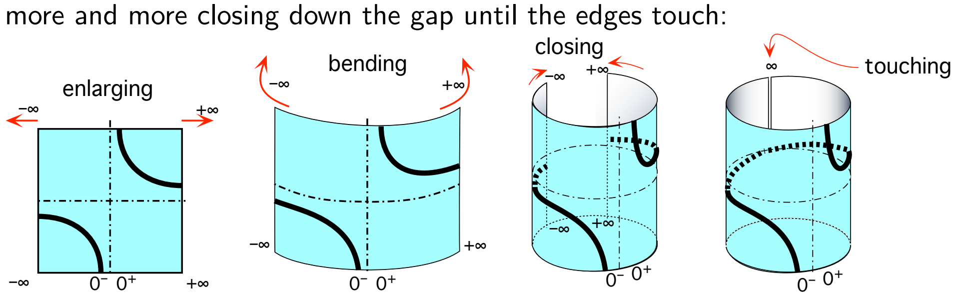
EXAMPLE 6.21. Given the monomial function specified by the global input-output rule

$$x \xrightarrow{MIKE} MIKE(x) = (+1)x^{-3}$$

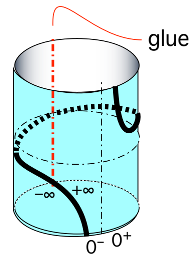
the local graph near 0 of $MIKE$ is: ¥



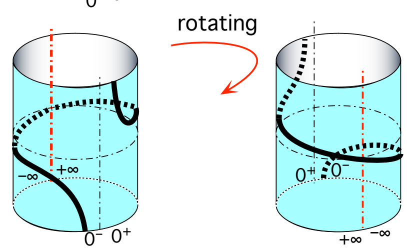
We enlarge the *extent* of the input ruler more and more while shrinking the *scale* by the edges more and more and, as we do so, we bend the screen backward more and more closing down the gap until the edges touch:



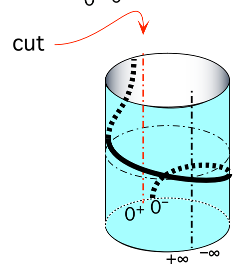
We then glue shut the edges of the screen at ∞ to get a cylinder. ¥



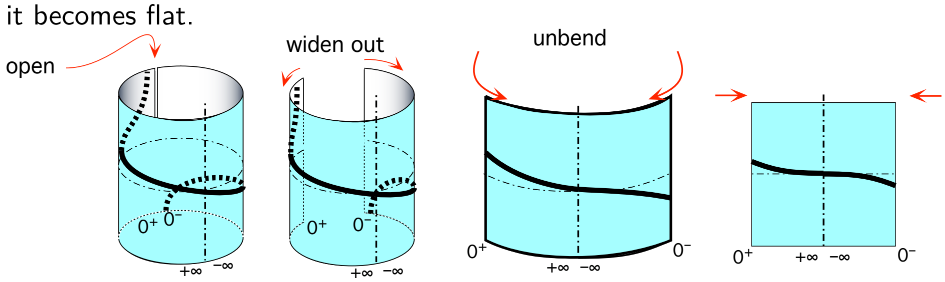
Then we rotate the cylinder half a turn so that ∞ gets to be in front of us: ¥



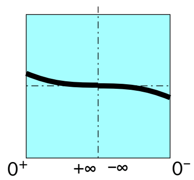
Now we cut open the cylinder along the input level line for 0 ¥



Finally we widen the cut and unbend the screen forward more and more until it becomes flat.

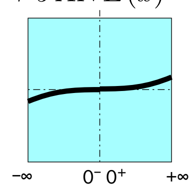


The local graph near ∞ that we just got for $MIKE$ is:



¥

It is the same as the local graph near 0 of the reciprocal function specified by the global input-output rule $x \xrightarrow{JANE} JANE(x) = (+1)x^{+3}$



¥

(Keep in mind that the left side of ∞ is the positive side of ∞ and the right side of ∞ is the negative side of ∞. So the graphs on the positive sides *are* the same and the local graphs on the negative sides also *are* the same.)

8 Global Graphing

We can of course get the global graph the way we will get the global graph of all the other functions in this text, that is as described in **??**, but, in the case of *regular monomial functions*, we will be taking advantage of the following TTHEOREMs which we must become *completely familiar* with—but which we certainly must not *memorize*:

- The first part of **??** on **??** namely:

> **THEOREM 6.10 Output Sign for *positive* inputs.** (For Regular Monomial Functions.)
> Output Sign for *positive* inputs = Coefficient Sign.

- **??** on **??**
- THEOREM 6.9 on page 299
- THEOREM 6.6 on page 286

Then, after a little bit of practice, we will be able to get the global graph *very rapidly*:

PROCEDURE 6.6 To graph a *regular* monomial function:

a. Locate the *graph place* for inputs near $+\infty$ as follows:

i. Determine if the graph place for inputs near $+\infty$ is *above* or

below the 0-output level line.
(Use THEOREM 6.10 on ??)

ii. Determine if the graph place for inputs near $+\infty$ is near the 0-*output* level line or near the ∞-*output* level line,
(Use ?? on ??)

b. Locate the *graph place* for inputs near 0^+.
(Use THEOREM 6.9 on page 299).

c. Locate the *graph places* for inputs near $-\infty$ and inputs near 0^-.
(Use THEOREM 6.6 on page 286)

d. Draw the *global graph* through the *graph places*.

DEMO 6.25

1. We locate the graph place for inputs near $+\infty$:

i. Since Coefficient Sign $= +$,

$$+ \xrightarrow{KIR} +$$

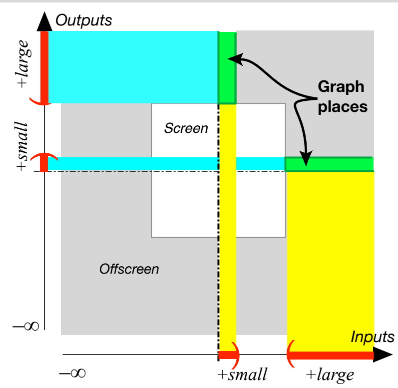
(Using THEOREM 6.10 on page 305.)

ii. Since Exponent Sign $= -$,

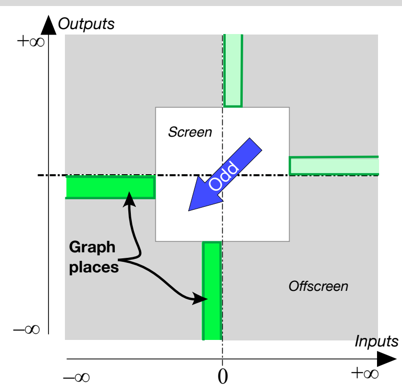
$$large \xrightarrow{KIR} small$$

(Using ?? on ??.)

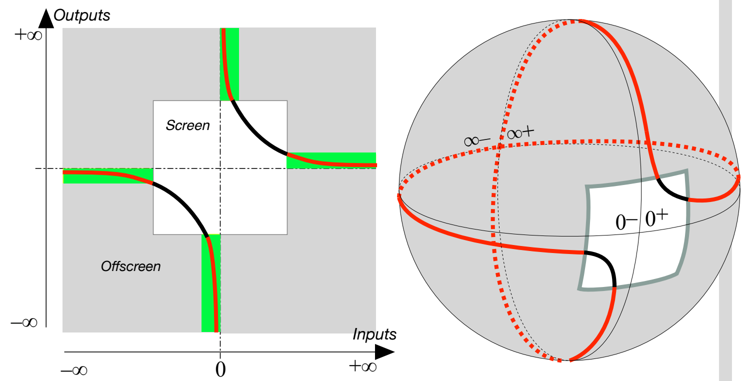
2. We locate the graph place for inputs near 0^+.
(Using THEOREM 6.9 on page 299.)



3. We locate the graph places for inputs near $-\infty$ and near 0^-.
(Using THEOREM 6.6 on page 286.)



4. We draw the *global graph* through the graph places.
And, to the right is a Magellan view of the global graph.



DEMO 6.26

1. We locate the graph place for inputs near $+\infty$:

i. Since Coefficient Sign $= -$,

$$+ \xrightarrow{KIM} +$$

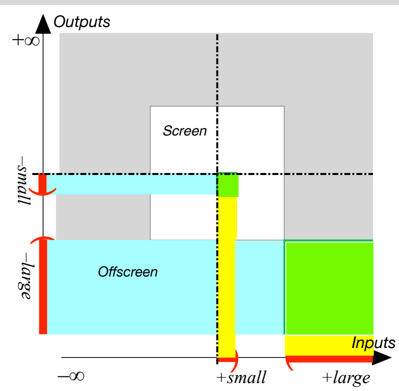
(Using THEOREM 6.10 on page 305.)

ii. Since Exponent Sign $= +$,

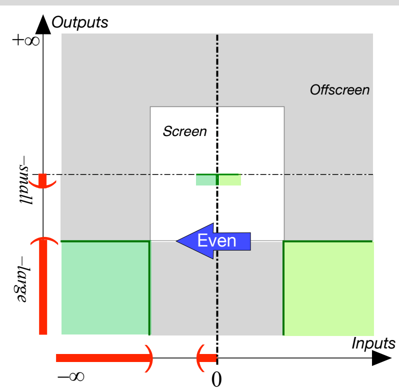
$$large \xrightarrow{KIM} large$$

(Using ?? on ??.)

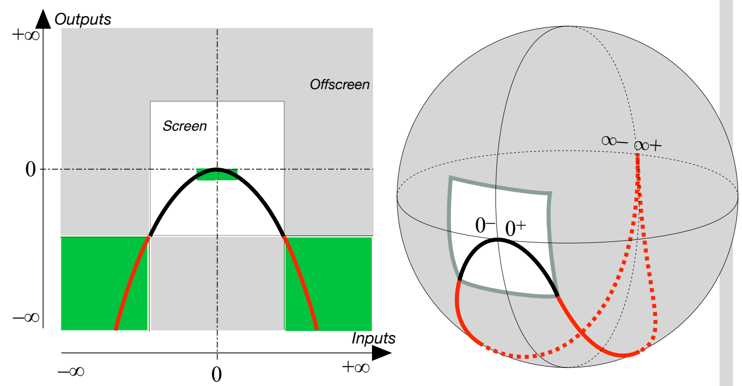
2. We locate the graph place for inputs near 0^+.
(Using THEOREM 6.9 on page 299.)



3. We locate the graph places for inputs near $-\infty$ and near 0^-.
(Using THEOREM 6.6 on page 286.)



4. We draw the *global graph* through the graph places.
And, to the right is a Magellan view of the global graph.



DEMO 6.27

1. We locate the graph place for inputs near $+\infty$:

i. Since Coefficient Sign $= -$,

$$+ \xrightarrow{KIN} -$$

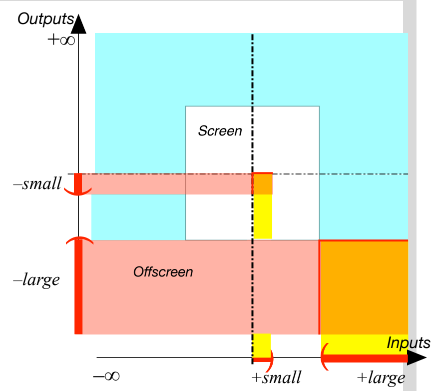
(Using THEOREM 6.10 on page 305.)

ii. Since Exponent Sign $= +$,

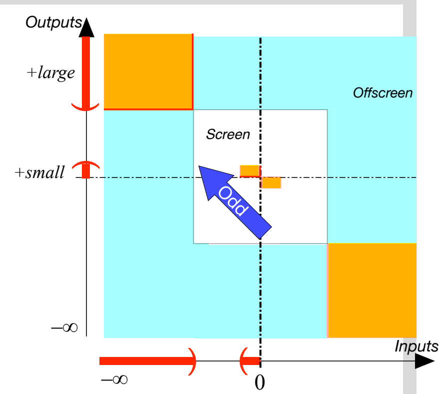
$$large \xrightarrow{KIN} large$$

(Using ?? on ??.)

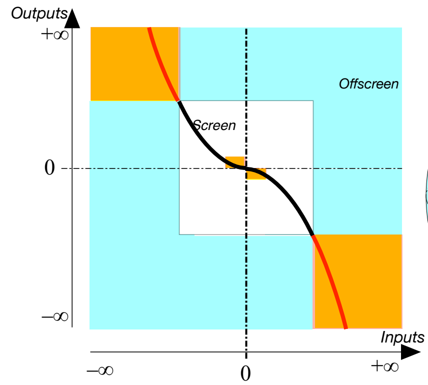
2. We locate the graph place for inputs near 0^+.
(Using THEOREM 6.9 on page 299.)

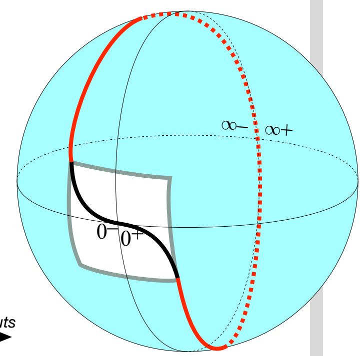


3. We locate the graph places for inputs near $-\infty$ and near 0^-.
(Using THEOREM 6.6 on page 286.)



4. We draw the *global graph* through the graph places.
And, to the right is a Magellan view of the global graph.





DEMO 6.28

1. We locate the graph place for inputs near $+\infty$:

i. Since Coefficient Sign $= +$,

$$+ \xrightarrow{KIB} +$$

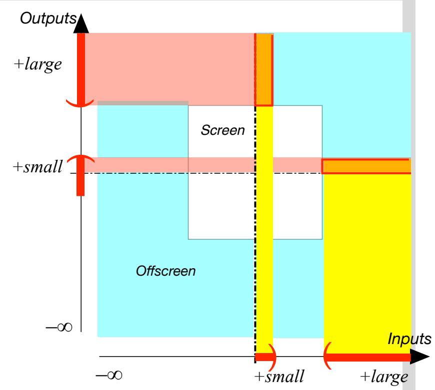
(Using THEOREM 6.10 on page 305.)

ii. Since Exponent Sign $= -$,

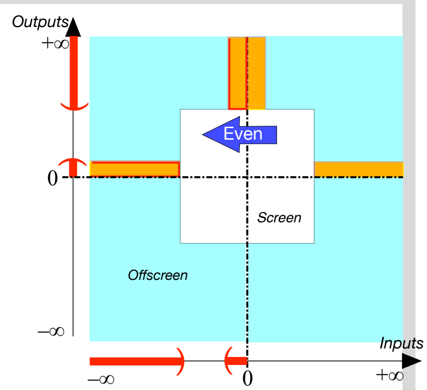
$$large \xrightarrow{KIB} small$$

(Using ?? on ??.)

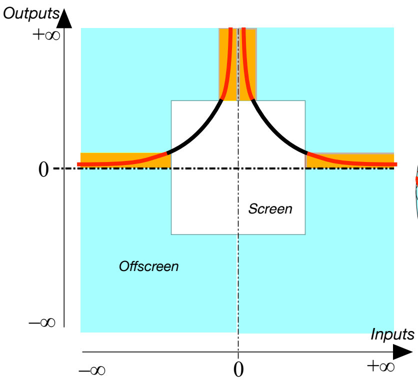
2. We locate the graph place for inputs near 0^+.
(Using THEOREM 6.9 on page 299.)

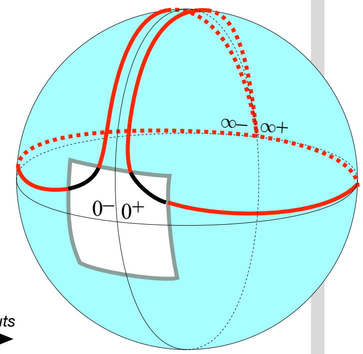


3. We locate the graph places for inputs near $-\infty$ and near 0^-.
(Using THEOREM 6.6 on page 286.)



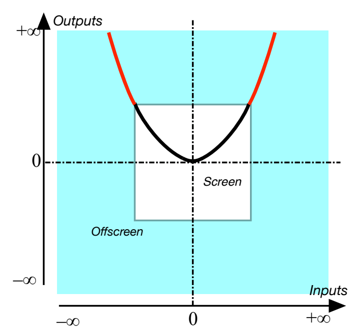
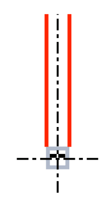
4. We draw the *global graph* through the graph places.
And, to the right is a Magellan view of the global graph.



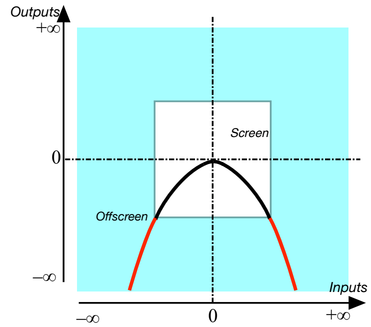
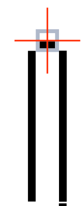
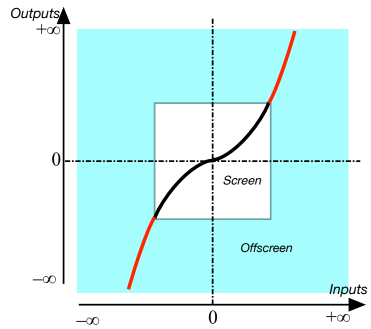
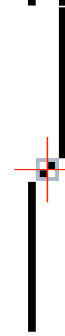
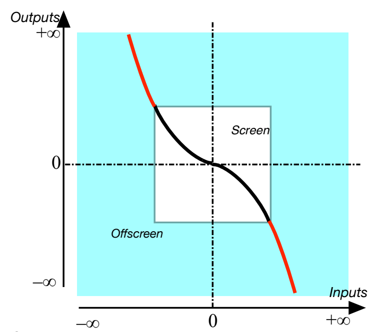
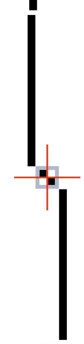
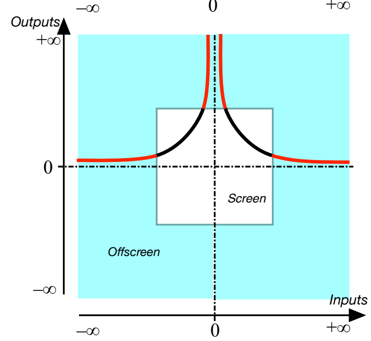
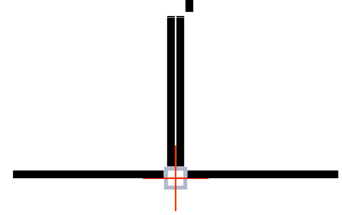


9 Types of Global Graphs

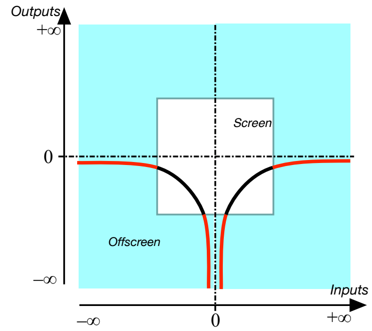
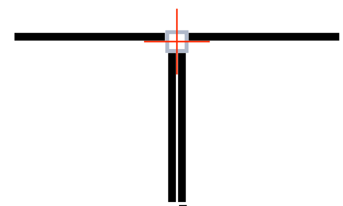
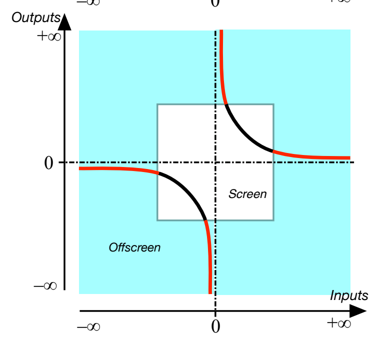
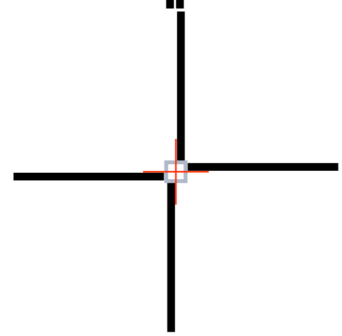
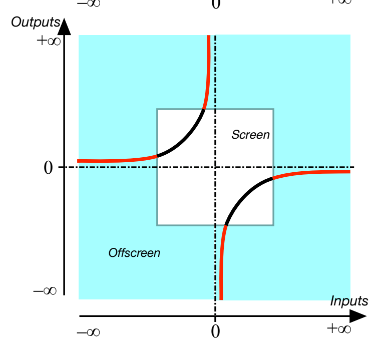
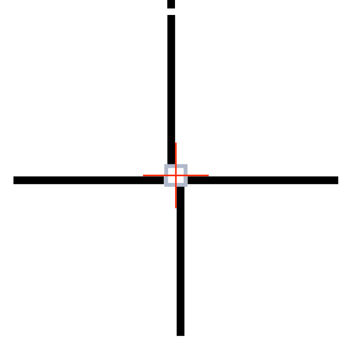
Each type of *global input-output rule* corresponds to a type of *global graph*. The global graphs are shown both from "close-up" to see the bounded graph and from "faraway" to see how the graphs flatten out.

Input-output rule	From "close-up"	From "faraway"
$x \xrightarrow{PEP} PEP(x) = (+1)x^{+even}$		

Continued on next page

<table>
<tr><th>Input-output rule</th><th>From “close-up”</th><th>From “faraway”</th></tr>
<tr><td>$x \xrightarrow{PEN} PEN(x) = (-1)x^{+even}$</td><td>
</td><td></td></tr>
<tr><td>$x \xrightarrow{POP} POP(x) = (+1)x^{+odd}$</td><td>
</td><td></td></tr>
<tr><td>$x \xrightarrow{PON} PON(x) = (-1)x^{+odd}$</td><td>
</td><td></td></tr>
<tr><td>$x \xrightarrow{NEP} NEP(x) = +x^{-even}$</td><td>
</td><td></td></tr>
</table>

Continued on next page

Input-output rule	From “close-up”	From “faraway”
$x \xrightarrow{NEN} NEN(x) = -x^{-even}$		
$x \xrightarrow{KIR} KIR(x) = +x^{-odd}$		
$x \xrightarrow{NON} NON(x) = -x^{-odd}$		

Chapter 7

No More Affine Functions

OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OK-soFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKso-FAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR - OKsoFAR

1 Binomial Functions

1.

2.

Example 7.1. **L** et $BASE$ be given by the global input-ouput rule

$$x \xrightarrow{BASE} BASE(x) = (-3)x^{+2}$$

and let $ADD\text{-}ON$ be given by the global input-ouput rule

$$x \xrightarrow{ADD\text{-}ON} ADD\text{-}ON(x) = (+5)x^{0}$$

$$= +5$$

then the SUM function is given by the global input-ouput rule

$$x \xrightarrow{SUM} SUM(x) = (-3)x^{+2} \oplus (+5)x^0$$
$$= (-3)x^{+2} + 5$$

To see that SUM *cannot* be replaced by a single monomial function, we first evaluate all three functions at some input, for instance $+2$:

$$+2 \xrightarrow{BASE} BASE(+2) = (-3)(+2)^{+2}$$
$$= -12$$

and

$$+2 \xrightarrow{ADD\text{-}ON} ADD\text{-}ON(+2 = (+5)(+2)^0$$
$$= +5$$

then

$$x \xrightarrow{SUM} SUM(x) = (-3)(+2)^{+2} \oplus (+5)(+2)^0$$
$$= -12 \oplus +5$$
$$= -7$$

The question then is what *monomial function* could return the output -7 for the input $+2$.

Of course, we can easily find a monomial function that would return the output -7 for the input $+2$. For instance, the dilation function $x \xrightarrow{f} f(x) = -\frac{7}{2}x$ does return the output -7 for the input $+2$. But$f$ is *not* going to return the same output as SUM for other inputs, say, $+3$, $+4$, etc which it should. So, the *binomial function*

$$x \xrightarrow{SUM} SUM(x) = (-3)x^{+2} + 5$$

cannot be replaced by the single *monomial function*

$$x \xrightarrow{f} f(x) = -\frac{7}{2}x$$

Cautionary note 7.1 e noted at the beginning of Chapter 5 that monomial functions were only rarely called *monomial functions* and that this was unfortunate: indeed, it would be nicer to say that a *binomial* function cannot be replaced by a single *monomial* function. (We cannot have two for the price of one.)

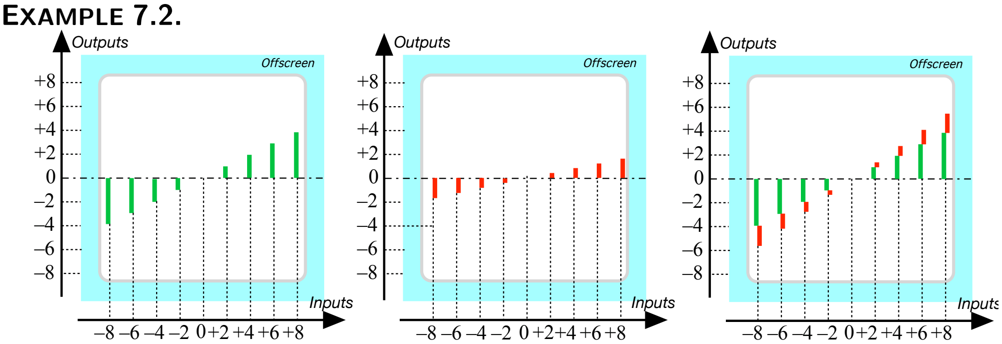
2 Graphs of Binomial Functions

1. When the exponent of the *add-on* function is the same as the exponent of the *base* function, the bar graphs show exactly why the *sum* function will have again the same exponent.

a. Given a constant base function, adding-on a constant function:

b. Given a dilation base function, adding-on a dilation function:

EXAMPLE 7.2.

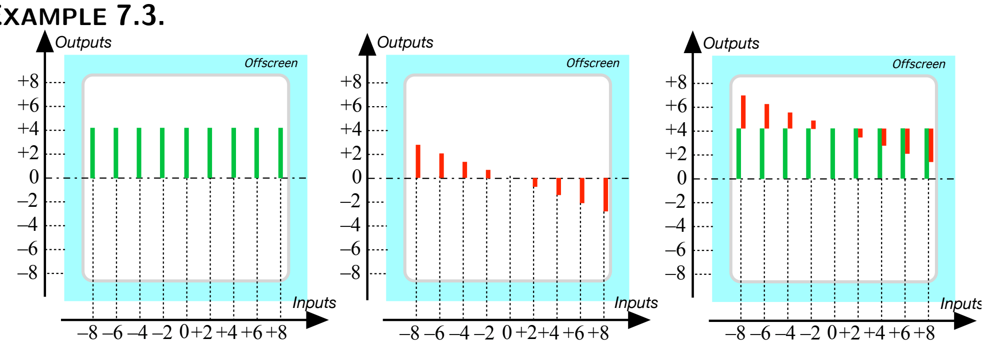


2. When the exponent of the add-on function is *not* the same as the exponent of the base function, the bar graphs show clearly why the *sum* function cannot be a monomial function.

a. Given a constant base function,

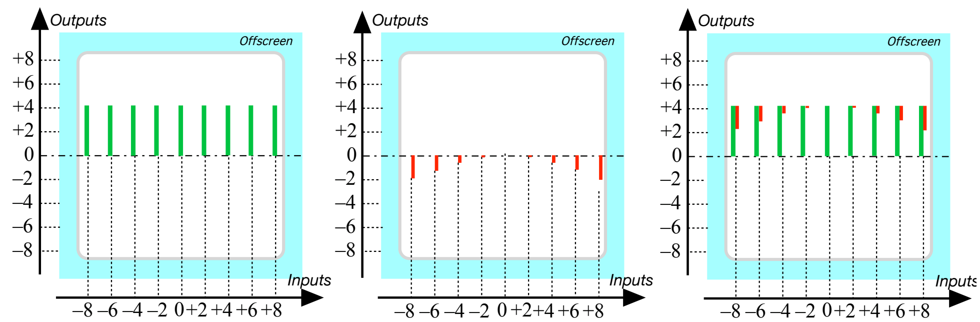
- Adding-on a dilation function:

EXAMPLE 7.3.



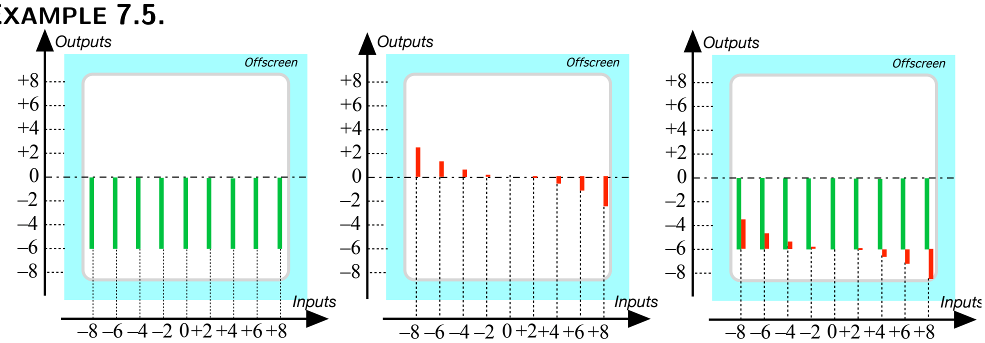
- Adding-on an even positive exponent monomial function:

EXAMPLE 7.4.



- Adding-on an odd positive exponent monomial function:

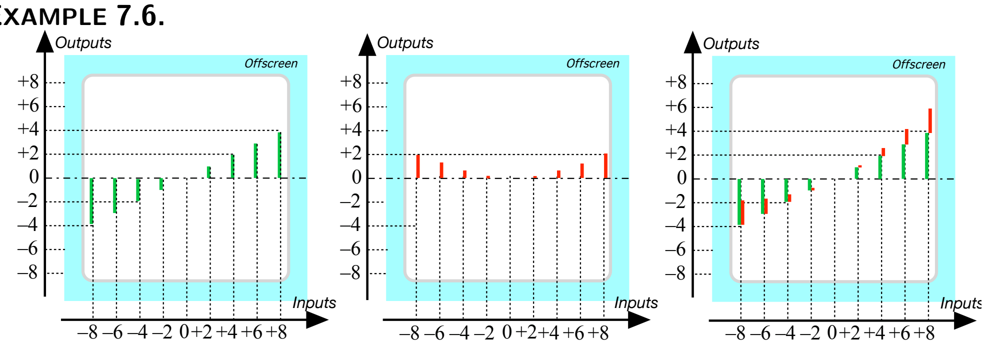
Example 7.5.



b. Given a dilation base function,

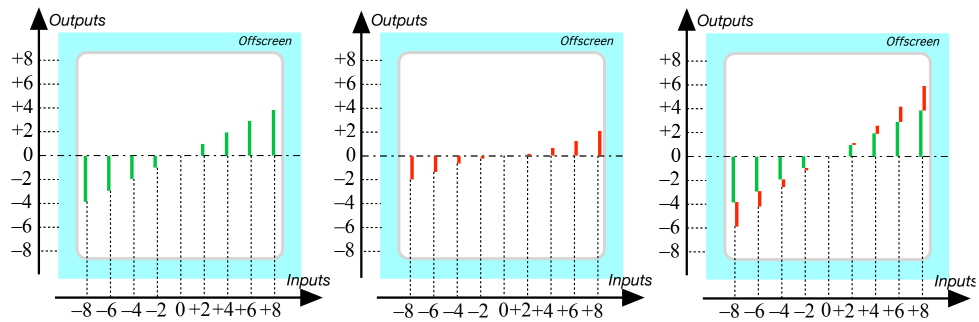
- Adding-on an even monomial function:

Example 7.6.



- Adding-on an odd monomial function:

Example 7.7.



=======End LOOK UP =======

3 Local graphs

Just as we get a *plot point at* a *bounded* input from the *output at* that input, we get the *local graph near* any input, be it *bounded* or *infinity*, from the *jet near* that input.

Procedure 7.1

1. Get the jet near ∞ using PROCEDURE 5.8 To evaluate near ∞ the function given by $x \xrightarrow{AFFINE} AFFINE(x) = ax + b$ on page 247

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = \big[a\big]\, x + \big[b\big]$$

2. Get the local graph near ∞ of each term:

a. Get the graph of the *linear term* near ∞ by graphing near ∞ the monomial function $x \to ax$ using ?? ?? on ??.

b. Get the graph of the *constant term* near ∞ by graphing near ∞ the monomial function $x \to b$ using ?? ?? on ??.

3. Get the local graph near ∞ of $AFFINE$ by adding-on the constant term to the linear term using ??.

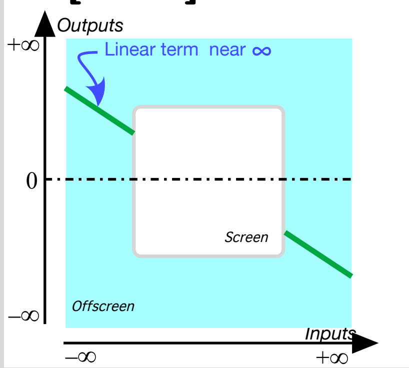
Demo 7.1

1. We get the jet near ∞: (See DEMO 5.8 on page 248)

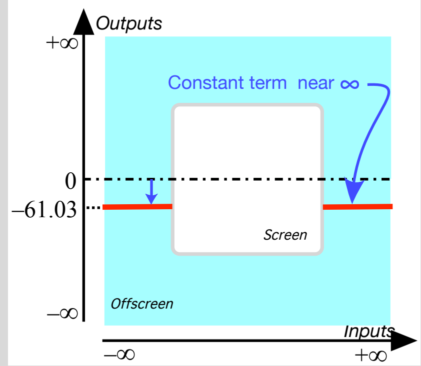
$$x \text{ near } \infty \xrightarrow{NINA} NINA(x) = \big[-82.47\big]x + \big[-61.03\big]$$

2. Get the local graph near ∞ of each term:

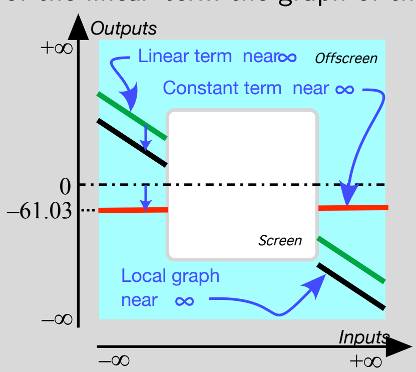
a. We get the graph of the linear term by graphing near ∞ the monomial function $x \to \Big[-82.47 \Big] x$ (See ?? on ??)



b. We get the graph of the constant term near ∞ by graphing near ∞ the monomial function $x \to \Big[-61.03 \Big]$ (See ?? on ??)



3. We get the local graph near ∞ of $NINA$ by adding-on to the graph of the *linear* term the graph of the *constant* term. (See ?? on ??)



PROCEDURE 7.2

i. Get the jet near x_0 of $AFFINE$ using PROCEDURE 5.9 To evaluate near x_0 the function given by $x \xrightarrow{AFFINE} AFFINE(x) = ax + b$ on page 251

ii. Get the graph of the constant term in the jet near x_0 namely of $\Big[ax_0 + b \Big]$

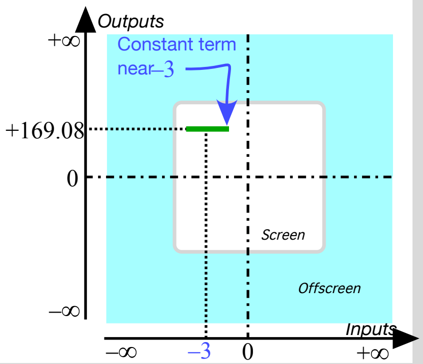
iii. Add-on the graph of the linear term in the jet near x_0 namely of $\Big[a \Big] h$

DEMO 7.2

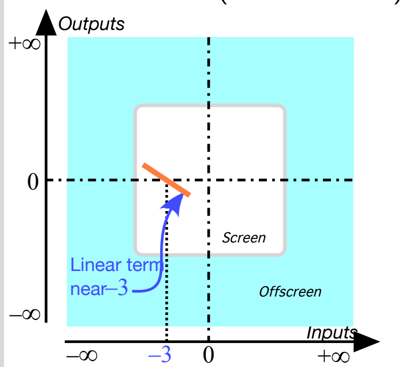
i. We get the jet near -3 of $ALDA$ by evaluating $ALDA$ near -3: (See DEMO 5.9 on page 252)

$$-3+h \xrightarrow{ALDA} ALDA(-3+h) = \underbrace{\Big[+169.08\Big] \oplus \Big[-32.67\Big]h}_{\text{output jet near } -3}$$

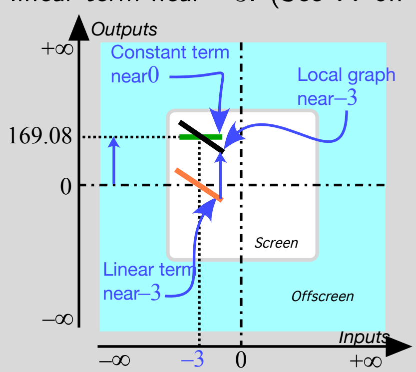
ii. We get the graph of the constant term near -3: (See ?? on ??)



iii. We get the graph of the linear term near -3 is: (See ?? on ??)



iv. We add-on the graph of the linear term near -3 to the graph of the linear term near -3. (See ?? on ??)



4 Local Feature-signs

As we saw in **?? ??**, a feature-sign near a given input, be it near ∞ or near x_0, can be read from the *local graph* and so all we need to do is:

i. Get the *output jet* from the global input-output rule. (See PROCE-

DURE 5.8 on page 247 when the given input is ∞ or PROCEDURE 5.9 on page 251 when the given input is x_0.)

ii. Get the *local graph* from the output jet. (See PROCEDURE 7.1 on page 317 when the given input is ∞ or PROCEDURE 7.1 on page 317 when the given input is x_0.)

iii. Get the *feature-sign* from the *local graph*(See **??**

However, with a little bit of reflection, it is faster and *much more useful* to read the feature-signs directly from the *jet* in the local input-output rule. But since, in order for the terms in the jet to be in *descending order of sizes*,

- In the case of *infinity*, the exponents of x have to be in *descending* order.
- In the case of a *bounded input*, the exponents of h have to be in *ascending* order.

we will deal with ∞ and with x_0 separately.

1. Near*infinity* things are quite straightforward:

PROCEDURE 7.3

i. Get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{AFFINE} AFFINE(x) = ax + b$$
$$= \underbrace{\Big[a\Big]x \oplus \Big[b\Big]}_{\text{output jet near } \infty}$$

ii. Then, in the *jet* near ∞:

- Get both the *Height-sign* and the *Slope-sign* from the *linear term* $\Big[a\Big]x$ because the next term $\Big[b\Big]$ is *too small to matter*.
- Since both the *linear term* and the *constant term* have no concavity, $AFFINE$ has no *Concavity-sign* near ∞.

DEMO 7.3 Get the Height-sign near ∞ of the function given by

$$x \xrightarrow{JULIE} JULIE(x) = -2x + 6$$

i. We get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{JULIE} JULIE(x) = -2x + 6$$
$$= \underbrace{\Big[-2\Big]x \oplus \Big[+6\Big]}_{\text{output jet near } \infty}$$

ii. We get *Height-sign* from the *linear* term $\Big[-2\Big]x$ because the *constant* term $\Big[+6\Big]$ is *too small to matter*.

Since the *linear coefficient* -2 is negative, we get that Height-sign

$JULIE$ near $\infty = \langle -, + \rangle$. (Seen from ∞.)

DEMO 7.4 Get the Slope-signs near ∞ of the function given by

$$x \xrightarrow{PETER} PETER(x) = +3x - 8$$

i. We get the local input-output rule near ∞:

$$\begin{aligned} x \text{ near } \infty \xrightarrow{PETER} PETER(x) &= +3x - 8 \\ &= \underbrace{\Big[+3\Big]x \oplus \Big[-8\Big]}_{\text{output jet near } \infty} \end{aligned}$$

ii. We get *Slope-sign* from the *linear* term $\Big[+3\Big]x$ because the *constant* term $\Big[-8\Big]$ is *too small to matter* (Not to mention that a *constant* term has *no* slope.)

Since the *linear coefficient* $+3$ is positive, we get that Slope-sign $PETER$ near $\infty = \langle \diagup, \diagup \rangle$. (Seen from ∞.)

2. In the case of a *bounded input*, things are a bit more complicated because the bounded input may turn out to be *ordinary* or *critical* for the *height*. But it will always be *ordinary* for the slope.

PROCEDURE 7.4

i. Get the local input-output rule near x_0:

$$\begin{aligned} x_0 + h \xrightarrow{AFFINE} AFFINE(x_0 + h) &= a(x_0 + h) + b \\ &= ax_0 + ah + b \\ &= ax_0 + b + ah \\ &= \underbrace{\Big[ax_0 + b\Big] \oplus \Big[a\Big]h}_{\text{output jet near } x_0} \end{aligned}$$

ii. Then, in the *jet* near x_0:

- If x_0 is *ordinary*, that is if $\Big[ax_0 + b\Big] \neq 0$, get the *Height-sign* from the *sign* of the constant term $\Big[ax_0 + b\Big]$ because the next term $\Big[a\Big]h$ is *too small to matter*. In other words, Height-sign $AFFINE$ near x_0 = Height-sign of the monomial function $h \to ax_0 + b$ near 0.
  But if x_0 is *critical*, that is if $\Big[ax_0 + b\Big] = 0$, the next term, namely the *linear term* $\Big[a\Big]h$, now does matter even though it is $small$. In other words, now Height-sign $AFFINE$ near x_0 = Height-sign of the monomial function $h \to ah$ near 0.

- Since the *constant term* has no slope, get the *Slope-sign* from the next smaller term in the jet, namely the *linear term*. In other words, Slope-sign $AFFINE$ near x_0 = Slope-sign of the monomial function $h \to ah$ near 0.
- Since both the *constant term* and the *linear term* have no concavity, $AFFINE$ has no *Concavity-sign* near x_0.

DEMO 7.5 Get the feature-signs near $+2$ of the function given by

$$x \xrightarrow{JULIE} JULIE(x) = -2x - 6$$

i. We get the local input-output rule near $+2$:

$$\begin{aligned} +2+h \xrightarrow{JULIE} JULIE(+2+h) &= -2(+2+h) - 6 \\ &= -2(+2) - 2h - 6 \\ &= -4 - 2h - 6 \\ &= -4 - 6 - 2h \\ &= \underbrace{\Big[-10\Big] \oplus \Big[-2\Big]h}_{\text{output jet near } +2} \end{aligned}$$

ii. Then, from the *jet*:

- We get the Height-sign of $JULIE$ from the *constant term* $\Big[-10\Big]$ and since the Height-sign of the monomial function $h \to -10$ near 0 is $\langle -, - \rangle$, we get that Height-sign $JULIE$ near $+2 = \langle -, - \rangle$.
- Since the *constant term* $\Big[-10\Big]$ has no slope we get Slope-sign from the next term, namely the *linear term* $\Big[-2\Big]h$, and since the Slope-sign of the monomial function $h \to -2h$ near 0 is $\langle \diagdown, \diagdown \rangle$, we get that Slope-sign $JULIE$ near $+2 = \langle \diagdown, \diagdown \rangle$.
- Since the *constant term* $\Big[-10\Big]$ and the *linear term* $\Big[-2h\Big]$ both have no concavity, $JULIE$ has no Concavity-sign near $+2$.

DEMO 7.6 Get the feature-signs near -2 of the function given by

$$x \xrightarrow{PETER} PETER(x) = +3x + 6$$

i. We get the local input-output rule near -2:

$$\begin{aligned} -2+h \xrightarrow{PETER} PETER(-2+h) &= +3(-2+h) + 6 \\ &= +3(-2) + 3h + 6 \\ &= -6 + 3h + 6 \end{aligned}$$

general local analysis

$$= -6 + 6 + 3h$$
$$= \underbrace{\Big[0\Big] \oplus \Big[+3\Big]h}_{\text{output jet near } -2}$$

ii. Then, from the *jet*:

- Since the *constant term* is 0, we get Height-sign of $PETER$ from the next term, namely the *linear term* $\Big[+3\Big]h$ even though it is $small$. Since the Height-sign of the monomial function $h \to +3h$ near 0 is $\langle -, + \rangle$ we get that Height-sign $PETER$ near $-2 = \langle -, + \rangle$.
- Since the *constant term* $\Big[0\Big]$ has no slope we get Slope-sign from the next term, namely the *linear term* $\Big[+3\Big]h$, and since the Slope-sign of the monomial function $h \to +3h$ near 0 is $\langle \diagup, \diagup \rangle$ we get that Slope-sign $PETER$ near $-2 = \langle \diagup, \diagup \rangle$
- Since the *constant term* $\Big[0\Big]$ and the *linear term* $\Big[+3h\Big]$ both have no concavity, $PETER$ has no Concavity-sign near -2.

==

Everything below was commented out

==

5 Affine Functions: Global Analysis

In contrast with *local* analysis which involves only inputs that are near a given input, be it ∞ or x_0, *global* analysis involves, one way or the other, *all* inputs. We will see that, while the *local analysis* of all algebraic functions will turn out to remain essentially the same, the *global analysis* of each kind of algebraic functions will turn out to be vastly different.

In fact, with most functions, we will be able to solve only *some* global problems and mostly only *approximately* so. *Affine functions*, though, are truly *exceptional* in that we will be able to solve *all* global problems *exactly*.

Anyway, the first step in investigating the global behavior of a kind of algebraic function will always be to do the **general local analysis** of that kind of algebraic function, that is the local analysis of the *generic algebraic function* of that kind near ∞ and near a generic input x_0.

6 Smoothness

Given the function given by the generic global input-output rule

$$x \xrightarrow{AFFINE} AFFINE(x) = ax + b$$

generic local input-output rule
first derivative

the **generic local input-output rule** is:

$$x_0 + h \xrightarrow{AFFINE} AFFINE(x_0+h) = \underbrace{\Big[ax_0 + b\Big] \oplus \Big[a\Big]h}_{\text{jet near } x_0}$$

1. The constant term in the jet near x_0, namely $\big[ax_0 + b\big]$, is just the output *at* x_0. (See PROCEDURE 5.7 on page 245). In other words:

> **THEOREM 7.1** The function which outputs *at* the given input the *constant coefficient in the jet* of a given affine function near a given *bounded* input is the given affine function itself.

EXAMPLE 7.8. Observe that in the local input-output rule in DEMO 5.9 on page 252 the *constant coefficient* in the jet *near* -3, namely $+169.08$, is just the output *at*-3. (See DEMO 8.1 on page 342)

2. Since the *linear term* in the jet of an affine function near x_0, namely $\big[a\big]h$, is small, we have:

> **THEOREM 7.2 Approximate output near x_0.** For *affine* functions, inputs *near* x_0 have outputs that are *near* the output *at* x_0.

which, with the language we introduced in **??**, we can rephrase as:

> **THEOREM 7.3 thm:10-2** Continuity All affine functions are *continuous* at all inputs.

(In fact, we will see that this will also be the case for all the functions which we will be investigating *in this text*.)

3. The function which outputs the linear coefficient in the jet of a given affine function near a given input is called the **first derivative** of the given function.

7 The Essential Question

As always when we set out to investigate any kind of functions, the first thing we must do is to find out if the *offscreen graph* of an *affine function* consists of just the *local graph near* ∞ or if it also includes the *local graph near one or more* ∞*-height inputs*.

In other words, we need to ask the **Essential Question**:

- Do all *bounded inputs* have *bounded outputs*

or

- Are there *bounded inputs* that are ∞-height inputs, that is are there inputs whose nearby inputs have unbounded outputs?

Now, given a *bounded* input x, we have that:
- since a is bounded, ax is also bounded
- b is bounded

and so, altogether, we have that $ax + b$ is bounded and that the answer to the **Essential Question** is:

THEOREM 7.4 Approximate output near ∞ . Under an *affine function*, all bounded inputs return *bounded outputs*.

and therefore

THEOREM 7.5 Offscreen Graph . The *offscreen graph* of an *affine function* consists of just the *local graph near* ∞.

EXISTENCE THEOREMS

The notable inputs are those
- whose existence is forced by the *offscreen graph* which, by the **Bounded Height Theorem** for affine functions, consists of only the *local graph near* ∞.
- whose number is limited by the interplay among the three features

Since polynomial functions have no *bounded* ∞-height input, the only way a feature can change sign is near an input where the feature is 0. Thus, with affine functions, the feature-change inputs will also be 0-feature inputs.

None of the theorems, though, will indicate *where* the notable inputs are. The **Location Theorems** will be dealt with in the last part of the chapter.

EXAMPLE 7.9. When somebody has been shot dead, we can say that there is a murderer somewhere but locating the murderer is another story.

8 Slope-sign

Given the affine function $AFFINE_{a,b}$, that is the function given by the global input-output rule

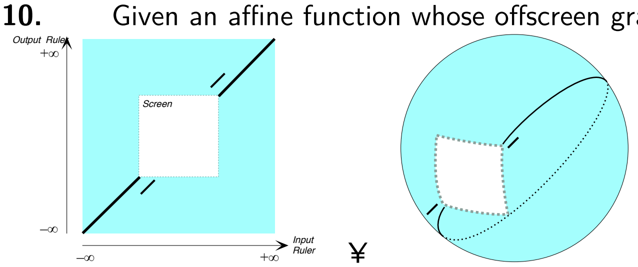
$$x \xrightarrow{AFFINE} AFFINE(x) = ax + b$$

recall that when x is near ∞ the **Slope-sign Near ∞ Theorem** says that:

- When a is $+$, $\text{Slope-Sign}|_{x \text{ near } \infty} = (\diagup, \diagup)$
- When a is $-$, $\text{Slope-Sign}|_{x \text{ near } \infty} = (\diagdown, \diagdown)$

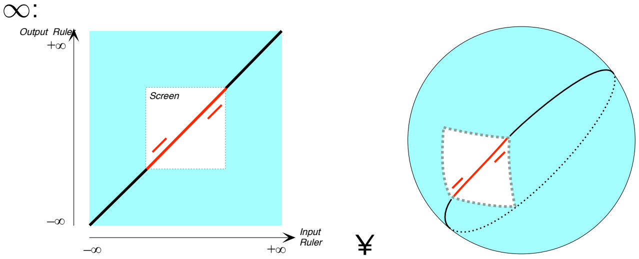
1. Since the slope does *not* changes sign as x goes through ∞ from the left side of ∞ to the right side of ∞, the slope need not change sign as x goes *across the screen* from the left side of ∞ to the right side of ∞ so there does not have to be a *bounded* Slope-sign change input:

EXAMPLE 7.10. Given an affine function whose offscreen graph is



Mercator view ¥ Magellan view

we don't need a bounded slope-sign change input to join smoothly the local graphs near ∞:



Mercator view ¥ Magellan view

2. In fact, not only does there not have to be a bounded slope-sign change input, there *cannot* be a bounded slope-sign change input since the *local* linear coefficient is equal to the *global* linear coefficient a and the slope must therefore be the same everywhere:

> **THEOREM 7.6 Slope-Sign Change Non-Existence.** An affine function has no *bounded* Slope-Sign Change input.

3. Another consequence of the fact that the local slope does not depend on x_0, and is thus the same everywhere, is that it is a feature of the function $AFFINE_{a,b}$ itself and so that the function $AFFINE_{a,b}$ has a **global slope** given by the global linear coefficient a. global slope

4. Moreover, the slope cannot be equal to 0 somewhere because the slope is equal to a everywhere. So, we also have:

> **THEOREM 7.7 0-Slope Input Non-Existence .** An affine function has *no* bounded 0-*slope* input.

9 Extremum

From the *optimization* viewpoint, an affine function has no extremum input, that is no bounded input whose output would be larger (or smaller) than the output of nearby inputs.

> **THEOREM 7.8 Extremum Non-existence .** An affine function has no bounded local extremum input.

10 Height-sign

Given the affine function $AFFINE_{a,b}$, that is the function given by the global input-output rule

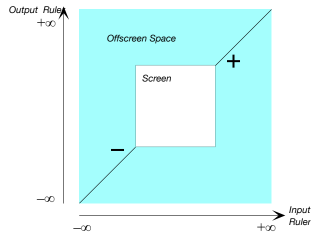
$$x \xrightarrow{AFFINE} AFFINE(x) = ax + b$$

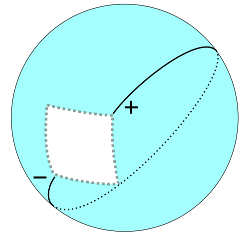
recall that when x is near ∞ the **Height-sign Near ∞ Theorem** says that:

- When a is $+$, $\text{Height-Sign}|_{x \text{ near } \infty} = (+, -)$
- When a is $-$, $\text{Height-Sign}|_{x \text{ near } \infty} = (-, +)$

1. Since the height changes sign as x goes from the left side to the right side of ∞ across ∞, the height must also change sign as x goes from the left side to the right side of ∞ *across the screen* so there has to be at least one *bounded Height-sign change input*:

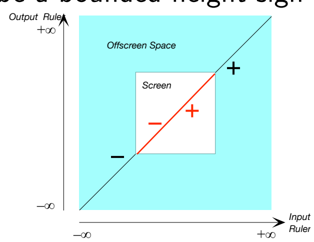
EXAMPLE 7.11. Given the affine function whose offscreen graph is straight

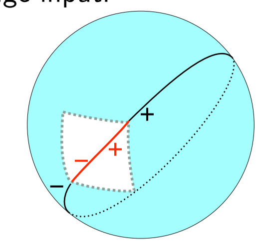


¥


Mercator view ¥ Magellan view

there has to be a bounded height-sign change input:



¥


Mercator view ¥ Magellan view

2. On the other hand, an affine function can have *at most one* 0-height input because, if it had more, it would have to have 0-slope inputs in-between the 0-height inputs which an affine function cannot have. So, we have:

> **Theorem 7.9 0-Height Existence.** An affine function has *exactly one* bounded 0-*height* input and it is a 0-height input: $x_{\text{Height-sign change}} = x_{\text{0-height}}$

11 Bounded Graph

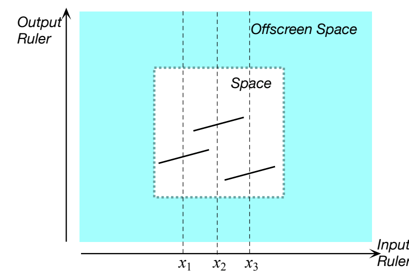
There are two ways to look at the shape of the bounded graph.

1. As a consequence of the **Bounded Height Theorem** for *affine* functions, the offscreen graph consists only of the local graph near ∞ and we can obtain the *forced bounded graph* by extrapolating smoothly the local graph near ∞.

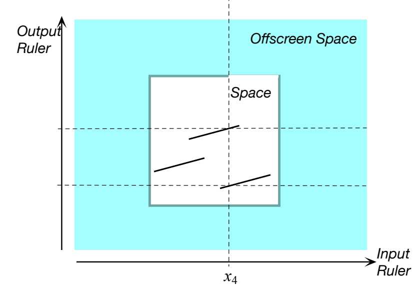
There remains however a question namely whether the extrapolated bounded graph is **straight** that is has no concavity. However, affine functions have no concavity and that settles the mater: the local graph near $-\infty$ and the local graph near $+\infty$ must be lined up and can therefore be joined smoothly with a straight line.

2. In the case of *affine functions*, it happens that we can also obtain the *bounded graph* by interpolating local graphs near bounded inputs:
We start from the local graphs near a number of bounded points as follows:

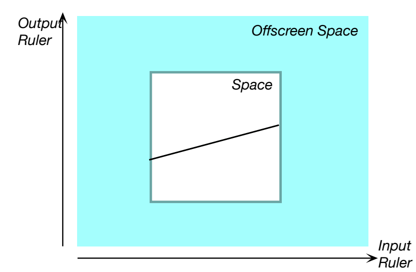
We construct local graphs near, say, three different *bounded* inputs, x_1, x_2, x_3. They would look something like this:



However, this is not possible because that would mean that inputs such as x_4 would have *two* outputs:



As a result, the *local* graphs near bounded inputs *must* all line up and so the *bounded* graph *must* be a straight line:



Of course, the bounded graph must line up with the local graph near ∞ as, otherwise, there would have to be a jump in the transition zone.

Location Theorems

Previously, we only established the *existence* of certain notable features of affine functions and this investigation was based on *graphic* considerations. Here we will investigate the *location* of the inputs where these notable features occur and this investigation will be based on *input-output rule* considerations.

12 0-Slope Location

We saw earlier that affine functions cannot have a 0-slope input. On the other hand, since the slope is the same everywhere, it is a global feature of the function itself and we have:

THEOREM 7.10 Global Slope-sign . Given the affine function $AFFINE_{a,b}$,
- When a is *positive*, Slope-sign $AFFINE = \diagup$.
- When a is *negative*, Slope-sign $AFFINE = \diagdown$

13 Locating Inputs Whose Output $= y_0$

The simplest global problem is, given a number y_0, to ask for the input numbers for which the function returns the output y_0.

PROCEDURE 7.5 Solve the equation $ax + b = y_0$ (See **??** on **??**.)

14 Locating Inputs Whose Output $> y_0$ Or $< y_0$

Given the affine function $AFFINE_{a,b}$, we are now ready to deal with the global problem of finding all inputs whose output is smaller (or larger) than some given number y_0.

EXAMPLE 7.12. G iven the inequation problem in which
- the *data set* consists of all numbers
- the *inequation* is

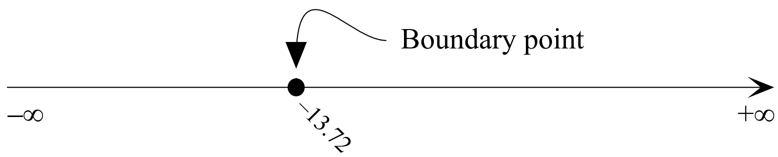
$$x \geqq -13.72$$

we locate separately.

i. The *boundary point* of the solution subset of the inequation problem is the solution of the *associated equation*:

$$x = -13.72$$

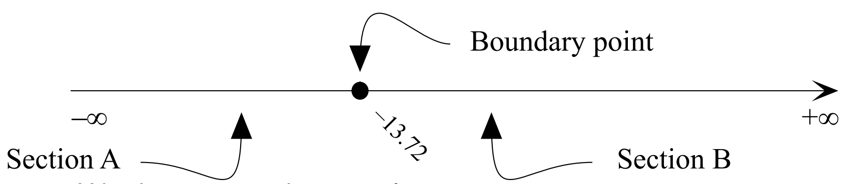
which, of course, is -13.72 and which we graph as follows since the boundary point is a *solution* of the inequation.



ii. The *interior* of the solution subset, that is the solution subset of the associated *strict inequation*

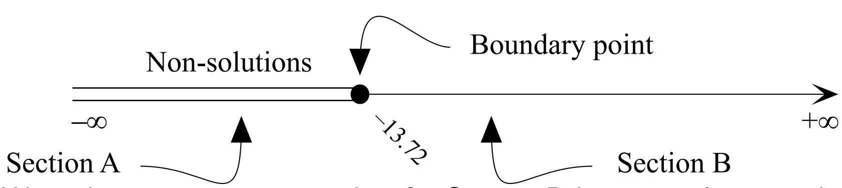
$$x > -13.72$$

i. The boundary point -13.72 separates the data set in two intervals, Section A and Section B:

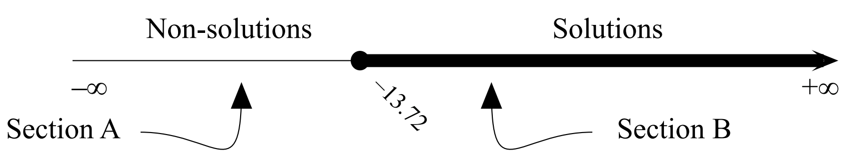


ii. We then test each interval:

- We pick $-1\,000$ as test number for Section A because, almost without a glance we know $-1\,000$ is going to be in Section A and because it is easy to check in the inequation: we find that $-1\,000$ is a *non-solution* so that, by **Pasch Theorem**, all numbers in Section A are *non-solutions*.



- We pick $+1\,000$ as test number for Section B because, almost without a glance we know $+1\,000$ is going to be in Section B and because it is easy to check in the inequation: we find that $+1\,000$ is a *solution* so that, by **Pasch Theorem**, all numbers in Section A are *solutions*.



15 Initial Value Problem

An **Initial Value Problem** asks the question:

What is the *input-output rule* of a function F given that:

- The function F is *affine*

- The *slope* of the function F is to be a given number a
- The *output* returned by the function F for a given input x_0 is to be a given number y_0.

Example 7.13. Find the global input-output rule of the function $KATE$ given that it is affine, that its slope is -3 and that the output for the input $+2$ is $+5$.
We use all three given pieces of information:
i. Since we are given that $KATE$ is an affine function, we give temporary names for the dilation coefficient, say a, and for the constant term, say b, and we write the global input-output rule of $KATE$ in terms of these names:

$$x \xrightarrow{KATE_{a,b}} KATE_{a,b}(x) = ax + b$$

ii. By the **Local Slope Theorem**, the slope is equal to the dilation coefficient:

$$-3 = a$$

which give the equation $a = -3$
iii. Since the output for the input $+2$ is $+5$, we write

$$\begin{aligned} KATE_{a,b}(x)\big|_{x:=+2} &= +5 \\ ax + b\big|_{x:=+2} &= +5 \\ a(+2) + b &= +5 \end{aligned}$$

which give the equation $2a + b = +5$
iv. So we must solve the system of two equations for two unknowns a and b:
===========

$$\text{AND}\begin{cases} a = -3 \\ 2a + b = +5 \end{cases}$$

This kind of system is very simple to solve since we need only replace a by -3 in the second equation to get the equation:

$$2(-3) + b = +5$$

which we solve using the Reduction Method:

$$\begin{aligned} -6 + b &= +5 \\ -6 + b + 6 &= +5 + 6 \\ b &= +11 \end{aligned}$$

v. So, the global input-output rule for $KATE$ is

$$x \xrightarrow{KATE_{-3,+11}} KATE_{-3,+11}(x) = -3x + 11$$

16 Boundary Value Problem

A **Boundary Value Problem** asks the question:

What is the *input-output rule* of a function F, given that:

- The function F is *affine*
- The *output* returned by the function F for a given input x_1 is to be a given number y_1.
- The *output* returned by the function F for a given input x_2 is to be a given number y_2.

In other words, we want to find an affine function F such that:

$$\text{BOTH}\begin{cases} x_1 \xrightarrow{F} F(x_1) = y_1 \\ x_2 \xrightarrow{F} F(x_2) = y_2 \end{cases}$$

EXAMPLE 7.14. Find the global input-output rule of the function $DAVE$ given that it is affine, that the output for the input $+2$ is -1 and that the output for the input -4 is -19.

We use all three pieces of information that we are given:

i. Since we are given that $DAVE$ is an affine function, we give temporary names for the dilation coefficient, say a, and for the constant term, say b, and we write the global input-output rule of $DAVE$ in terms of these names:

$$x \xrightarrow{DAVE_{a,b}} DAVEa, b(x) = ax + b$$

ii. Since the output for the input $+2$ is -1 we write:

$$\begin{aligned} DAVE_{a,b}(x)\big|_{x:=+2} &= -1 \\ ax + b\big|_{x:=+2} &= -1 \\ a(+2) + b &= -1 \end{aligned}$$

which give the equation $+2a + b = -1$

iii. Since the output for the input -4 is -19 we write:

$$\begin{aligned} DAVE_{a,b}(x)\big|_{x:=-4} &= -19 \\ ax + b\big|_{x:=+2} &= -19 \\ a(-4) + b &= -19 \end{aligned}$$

which give the equation $-4a + b = -19$

iv. So we must solve the system of two equations for two unknowns a and b:

$$\begin{cases} +2a + b = -1 \\ -4a + b = -19 \end{cases}$$

This kind of system is a bit more complicated to solve but since b appears in both equations, we replace one of the two equations, say the second one, by "the first one minus the second one":

$$\begin{cases} +2a+b=-1 \\ [+2a+b]-[-4a+b]=[-1]-[-19] \end{cases}$$

This gives us:

$$\begin{cases} +2a+b=-1 \\ +2a+b+4a-b=-1+19 \end{cases}$$

that is

$$\begin{cases} +2a+b=-1 \\ +6a=+18 \end{cases}$$

that is

$$\begin{cases} +2a+b=-1 \\ \dfrac{+6a}{+6}=\dfrac{+18}{+6} \end{cases}$$

that is

$$\begin{cases} +2a+b=-1 \\ a=+3 \end{cases}$$

and now we replace in the first equation a by $+3$:

$$\begin{cases} +2a+b=-1|_{a:=+3} \\ a=+3 \end{cases}$$

that is

$$\begin{cases} +2(+3)+b=-1 \\ a=+3 \end{cases}$$

that is

$$\begin{cases} +6+b=-1 \\ a=+3 \end{cases}$$

and we reduce the first equation

$$\begin{cases} +6+b-6=-1-6 \\ a=+3 \end{cases}$$

which gives us, finally

$$\begin{cases} b=-7 \\ a=+3 \end{cases}$$

v. So the global input-output rule of $DAVE$ is

$$x \xrightarrow{DAVE_{+3,-7}} DAVE_{+3,-7}(x)=+3x-7$$

17 Piecewise affine functions

Part III

(Laurent) Polynomial Functions

xxxxxxxxxx]

trinomial function

Chapter 8

Quadratic Functions

=======Begin LOOK UP=======

1 Trinomial Functions

There is of course no reason why the base function could not itself be a binomial function. In fact, this can very well be the case and the sum function will then be called a **trinomial function**.

EXAMPLE 8.1. **L** et $BASE$ be specified by the global input-ouput rule

$$x \xrightarrow{BASE} BASE(x) = (-3)x^0 \oplus (+7)x^{+1}$$

and let $ADD\text{-}ON$ be specified by the global input-ouput rule

$$x \xrightarrow{ADD\text{-}ON} ADD\text{-}ON(x) = (+5)x^{+3}$$

then the SUM function is specified by the global input-ouput rule

$$\begin{aligned} x \xrightarrow{SUM} SUM(x) &= (-3)x^0 \oplus (+7)x^{+1} \oplus (+5)x^{+3} \\ &= -3 + 7x + 5x^{+3} \end{aligned}$$

Quadratic␣function
quadratic␣coefficient
linear␣coefficient
constant␣coefficient

EXAMPLE 8.2. **L** et $BASE$ be specified by the global input-ouput rule

$$x \xrightarrow{BASE} BASE(x) = (-3)x^{+1} \oplus (+7)x^0$$

and let $ADD\text{-}ON$ be specified by the global input-ouput rule

$$x \xrightarrow{ADD\text{-}ON} ADD\text{-}ON(x) = (+5)x^{-2}$$

then the SUM function is specified by the global input-ouput rule

$$\begin{aligned} x \xrightarrow{SUM} SUM(x) &= (-3)x^{+1} \oplus (+7)x^0 \oplus (+5)x^{-2} \\ &= -3x + 7 + 5x^{-2} \end{aligned}$$

=======End LOOK UP =======

Quadratic functions are specified by global input-output rules like the generic global input-output rule:

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{ax^{+2} \oplus bx^{+1} \oplus cx^0}_{\text{output-specifying code}}$$

which we usually write

$$= \underbrace{ax^2 + bx + c}_{\text{output-specifying code}}$$

where a, called the **quadratic coefficient**, b, called the **linear coefficient**, and c, called the **constant coefficient**, are the *bounded* numbers that specify the function $QUADRATIC$.

EXAMPLE 8.3. The quadratic function $RINA$ specified by the quadratic coefficient -23.04, the linear coefficient -17.39 and the constant coefficient $+5.84$ is the function specified by the global input-output rule

$$x \xrightarrow{RINA} RINA(x) = \underbrace{-23.04}_{\text{quadratic coeff.}} x^2 \underbrace{-17.39}_{\text{linear coeff.}} x \underbrace{+5.84}_{\text{constant coeff.}}$$

It is worth noting again that

> **CAUTIONARY NOTE 8.1** The terms in the global input output rule *need not* be written in order of *descending* exponent. This is just a habit we have.

EXAMPLE 8.4. The function specified by the global input-output rule

$$x \xrightarrow{BIBI} BIBI(x) = +21.03x^2 - 31.39x + 5.34$$

could equally well be specified by the global input-output rule

$$x \xrightarrow{BIBI} BIBI(x) = +5.34 + 21.03x^2 - 31.39x$$

or by the global input-output rule

$$x \xrightarrow{BIBI} BIBI(x) = -31.39x + 5.34 + 21.03x^2$$

We now introduce some standard terminology to help us describe very precisely what we will be doing. The output-specifying code of the affine function specified by

term
quadratic term
linear term
constant term
affine␣part

$$x \xrightarrow{AFFINE} QUADRATIC(x) = \underbrace{ax^2 + bx + c}_{\text{output-specifying code}}$$

consists of three **terms**:

- ax^2 which is called the **quadratic term**.
- bx which is called the **linear term**.
- c which is called the **constant term**,

and there is of course also

- $bx + c$ which is called the **affine part**

EXAMPLE 8.5. The output-specifying code of the function specified by the global input-output rule

$$x \xrightarrow{RINA} RINA(x) = \underbrace{-23.04}_{\text{quadratic coeff.}} x^2 \underbrace{-31.39}_{\text{linear coeff.}} x \underbrace{+5.84}_{\text{constant coeff.}}$$

consists of three terms:

$$= \underbrace{-23.04x^2}_{\text{quadratic term}} \underbrace{-31.39x}_{\text{linear term}} \underbrace{+5.34}_{\text{constant term}}$$

LANGUAGE NOTE 8.1 Whether we look upon c as the constant *coefficient*, that is as the *coefficient* of x^0 in the constant *term* cx^0 or as the constant *term* cx^0 itself with the power x^0 "going without saying" will be clear from the context.

2 Output *at* x_0

1. Remember from subsection 8.1 that x_0 is a *generic given input*, that is that x_0 is a *bounded* input that has been *given* but whose identity remains *undisclosed* for the time being.

2. We will use

PROCEDURE 8.1

i. *Declare* that x is to be replaced by x_0

$$x\Big|_{x \leftarrow x_0} \xrightarrow{QUADRATIC} QUADRATIC(x)\Big|_{x \leftarrow x_0} = ax^2 + bx + c\Big|_{x \leftarrow x_0}$$

which gives:

$$x_0 \xrightarrow{QUADRATIC} QUADRATIC(x_0) = \underbrace{a{x_0}^2 + bx_0 + c}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into an output *number*:

$$= ax_0^2 + bx_0 + c$$

which gives the input-output pair

$$\left(x_0, ax_0^2 + bx_0 + c\right)$$

Demo 8.1

i. We *declare* that x is to be replaced by -3

$$x\Big|_{x \leftarrow -3} \xrightarrow{AVIA} AVIA(x)\Big|_{x \leftarrow -3} = +21.03x^2 - 32.67x + 71.07\Big|_{x \leftarrow -3}$$

which gives

$$-3 \xrightarrow{AVIA} AVIA(-3) = \underbrace{+21.03(-3)^2 - 32.67(-3) + 71.07}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into an output *number*:

$$= \underbrace{+189.26 \oplus +98.01 \oplus +71.07}_{\text{output number at } -3}$$

$$= \underbrace{+358.34}_{\text{output number at } -3}$$

which gives the input-output *pair*

$$(-3\ ,\ \underbrace{+358.34}_{\text{output number at } -3})$$

3. However, as already discussed in **?? ??** and as has already been the case with *monomial* functions and *affine* functions, instead of getting the output *number* returned by a quadratic function *at* a given input, we will usually want *all* the outputs returned by the quadratic function for inputs *near* that given input. So, instead of getting the single *input-output pair at* the given input, we will get the *local input-output rule* with which to get *all* the input-output pairs *near* the given input.

3 Output *near* ∞

output jet

As already discussed in subsection 8.2 Output *near* ∞, in order to input a neighborhood of ∞, we will *declare* that "x is near ∞" but write only x after that. This, again, is extremely dangerous as it is easy to forget that what we write may be TRUE *only* because x has been declared to be near ∞.

1. output jet (https://en.wikipedia.org/wiki/Jet_(mathematics)) We will *execute* the output-specifying code, namely $ax^2 + bx + c$, into an *OUTPUT JET*, that is with the terms in *descending* order of sizes, which, since here x is *large*, means that here the powers of x must be in *descending* order of exponents. We will then have the *local input-output rule near* ∞:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{\text{Powers of } x \text{ in } \textit{descending order of exponents}}_{\text{output jet near } \infty}$$

Example 8.6. Given the function specified by the global input-output rule

$$x \xrightarrow{RIBA} RIBA(x) = -61.03 - 82.47x + 45.03x^2$$

To get the *output jet* near ∞, we first need to get the *order of sizes*.
i. -61.03 is *bounded*
ii. -82.47 is *bounded* and x is *large*. So, since *bounded* $\cdot$ *large* = *large*, $-82.47 \cdot x$ is *large*
iii. $+45.03$ is *bounded* and x is *large*. So, since *bounded* $\cdot$ *large* = *large*, $+45.03 \cdot x$ is *large* too. But *large* $\cdot$ *large* is larger in size than *large* so $+45.03 \cdot x^2$ is even larger than $-82.47 \cdot x$
So, in the output jet near ∞, $+45.03x^2$ must come first, $-82.47x$ comes second and -61.03 comes third
Then, we can write the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{RIBA} RIBA(x) = \underbrace{+45.03x^2 - 82.47x - 61.03}_{\text{output jet near } \infty}$$

2. So, we will use:

Procedure 8.2
i. *Declare* that x is near ∞:

$$x\,\Big|_{x \text{ near } \infty} \xrightarrow{QUADRATIC} QUADRATIC(x)\,\Big|_{x \text{ near } \infty} = ax^2 + bx + c\,\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{ax^2 + bx + c}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into an *output jet*:

$$= \underbrace{\Big[a\Big]\, x^2 \oplus \Big[b\Big]\, x \oplus \Big[c\Big]}_{\text{output jet near } \infty}$$

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{\Big[a\Big]\, x^2 \oplus \Big[b\Big]\, x \oplus \Big[c\Big]}_{\text{output jet near } \infty}$$

(The *output jet* in the local input-output rule near ∞ looks the same as the *output-specifying code* in the given global input-output rule but that is only because *here* the output-specifying code *happened* to be written in *descending* order of exponents.)

Demo 8.2

i. We *declare* that x is near ∞:

$$x\,\Big|_{x \text{ near } \infty} \xrightarrow{KINA} KINA(x)\,\Big|_{x \text{ near } \infty} = -61.03 + 51.32x^2 - 82.47x\,\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{NINA} KINA(x) = \underbrace{-61.03 + 51.32\, x^2 - 82.47\, x}_{\text{output-specifying code}}$$

ii. We *execute* the output-specifying code into an *output jet*:

$$= \Big[+51.32\Big]\, x^2 \oplus \Big[-82.47\Big]\, x \oplus \Big[-61.03\Big]$$

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{KINA} KINA(x) = \underbrace{\Big[+51.32\Big]\, x^2 \oplus \Big[-82.47\Big]\, x \oplus \Big[-61.03\Big]}_{\text{output jet near } \infty}$$

(The *output jet* in the local input-output rule near ∞ does *not* look the same as the *output-specifying code* in the *global* input-output rule because *here* the output-specifying code happened *not* to be in descending order of exponents.)

3. The reason we use *jets* here is that the term *largest in size* is the *first* term so that to *approximate* the output we need only write the *first* term in the jet and just replace the remaining terms by [...] which stands for "something too small to matter here". In other words, addition formula

THEOREM 8.1 Approximate output near ∞. For *quadratic* functions, what contributes most to the output near ∞ is the *highest degree term* in the output jet near ∞:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \Big[a\Big]x^2 + [...]$$

EXAMPLE 8.7. Given the function specified by the global input-output rule

$$x \xrightarrow{KINA} KINA(x) = -61.03 + 51.32x^2 - 82.47x$$

near ∞ we will often just use the *approximation*

$$x \text{ near } \infty \xrightarrow{KINA} KINA(x) = \Big[+51.32\Big] x^2 \oplus [...]$$

4 Output *near x_0*

We now deal with the output of the neighborhood of some *given* bounded input x_0.

1. In order to input a neighborhood of a given input x_0 we will declare that $x \leftarrow x_0 \oplus h$ that is that x is to be replaced by $x_0 \oplus h$. As a result, we will have to compute $(x_0 \oplus h)^2$ for which we will have to use an **addition formula** from ALGEBRA, namely **??** in appendix 1 on page page 519.

2. We can then *execute* the input-output specifying phrase into an *output jet* that is with the terms in *descending order of sizes* which here, since h is *small*, means that the powers of h will have to be in *ascending* order of exponents. We will then have the *local input-output rule* near the given input:

$$x_0 \oplus h \xrightarrow{QUADRATIC} QUADRATIC(x_0 \oplus h) = \underbrace{\text{Powers of } h \text{ in } \textit{ascending order of exponents}}_{\text{output jet near } \infty}$$

We will therefore use:

PROCEDURE 8.3

i. *Declare* that x is near x_0: (So x is to be replaced by $x_0 + h$.)

$$x\Big|_{x \leftarrow x_0+h} \xrightarrow{QUADRATIC} QUADRATIC(x)\Big|_{x \leftarrow x_0+h} = ax^2 + bx + c\Big|_{x \leftarrow x_0+h}$$

which gives:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0 + h) = \underbrace{a(x_0 + h)^2 + b(x_0 + h) + c}_{\text{output-specifying code}}$$

output jet near x_0

ii. *Execute* the output-specifying code into an *output jet*:

$$= a\left(x_0^2 + 2x_0h + h^2\right) + b\,(x_0 + h) + c$$
$$= ax_0^2 \oplus 2ax_0\, h \oplus a\, h^2$$
$$\oplus\ bx_0 \oplus b\, h$$
$$\oplus\ c$$
$$= \underbrace{\Big[ax_0^2 + bx_0 + c\Big] \oplus \Big[2ax_0 + b\Big]h \oplus \Big[a\Big]h^2}_{\text{output jet near } x_0}$$

which gives the *local input-output rule* near x_0:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0 + h) = \underbrace{\Big[ax_0^2 + bx_0 + c\Big] \oplus \Big[2ax_0 + b\Big]h \oplus \Big[a\Big]h^2}_{\text{output jet near } x_0}$$

Demo 8.3

i. We *declare* that x is near -3: (So x is to be replaced by $-3 + h$.)

$$x\,\Big|_{x\leftarrow -3+h} \xrightarrow{ARNA} ARNA(x)\,\Big|_{x\leftarrow -3+h} = -32.67x + 71.07 + 81.26x^2\,\Big|_{x\leftarrow -3+h}$$

which gives

$$-3 + h \xrightarrow{ARNA} ARNA(-3 + h) = \underbrace{-32.67(-3 + h) + 71.07 + 81.26(-3 + h)^2}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into an *output jet*:

$$= -32.67(-3 + h) + 71.07 + 81.26\left((-3)^2 + 2(-3)h + h^2\right)$$
$$= -32.67(-3) - 32.67h$$
$$+\ 71.07$$
$$+\ 81.26(-3)^2 + 81.26(2)(-3)h + 81.26h^2$$
$$= +98.01 \oplus -32.67\, h$$
$$\oplus +71.07$$
$$\oplus +731.34 \oplus -487.56\, h \oplus +81.26\, h^2$$
$$= \Big[+98.01 + 71.07 + 731.34\Big] \oplus \Big[-32.67 - 487.56\Big]h \oplus \Big[+81.26\Big]h^2$$
$$= \underbrace{\Big[+900.42\Big] \oplus \Big[-519.63\Big]h \oplus \Big[+81.26\Big]h^2}_{\text{output jet near } -3}$$

which gives the *local input-output rule* near -3:

$$-3+h \xrightarrow{ARNA} ARNA(-3+h) = \underbrace{\Big[+900.42\Big] \oplus \Big[-519.63\Big]h \oplus \Big[+81.26\Big]h^2}_{\text{output jet near } -3}$$

3. When all we want is a feature-sign, though, the above procedure is very inefficient and we will then use the following procedure based directly on the fact that a *quadratic function* is the addition of:

- a *square function*, (See DEFINITION 6.2 on page 262)
- a *linear function*, (See **??** on **??**.)
- a *constant function*. (See **??** on **??**.)

that is:

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{bx^2}_{\text{square}} \oplus \underbrace{cx}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

We declare that x is near x_0 that is that x must be replaced by x_0+h:

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{b\,(x_0+h)^2}_{\text{square}} \oplus \underbrace{c\,(x_0+h)}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

The output of the local input-output rule near x_0 will have to be a *jet*:

$$x_0+h \xrightarrow{QUADRATIC} QUADRATIC(x_0+h) = \Big[\quad\Big] \oplus \Big[\quad\Big]h \oplus \Big[\quad\Big]h^2$$

and we want to be able to get any one of the coefficients of the output jet without having to compute any of the other coefficients. So, what we will do is to get the contribution of each monomial function to the term we want. This requires us to have the *addition formula* at our finger tips:

a.

$$(x_0+h)^2 = x_0^2 + 2x_0h + h^2 \text{ (See ?? on page 519)}$$

More precisely,

i. If we want the *coefficient* of h^0 in the output jet:

- The square function contributes bx_0^2
- The linear function contributes cx_0
- The constant function contributes d

so we have:

$$x_0+h \xrightarrow{QUADRATIC} QUADRATIC(x_0+h) = \Big[bx_0^2 + cx_0 + d\Big] \oplus \Big[\quad\Big]h \oplus \Big[\quad\Big]h^2$$

ii. If we want the *coefficient* of h^1 in the output jet:

- The square function contributes $2bx_0$

- The linear function contributes c
- The constant function contributes nothing

so we have:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0 + h) = \big[\quad\big] \oplus \big[2bx_0 + c\big]h \oplus \big[\quad\big]h^2$$

iii. If we want the *coefficient* of h^2 in the output jet:

- The square function contributes c
- The linear function contributes nothing
- The constant function contributes nothing

so we have:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0 + h) = \big[\quad\big] \oplus \big[\quad\big]h \oplus \big[c\big]h^2$$

5 Local graphs

Just the way we get the *plot point at* a given *bounded* input from the *output number at* that input, we get the *local graph near* any given input, be it *bounded* or *infinity*, from the *output jet near* that input.

Procedure 8.4

1. Get the *local input-output rule* near ∞ using PROCEDURE 8.2 on page 343:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{\big[a\big]\, x^2 \oplus \big[b\big]\, x \oplus \big[c\big]}_{\text{output jet near } \infty}$$

2. Get the *local graph* near ∞ of each term:

a. For the *quadratic term*, graph near ∞ the monomial function $x \to \big[a\big]x^2$ (See PROCEDURE 6.6 on page 305.)

b. For the *linear term*, graph near ∞ the monomial function $x \to \big[b\big]x$ (See ?? on ??.)

c. For the *constant term*, graph near ∞ the monomial function $x \to \big[c\big]$(See ?? on ??.)

3. Get the *local graph* near ∞ of $QUADRATIC$ by adding to the local graph of the *quadratic term* the local graph of the *linear term* and the local graph of the *constant term*.(See ??)

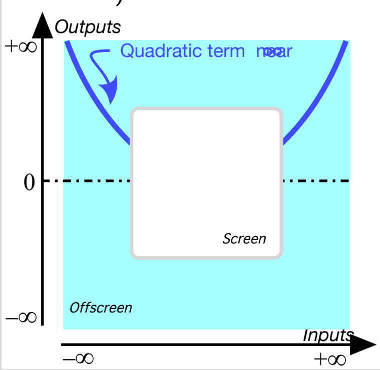
Demo 8.4

1. We get the local *input-output rule* near ∞: (See DEMO 8.2 on page 344)

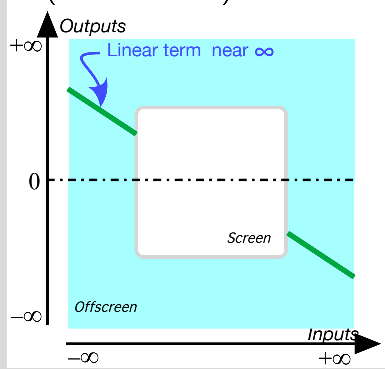
$$x \text{ near } \infty \xrightarrow{KINA} KINA(x) = \underbrace{\left[+51.32\right]x^2 + \left[-82.47\right]x + \left[-61.03\right]}_{\text{output jet near } \infty}$$

2. We get the *local graph* near ∞ of each term:

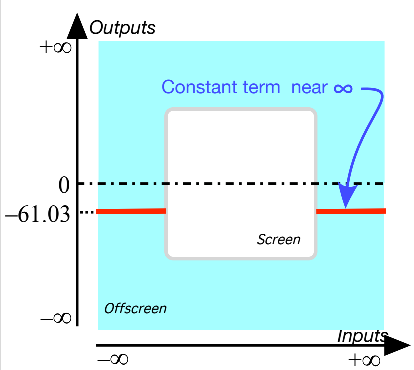
a. For the graph of the *quadratic* term, we graph the monomial function $x \to \left[+51.32\right]x^2$ near ∞ (See ?? on ??)



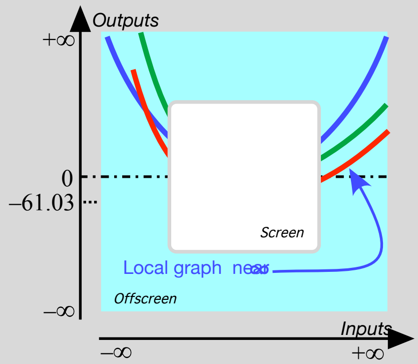
b. For the graph of the *linear* term, we graph the monomial function $x \to \left[-82.47\right]x$ near ∞ (See ?? on ??)



c. For the graph of the *constant* term, we graph the monomial function $x \to \left[-61.03\right]$ near ∞ (See ?? on ??)



3. We get the *local graph* near ∞ of $KINA$ by adding to the local graph of the *quadratic* term the local graph of the *linear* term and the local graph of the *constant* term. (See ?? on ??)



Procedure 8.5

1. Get the *local input-output rule* near x_0 using PROCEDURE 8.3 on page 345:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0 + h) = \underbrace{\Big[ax_0^2 + bx_0 + c\Big] \oplus \Big[2ax_0 + b\Big]h \oplus \Big[a\Big]h^2}_{\text{output jet near } x_0}$$

2. Get the *local graphs* near 0 of each term:

a. For the *constant term*, graph near 0 the monomial function $x \to \Big[ax_0^2 + bx_0 + c\Big]$. (See ?? on ??.)

b. For the *linear term*, graph near 0 the monomial function $x \to \Big[2ax_0 + b\Big]x$. (See ?? on ??.)

c. For the *quadratic term*, graph near 0 the monomial function $x \to \Big[a\Big]x^2$. (See ?? on ??.)

3. Get the *local graph* near x_0 of $QUADRATIC$ by adding to the local graph of the *constant term* the local graph of the *linear term*, the local graph of the *quadratic term*.

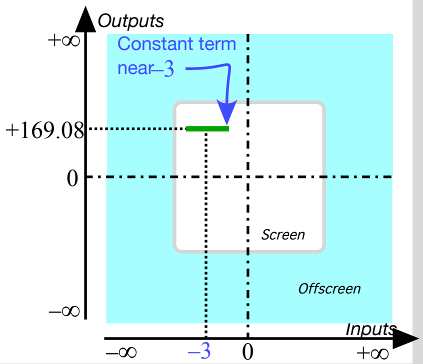
Demo 8.5

1. We get the local *input-output rule* near -3. (See DEMO 8.3 on page 346):

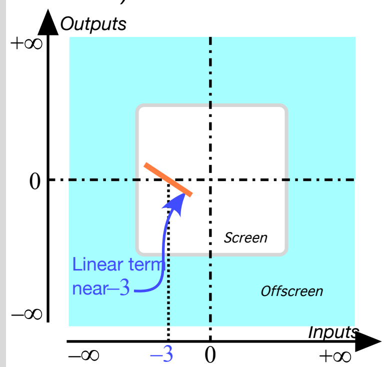
$$-3 + h \xrightarrow{ARNA} ARNA(-3 + h) = \underbrace{\Big[+900.42\Big] \oplus \Big[-519.63\Big]h \oplus \Big[+81.26\Big]h^2}_{\text{output jet near } -3}$$

2. We get the local graph near -3 of each term:

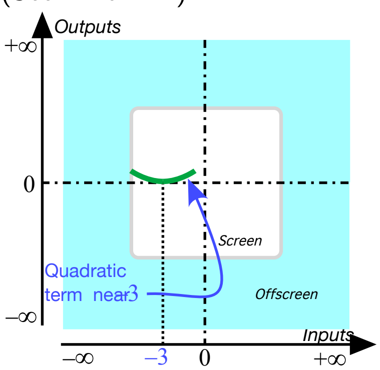
a. For the *constant* term, we graph near 0 the monomial function $x \to \left[+900.428 \right]$. (See ?? on ??)



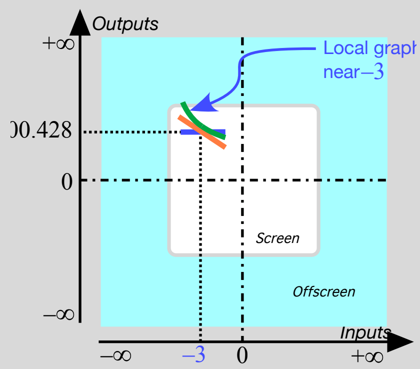
b. For the *linear* term, we graph near 0 the monomial function $x \to \left[-519.63 \right] x$ (See ?? on ??)



c. For the *quadratic* term, we graph near 0 the monomial function $x \to \left[+81.26 \right] x^2$: (See ?? on ??)



3. We get the local graph near -3 of $ARNA$ by adding to the local graph of the *constant term* the local graph of the *linear term* and the local graph of the *quadratic term*. (See ?? on ??)



6 Local Feature-signs

As we saw in **?? ??**, a feature-sign near a given input, be it near ∞ or near x_0, can be read from the *local graph* and so we already know how to proceed:

i. Get the *local input-output rule* near the given input (See PROCEDURE 8.2 on page 343 when the given input is ∞ or PROCEDURE 8.3 on page 345 when the given input is x_0.)

ii. Get the *local graph* from the local input-output rule (See PROCEDURE 8.4 on page 348.)

iii. Get the *feature-sign* from the *local graph*. (See **?? ??**.)

However, things are in fact much simpler: Given an input, be it ∞ or a bounded input x_0, to get a required feature-sign near that given input, we look for the term in the output jet near that input that

i. Has the required feature.

and

ii. Is the largest-in-size of all those terms with the required feature.

So, as we will now see, we usually need to get only *one* term in the output jet rather than the whole output jet.

1. Near *infinity* things are quite straightforward because, for a quadratic function, the first term in the output jet near ∞ is both the *largest-in-size* and a *regular* monomial so that it has *all three features*:

PROCEDURE 8.6

i. Get the *approximate* local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{QUADRATIC} QUADRATIC(x) = \underbrace{\left[a\right]x^2 \oplus \left[b\right]x \oplus \left[c\right]}_{\text{output jet near } \infty}$$

$$= \underbrace{\big[a\big]x^2 \oplus [...]}_{\text{approximate output jet near } \infty}$$

ii. Then, in the *approximate output jet* near ∞:

- Get the *Height-sign*, the *Slope-sign* and the *Concavity-sign* all from the *quadratic term* $\big[a\big]x^2$ because the next terms, $\big[b\big]x$ and $\big[c\big]$ are *too small to matter*. (Not to mention the fact that a linear term has no concavity and a constant term has neither concavity nor slope.)

Demo 8.6 et $CELIA$ be the function specified by

$$x \xrightarrow{CELIA} CELIA(x) = -2x^2 + 63x - 155$$

Get Height-sign near ∞.

i. We get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{CELIA} CELIA(x) = -2x^2 + 63x - 155$$
$$= \underbrace{\big[-2\big]x^2 \oplus \big[+63\big]x \oplus \big[-155\big]}_{\text{output jet near } \infty}$$

ii. We get *Height-sign* from the *quadratic* term $\big[-2\big]x^2$ because the linear term $\big[+63\big]x$ and the *constant* term $\big[-155\big]$ are *too small to matter*.

iii. Since the *quadratic coefficient* $\big[-2\big]$ is *negative*, we get that Height-sign $CELIA$ near $\infty = \langle -, - \rangle$. (Seen from ∞.)

Demo 8.7 et $PETER$ be the function specified by the global input-output rule

$$x \xrightarrow{DIETER} DIETER(x) = +3.03x^2 - 81.67x + 46.92$$

Get Slope-signs near ∞.

i. We get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{DIETER} DIETER(x) = +3.03x^2 - 81.67x + 46.92$$
$$= \underbrace{\big[+3.03\big]x^2 \oplus \big[-81.67\big]x \oplus \big[+46.92\big]}_{\text{output jet near } \infty}$$

ii. We get *Slope-sign* from the *quadratic* term $\big[+3.03\big]x^2$ because the *linear* term $\big[-81.67\big]$ is *too small to matter* and the *constant* term has *no* slope.

critical for the Height
critical for the Slope

Since the *linear coefficient* $+3$ is positive, we get that Slope-sign $DIETER$ near $\infty = \langle \diagup, \diagup \rangle$. (Seen from ∞.)

2. Near a *bounded input* though, things are a bit more complicated:
i. The *first* term in the output jet is *usually* the *largest-in-size* so that it gives the Height-sign. However, the first term *usually* has neither Slope nor Concavity because the first term is *usually* a constant term.
ii. The *second* term in the output jet is *usually* too smalll-in-size to change the Height-sign as given by the first term but it is *usually* the *largest-in-size* term that can give the Slope-sign. However, the second term has no Concavity because the second term is *usually* a linear term.
iii. The third *term* in the output jet is *usually* too smalll-in-size to change the Height-sign given by the first term and the Slope-sign given by the second term but it is *usually* the *only term* that can give the Concavity-sign.
So we can *usually* read each feature-sign directly from the appropriate term in the output jet - keeping in mind that the exceptional monomial functions do not have all the features.
However, near a *bounded input*, the given bounded input may turn out to be *critical* for the local feature:
i. If the *constant term* in the output jet is 0, then the term which gives the Height-sign can be either the *linear term* or even the *quadratic term* if the *linear term* is 0. The bounded input is then said to be **critical for the Height**.
ii. If the *linear term* in the output jet is 0, then the term which gives the Slope-sign is the *quadratic term*. The bounded input is then said to be **critical for the Slope**.
So, we *usually* need to compute only one coefficient in the output jet. But if the given bounded input turns out to be *critical* for that feature, then we need to compute the next coefficient: So we use

PROCEDURE 8.7
i. Get the local input-output rule near x_0:

$$x_0 + h \xrightarrow{QUADRATIC} QUADRATIC(x_0+h) = a(x_0+h)^2 + b(x_0+h) + c$$
$$= a\left(x_0^2 + 2x_0h + h^2\right) + b\,(x_0+h) + c$$
$$= \underbrace{\Big[ax_0^2 + bx_0 + c\Big] \oplus \Big[2ax_0 + b\Big]h \oplus \Big[a\Big]h^2}_{\text{output jet near } x_0}$$

ii. Then, in the *output jet* near x_0:
- Get the *Height-sign* from the *constant term* $\Big[ax_0^2 + bx_0 + c\Big]$ (The

linear term and the *quadratic term* are *too small to matter*.)
If the *constant coefficient* is 0, get the *Height-sign* from the *linear term* $\Big[2ax_0 + b\Big]h$. (The quadratic term is *too small to matter*.)
If the *linear coefficient* is 0, get the *Height-sign* from the *quadratic term* $\Big[a\Big]h^2$.

- Since the *constant term* has no slope, get the *Slope-sign* from the *linear term* $\Big[2ax_0 + b\Big]h$.
  If the *linear coefficient* is 0, get the *Slope-sign* from the *quadratic term* $\Big[a\Big]h^2$
- Since both the *constant term* and the *linear term* have no concavity, we get Concavity-sign from the *quadratic term*..

DEMO 8.8 et $ARNA$ be the function specified by the global input-output rule

$$x \xrightarrow{ARNA} ARNA(x) = -32.67x + 71.07 + 81.26x^2$$

Get the feature-signs near -3.

i. We get the local input-output rule near -3 as in DEMO 8.3 on page 346:

$$-3+h \xrightarrow{ARNA} ARNA(-2+h) = \underbrace{-32.67(-3+h) + 71.07 + 81.26(-3+h)^2}_{\text{output specifying code}}$$
$$= \underbrace{\Big[+900.428\Big] \oplus \Big[-519.63\Big]h \oplus \Big[+81.26\Big]h^2}_{\text{output jet near } -3}$$

ii. Then, from the *jet*:

- Since the *constant term* $\Big[+900.428\Big]$ is *positive*, we get that Height-sign $ARNA$ near $-3 = \langle +, + \rangle$.
- Since the *linear term* $\Big[-519.63\Big]h$ is *negative*. we get that Slope-sign $ARNA$ near $-3 = \langle \searrow, \searrow \rangle$
- Since the *quadratic term* $\Big[+81.26\Big]h^2$ is *positive*, we get that Concavity-sign $ARNA$ near $-3 = \langle \cup, \cup \rangle$

7 Quadratic Functions: Global Analysis

=======Begin WORK ZONE=======

The "style" of this chapter is going to be very different from the "style" of the other chapters because we want to take the occasion to give the reader an idea of what happens when a research mathematician is facing a "new problem", that is a problem that no one else has solved before so that s/he cannot just look somewhere or ask someone "how to do it". So, in this chapter, instead of *showing* how to determine the global behavior of a quadratic function $x \xrightarrow{q} q(x) = ax^2 + bx + c$, we will pretend that this is a "research problem".

The first thing we do is to think about the problem itself: What do we mean by "global behavior"? Exactly *what* are we after? The idea is to see what a *precise* statement of the problem might suggest.

One answer might be that "we want to know everything there is to know about a quadratic function". But that is still much too vague to give us any hint as to what to do. Another answer might be "We want to see how the global graph of $x \xrightarrow{q} q(x) = ax^2 + bx + c$ looks?" This is already much better because it specifies the function we want to know about—even if the coefficients a, b, c remain to be specified later. But we really should say what we mean by "global graph", in particular what we want the global graph to show as opposed to what we don't expect the global graph to show.

On the other hand, we care about the global graph only inasmuch as it makes information "graphic" and it is really the information itself that we are after. So, what might this information be that we want? Exactly as with power functions, we will want to know about 0-feature inputs, namely:

- 0-height inputs,
- 0-slope inputs,
- 0-concavity inputs

and about feature-sign change inputs, namely

- height-sign change inputs,
- slope-sign change inputs,
- concavity-sign change inputs.

There still remains a question about what we want to know about these inputs. Do we want to know about:

- The *existence* or *non-existence* of these inputs,

or

- The *location* of these inputs—assuming they exist.

Let us say we want to know everything (But now, as opposed to before, we know exactly what "everything" covers.).

So, now that we know exactly what we want, what do we do to get it? First, though, let us review the equipment we have available:

-

-
-

=======End WORK ZONE=======

In the case of quadratic functions, we will still be able to solve *some* global problems *exactly* but since everything begins to be computationally more complicated, we will deal with only a few types of global problems.

8 The Essential Question

As usual, the first thing we do is to find out if the *offscreen graph* of a *quadratic function* consists of just the *local graph near* ∞ or if it also includes the *local graph near one or more* ∞*-height inputs.*
In other words, given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = a^2x + bx + c$$

we ask the **Essential Question**:

- Do all *bounded inputs* have *bounded outputs*

or

- Are there *bounded inputs* that have ∞-height, that is are there inputs whose nearby inputs have *large* outputs?

Now, given a *bounded* input x, we have that:
- since a is bounded, ax^2 is also bounded
- since b is bounded, bx is also bounded
- c is bounded

and so, altogether, we have that ax^2+bx+c is bounded and that the answer to the **Essential Question** is:

Theorem 8.2 Bounded Height Under a *quadratic* functions, all bounded inputs have *bounded outputs.*

and therefore that

Theorem 8.3 Offscreen Graph The *offscreen graph* of a *quadratic* function consists of just the *local graph near* ∞.

Existence Theorems

The notable inputs are those

- whose existence is forced by the *offscreen graph* which, by the **Bounded Height Theorem** for quadratic functions, consists of only the *local graph near* ∞.
- whose number is limited by the interplay among the three features

Since polynomial functions have no *bounded* ∞-height input, the only way a feature can change sign is near an input where the feature is 0. Thus, with quadratic functions, the feature-change inputs will also be 0-feature inputs.

None of the theorems, though, will indicate *where* the notable inputs are. The **Location Theorems** will be dealt with in the last part of the chapter.

9 Concavity-sign

Given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

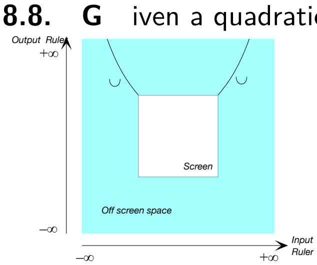
$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = a^2x + bx + c$$

recall that when x is near ∞ the **Concavity-sign Near ∞ Theorem** for quadratic functions says that:

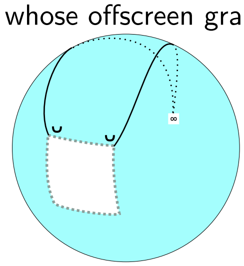
- When a is $+$, Concavity-Sign$|_{x \text{ near } \infty} = (\cup, \cup)$
- When a is $-$, Concavity-Sign$|_{x \text{ near } \infty} = (\cap, \cap)$

1. Since the concavity does *not* changes sign as x goes through ∞ from the left side of ∞ to the right side of ∞, the concavity does not have to change sign as x goes *across the screen* from the left side of ∞ to the right side of ∞ so there does not have to be a *bounded* Concavity-sign change input:

EXAMPLE 8.8. **G** iven a quadratic function whose offscreen graph is

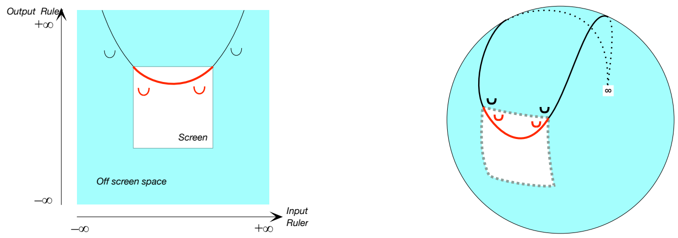


Mercator view



Magellan view

there is no need for a bounded concavity-sign change input, $x_{\text{Concavity-sign change}}$ and therefore we can have global concavity



Mercator view   Magellan view

2. In fact, not only does there not have to be a bounded concavity-sign change input, there *cannot* be a bounded concavity-sign change input since the *local* square coefficient is equal to the *global* square coefficient a and the concavity must therefore be the same everywhere:

> **THEOREM 8.4 Concavity-sign Change Non-Existence** A quadratic function has no bounded *Concavity-sign change* input.

3. Another consequence of the fact that the local concavity does not depend on x_0, and is thus the same everywhere, is that it is a feature of the function $QUADRATIC_{a,b,c}$ itself and so that the function $QUADRATIC_{a,b,c}$ has a **global concavity** specified by the global square coefficient a.

4. Moreover, the concavity cannot be equal to 0 somewhere because the concavity is equal to a everywhere. So, we also have:

> **THEOREM 8.5 0-Concavity Input Non-Existence** A quadratic function has *no* bounded 0-*concavity* input.

10 Slope-sign

Given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

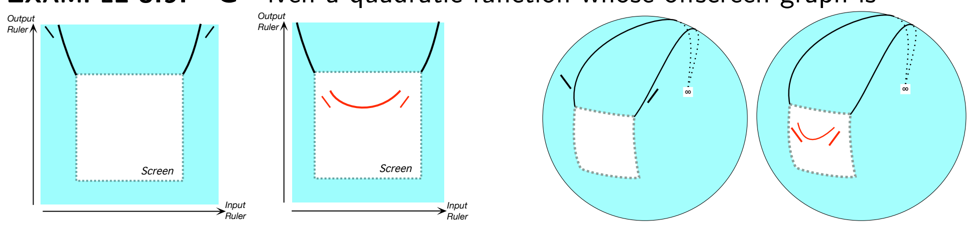
$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = a^2x + bx + c$$

recall that when x is near ∞ the **Slope-sign Near ∞ Theorem** for quadratic functions says that:

- When a is $+$, Slope-Sign$|_{x \text{ near } \infty} = (\diagup, \diagdown)$
- When a is $-$, Slope-Sign$|_{x \text{ near } \infty} = (\diagdown, \diagup)$

1. Since the slope changes sign as x goes from the left side of ∞ to the right side of ∞ *across* ∞ , the slope has also to change sign as x goes from the left side of ∞ to the right side of ∞ *across the screen*. In other words, there has to be a *bounded* slope-sign change input.

EXAMPLE 8.9. **G** iven a quadratic function whose offscreen graph is



Mercator view Magellan view

there has to be a *bounded* slope-sign change input to make up.

So we have

> **THEOREM 8.6 Slope-sign Change Existence** A quadratic function must have at least one bounded Slope-sign change input.

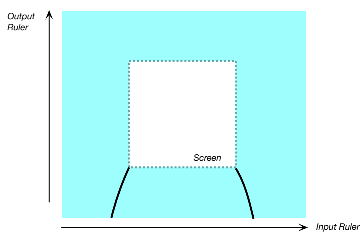
2. On the other hand, a quadratic function can have *at most one* 0-slope input because, if it had more, it would have to have 0-concavity inputs in-between the 0-slope inputs which a quadratic function cannot have. So we have

> **THEOREM 8.7 0-Slope Existence** A quadratic function has exactly one slope-sign change input and it is a 0-slope input: $x_{\text{Slope-sign change}} = x_{\text{0-slope}}$

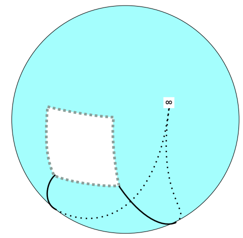
11 Extremum

From the *optimization* viewpoint, a quadratic function has an extreme input, that is an bounded input whose output is larger (or smaller) than the output of nearby inputs

EXAMPLE 8.10. **G** iven a quadratic function whose offscreen graph is

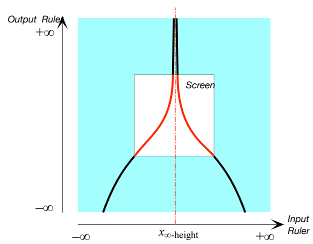


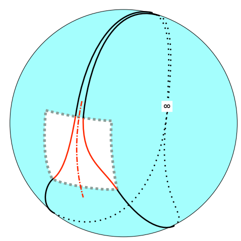
¥



Mercator view ¥ Magellan view

and since quadratic function cannot have an ∞-height input, we cannot have

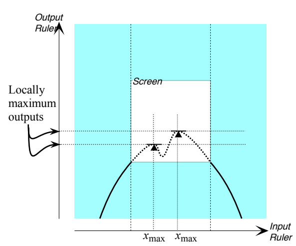




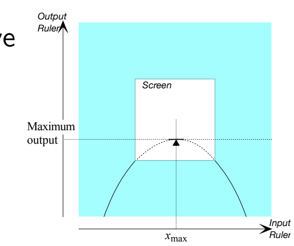
Mercator view ¥ Magellan view

and therefore there has to be at least one input, $x_{\textbf{max}}$, whose output is *maximum*.

But since we cannot have a Concavity-sign change input as in



we must have



THEOREM 8.8 Extremum Existence A quadratic function has a single extremum input

12 Height-sign

Given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

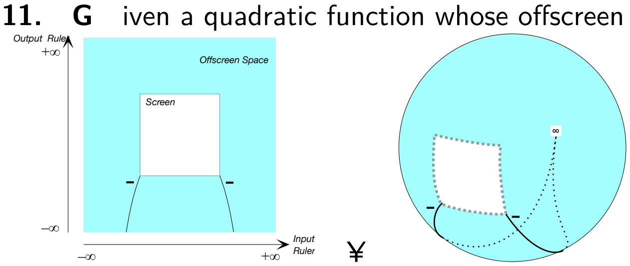
$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = a^2x + bx + c$$

recall that when x is near ∞ the **Height-sign Near ∞ Theorem** for quadratic functions says that:

- When a is $+$, Height-Sign$|_{x \text{ near } \infty} = (+,+)$
- When a is $-$, Height-Sign$|_{x \text{ near } \infty} = (-,-)$

1. Since the height does *not* changes sign as x goes through ∞ from the left side of ∞ to the right side of ∞, the height need not change sign as x goes *across the screen* from the left side of ∞ to the right side of ∞ so there does not have to be at least one *bounded* Height-sign change input:

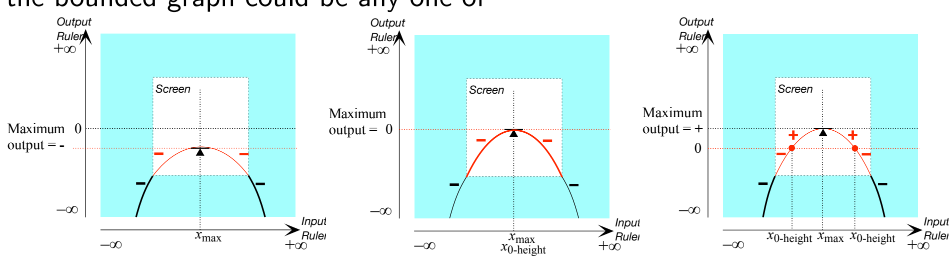
EXAMPLE 8.11. G iven a quadratic function whose offscreen graph is



¥

Mercator view ¥ Magellan view

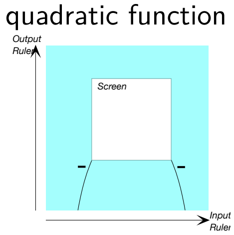
the bounded graph could be any one of



depending on where the *maximum-output* is in relation to the 0-output level line.

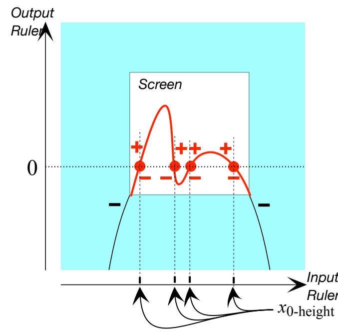
2. On the other hand, a quadratic function can have two 0-height inputs because there *can* be a 0-slope input in-between the two 0-height inputs. However, a quadratic function cannot have more than two 0-height inputs because a quadratic function has only one 0-slope input.

EXAMPLE 8.12. G iven a quadratic function whose offscreen graph is



the reason that we cannot have four 0-height inputs is that that would require three 0-slope inputs in-between (as well as two concavity-sign change inputs which quadratic functions do not have.):

¥



We thus have

THEOREM 8.9 0-Height Existence A quadratic function may have zero, one, or two 0-Height Input(s):
If the *extremum output*

- has the same sign as the sign of the outputs near ∞, then there is *no* 0-height input
- is 0, then there is *one* 0-height input
- has the sign opposite from the sign of the outputs near ∞, then there are *two* 0-height input

13 Bounded Graph

Here too, there are two ways to look at the shape of the bounded graph.

1. As a consequence of the **Bounded Height Theorem** for *quadratic* functions, the offscreen graph consists only of the local graph near ∞ and we can obtain the *forced bounded graph* by extrapolating smoothly the local graph near ∞.

EXAMPLE 8.13. **G** iven the quadratic function $LAON_{+34.54,-40.38,-94.21}$ whose input-output rule is

$$x \xrightarrow{LAON_{+34.54,-40.38,-94.21}} LAON_{+34.54,-40.38,-94.21}(x) = +34.54x^{+2} - 40.38x - 94.21$$

find its forced bounded graph:

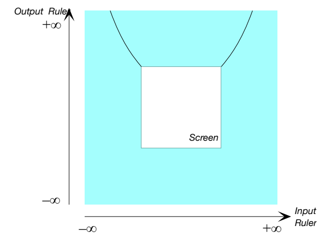
i. We normalize the global input-output rule

$$x \xrightarrow{LAON} LAON(x) = +x^{+2} - 40.38x - 94.21$$

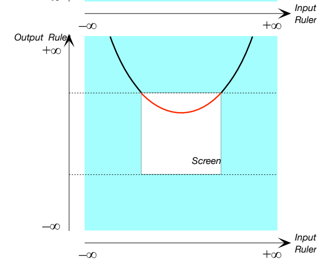
ii. We get the approximate local input-output rule near ∞

$$x \xrightarrow{LAON} LAON(x) = +x^{+2} + [...]$$

iii. We get the local graph near ∞ ¥

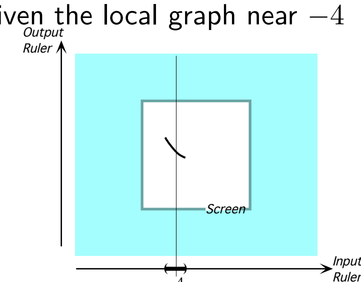


iv. We extrapolate smoothly ¥

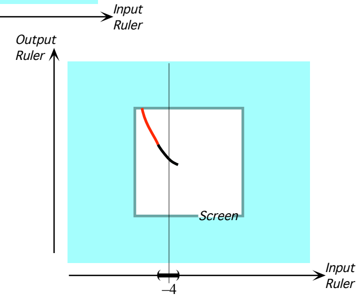


2. In the case of *quadratic functions*, it happens that we can also obtain the *bounded graph* by extrapolating the local graph near a bounded input:

EXAMPLE 8.14. G iven the local graph near -4

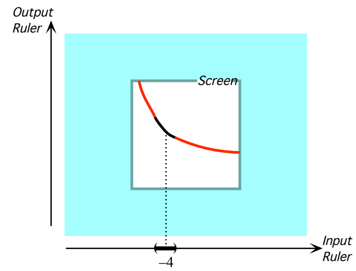


On the left, since the concavity-sign cannot change, we can extrapolate only one way: ¥

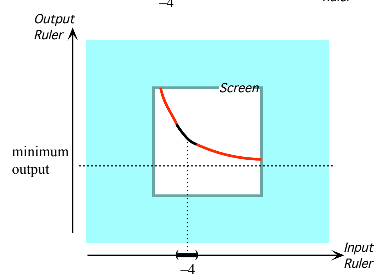


On the right, though, there are several possibilities up front but only one fits what we already know:

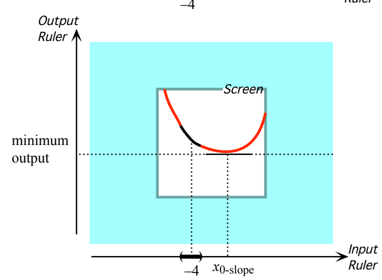
With this extrapolation, we don't have a 0-slope input : ¥



With this extrapolation we do have a 0-slope input but it is at ∞ ¥



But this extrapolation fits all that we already know ¥



Location Theorems

Previously, we only established the *existence* of certain notable features of quadratic functions and this investigation was based on *graphic* considerations. Here we will investigate the *location* of the inputs where these notable features occur and this investigation will be based on *input-output rule* considerations.

14 0-Concavity Location

We saw earlier that quadratic functions cannot have a 0-concavity input. On the other hand, since the concavity is the same everywhere, it is a global feature of the function itself and we have:

THEOREM 8.10 Global Concavity-sign Given the quadratic function $QUADRATIC_{a,b,c}$,

- When a is *positive*, Concavity-sign $QUADRATIC = \cup$.
- When a is *negative*, Concavity-sign $QUADRATIC = \cap$

This is just like affine function having a *global slope*.

15 0-Slope Location

Given a quadratic function, the global problem of *locating* an input where the local slope is 0 is still fairly simple.

More precisely, given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = ax^2 + bx + c$$

since the *slope* near x_0 is the local linear coefficient $2ax_0 + b$, in order to find the input(s) where the local slope is 0, we just need to solve the equation

$$2ax + b = 0$$

which is an affine equation that we solve by reducing it to a basic equation:

$$\begin{aligned} 2ax + b -b &= 0 \text{ -b} \\ 2ax &= -b \\ \frac{2ax}{2a} &= \frac{-b}{2a} \\ x &= \frac{-b}{2a} \end{aligned}$$

So, we have:

THEOREM 8.11 0-slope Location For any quadratic function $QUADRATIC_{a,b,c}$,

$$x_{0-slope} = \frac{-b}{2a}$$

In fact, we also have:

THEOREM 8.12 Global Slope-sign Given a quadratic function $QUADRATIC_{a,b,c}$,

- When a is positive,

$$\text{Slope-sign } QUADRATIC|_{\text{Everywhere} < \frac{-b}{2a}} = (\searrow, \searrow)$$

$$\text{Slope-sign } QUADRATIC|_{\frac{-b}{2a}} = (\searrow, \nearrow)$$
$$\text{Slope-sign } QUADRATIC|_{\text{Everywhere} > \frac{-b}{2a}} = (\nearrow, \nearrow)$$

- When a is negative,
$$\text{Slope-sign } QUADRATIC|_{\text{Everywhere} < \frac{-b}{2a}} = (\nearrow, \nearrow)$$
$$\text{Slope-sign } QUADRATIC|_{\frac{-b}{2a}} = (\nearrow, \searrow)$$
$$\text{Slope-sign } QUADRATIC|_{\text{Everywhere} > \frac{-b}{2a}} = (\searrow, \searrow)$$

The case is easily made by testing the corresponding inequations near ∞.

16 Extremum Location

From the **Extremum Existence Theorem**, we know that

$$x_{\text{extremum}} = x_{\text{0-slope}}$$

and so we have that

$$x_{\text{extremum}} = \frac{-b}{2a}$$

We now want to compute the extremum *output* which is the output for $x_{\text{0-slope}}$:

$$\begin{aligned}
QUADRATIC(x_{\text{0-slope}}) &= ax_{\text{0-slope}}^2 + bx_{\text{0-slope}} + c \\
&= a\left(\frac{-b}{2a}\right)^2 + b\left(\frac{-b}{2a}\right) + c \\
&= a\left(\frac{(-b)^2}{(2a)^2}\right) + b\left(\frac{-b}{2a}\right) + c \\
&= a\left(\frac{b^2}{4a^2}\right) + b\left(\frac{-b}{2a}\right) + c \\
&= \frac{ab^2}{4a^2} + b\left(\frac{-b}{2a}\right) + c \\
&= \frac{\not{a}b^2}{4\not{a}a} + b\left(\frac{-b}{2a}\right) + c \\
&= \frac{b^2}{4a} + b\left(\frac{-b}{2a}\right) + c \\
&= \frac{b^2}{4a} + \frac{-2\cdot b^2}{2\cdot 2a} + c
\end{aligned}$$

discriminant

$$= \frac{b^2}{4a} + \frac{-2b^2}{4a} + c$$
$$= \frac{-b^2}{4a} + c$$
$$= \frac{-b^2}{4a} + \frac{4a \cdot c}{4a}$$
$$= \frac{-b^2 + 4ac}{4a}$$

It is standard to call the quantity $b^2 - 4ac$, that is the opposite of the above numerator, the **discriminant** of the function $QUADRATIC_{a,b,c}$ and we will write $\textbf{Discriminant}_{QUADRATIC}$
So we have that the extremum output

$$QUADRATIC(x_{\text{extremum}}) = QUADRATIC(x_{\text{0-slope}}) = \frac{-\text{Discriminant}_{QUADRATIC}}{4a}$$

Altogether then, we have

THEOREM 8.13 G iven a quadratic function $QUADRATIC_{a,b,c}$, the extremum input is

$$x_{\text{extremum}} = x_{\text{0-slope}} = \frac{-b}{2a}$$

and the extremum output is

$$QUADRATIC(x_{\text{extremum}}) = \frac{-b^2 + 4ac}{4a} = \frac{-Discriminant_{QUADRATIC}}{4a}$$

17 0-Height Location

Given a quadratic function, the global problem of *locating* a given local height is the problem of locating the input(s), if any, whose output is equal to the given height.

More precisely, given the quadratic function $QUADRATIC_{a,b,c}$, that is the function specified by the global input-output rule

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = ax^2 + bx + c$$

and given the local height H_0, what we are looking for are the input(s), if any, whose output is equal to H_0, that is:

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = H_0$$

In other words, we must solve the equation

$$ax^2 + bx + c = H_0$$

This is called a **quadratic equation**. Since we are looking for the 0-height inputs, we let H_0 be 0 and we will want to solve the equation

quadratic equation

$$ax^2 + bx + c = 0$$

Solving a quadratic equation is quite a bit more complicated than solving an affine equation because we cannot reduce a quadratic equation to a basic equation the way we reduce an affine equation to a basic equation.

The reason is that affine equations have *two* terms and the = sign has *two* sides so that we could *separate* the terms by having an x-term on the left side of the = sign and a constant term on the right side of the = sign which gave us a basic equation.

However, we cannot *separate* the terms in a quadratic equation because the output $QUADRATIC(x)$ has *three* terms while the = sign has only *two* sides.

This, though, may have something to do with the fact that inputs are counted from the 0 on the ruler which can be anywhere in relation to the *global graph* of the function, rather than from an input which is meaningful for the global graph of that function.

What we will do then is to try to use, instead of the inputs themselves, the *location* of the inputs relative to an input that is meaningful for the function at hand and the obvious thing is to try is $x_{\text{0-slope}}$ and so we will try to use:

$$u = x - x_{\text{0-slope}}$$

$$x = x_{\text{0-slope}} + u$$

and therefore, instead of using the global input-output rule

$$x \xrightarrow{QUADRATIC} QUADRATIC(x) = ax^2 + bx + c$$

we will use the global input-ouput rule

$$x|_{x \leftarrow x_{\text{0-slope}}+u} \xrightarrow{QUADRATIC} QUADRATIC(x)\,|_{x \leftarrow x_{\text{0-slope}}+u} = ax^2 + bx + c\Big|_{x \leftarrow x_{\text{0-slope}}+u}$$

that is

$$u \xrightarrow{QUADRATIC_{(x_{\text{0-slope}})}} QUADRATIC(x_{\text{0-slope}} + u)$$
$$= \Big[a\Big]u^2 + \Big[2ax_{\text{0-slope}} + b\Big]u + \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]$$

By the way, note that we will continue to count the *outputs* from the 0 on the output ruler. (Some people don't and prefer to count the outputs from $QUADRATIC(x_{\text{0-slope}})$.) But since $x_{\text{0-slope}} = \frac{-b}{2a}$, this reduces to

$$u \xrightarrow{QUADRATIC_{(x_{\text{0-slope}})}} QUADRATIC(x_{\text{0-slope}} + u)$$
$$= \Big[a\Big]u^2 + \Big[0\Big]u + \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]$$

that is to only two terms

$$= \Big[a\Big]u^2 + \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]$$

and the equation we want to solve, then, is

$$\Big[a\Big]u^2 + \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big] = H_0$$

that is

$$\Big[a\Big]u^2 = H_0 - \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]$$

that is

$$u^2 = \frac{H_0 - \Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]}{a}$$

in which everything on the right-hand side is known so that we have *separated* the known from the unknown. Since we are trying to locate the 0-height inputs, we let $H_0 = 0$.

In that case, the equation reduces to

$$\begin{aligned} u^2 &= \frac{-\Big[ax^2_{\text{0-slope}} + bx_{\text{0-slope}} + c\Big]}{a} \\ &= \frac{-QUADRATIC(x_{\text{extremum}})}{a} \end{aligned}$$

and, using the Extremum Location Theorem,

$$\begin{aligned} &= \frac{-\frac{-\text{Discriminant}_{QUADRATIC}}{4a}}{a} \\ &= \frac{\text{Discriminant}_{QUADRATIC}}{4a^2} \end{aligned}$$

Altogether then, instead of the original equation

$$ax^2 + bx + c = 0$$

we have the rather nice (nicer?) equation

$$\boxed{u^2 = \frac{\text{Discriminant}_{QUADRATIC}}{4a^2}}$$

Now, of course, whether or not we can solve depends on whether or not the right hand side is positive and since the denominator is a square, and therefore always positive, whether or not we can solve depends only on the sign of $\text{Disc}_{QUADRATIC}$ (hence the name “discriminant”):

- ▶ If $\text{Disc}_{QUADRATIC}$ is *negative*, the equation has *no* solution,
- ▶ If $\text{Disc}_{QUADRATIC}$ is 0, the equation has *one* solution, namely 0,
- ▶ If $\text{Disc}_{QUADRATIC}$ is *positive*, the equation has *two* solutions, namely

- $u = -\frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$
- $u = +\frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$

de-locate

This, of course, is hardly surprising inasmuch as the discriminant is intimately tied with the extremum output and thus this theorem fits very well with the 0**-height Existence Theorem**. It remains only to **de-locate**, that is to return to the input x. For that, we need only use the fact that

$$u = x - x_{\text{0-slope}}$$

to get

- $x - x_{\text{0-slope}} = -\frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$
- $x - x_{\text{0-slope}} = +\frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$

that is

- $x = x_{\text{0-slope}} - \frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$
- $x = x_{\text{0-slope}} + \frac{\sqrt{\text{Disc}_{QUADRATIC}}}{2a}$

and thus the celebrated "quadratic formula":

- $x = x_{\text{0-slope}} - \frac{\sqrt{b^2-4ac}}{2a}$
- $x = x_{\text{0-slope}} + \frac{\sqrt{b^2-4ac}}{2a}$

which, by the way, shows that, when they exist, the two 0-height inputs are symmetrical with respect to $x_{\text{0-slope}}$ Altogether, then, we have

THEOREM 8.14 0-height Location For any quadratic function $QUADRATIC_{a,b,d}$,

- ▶ If $\text{Disc}_{QUADRATIC}$ is *negative*, $QUADRATIC$ has *no* 0-height input,
- ▶ If $\text{Disc}_{QUADRATIC}$ is 0, $QUADRATIC$ has *one* 0-height input, namely $\frac{-b}{2a}$,
- ▶ If $\text{Disc}_{QUADRATIC}$ is *positive*, $QUADRATIC$ has *two* solutions, namely
  - $\frac{-b}{2a} - \frac{\sqrt{b^2-4ac}}{2a}$
  - $\frac{-b}{2a} + \frac{\sqrt{b^2-4ac}}{2a}$

Finally, here are a couple of examples.

Example 8.15. **T** o find the 0-height inputs of the quadratic function specified by the global input-output rule

$$x \xrightarrow{Rick} Rick(x) = +4x^2 - 24x + 7$$

we can proceed as follows:

i. Either we remember that $x_{0-slope} = \frac{-b}{2a}$ so that we get $x_{0-slope} = \frac{+12}{2(+4)} = +3$, or, if worse comes to worst, we look for the 0-slope input by localizing at an undisclosed input x_0 and then setting the coefficient of u equal to 0 to get $x_{0-slope}$.

ii. Then, we get the u-equation by setting $x = x_{0-slope} + u$, that is, here, by setting $x = +3 + u$:

$$\begin{aligned} +3+u \xrightarrow{Rick} Rick(x)|_{\text{when } x=+3+u} &= +4x^2 - 24x + 7\Big|_{\text{when } x=+3+u} \\ &= +4\,[+3+u]^2 - 24\,[+3+u] + 7 \\ &= +4\left[+9 + 6u + u^2\right] - 24\,[+3+u] + 7 \\ &= +36 + 24u + 4u^2 - 72 - 24u + 7 \\ &= -29 + 4u^2 \end{aligned}$$

iii. We now solve the u-equation

$$\begin{aligned} -29 + 4u^2 &= 0 \\ +4u^2 &= +29 \\ u^2 &= \frac{+29}{+4} \\ u^2 &= +7.25 \end{aligned}$$

and so we have:

- ▶ $u_{0-output} = +\sqrt{+7.25} = +2.69 + [...]$

and

- ▶ $u_{0-output} = -\sqrt{+7.25} = -2.69 + [...]$

and therefore

- ▶ $x_{0-output} = +3 + 2.693 + [...] = +5.693 + [...]$

and

- ▶ $x_{0-output} = +3 - 2.693 + [...] = +0.307 + [...]$

Alternatively, if we remember the **0-height Theorem**, then we can proceed by first computing the discriminant and

Example 8.16. **W** e look at the same equation but assume that we remember the **0-height Theorem**

$$x \xrightarrow{Rick} Rick(x) = +4x^2 - 24x + 7$$

that is:

$$\begin{aligned}\text{Discriminant } Rick &= (-24)^2 - 4(+4)(+7)\\ &= +576 - 112\\ &= +464\end{aligned}$$

And since the discriminant is positive, we have

$$\begin{aligned}x_{0-output} &= x_{0-slope} + \frac{\sqrt{\text{Discriminant}}}{2a}\\ &= \frac{+24}{2(+4)} + \frac{\sqrt{+464}}{2(+4)}\\ &= \frac{+24}{+8} + \frac{21.541 + [...]}{+8}\\ &= \frac{45.541 + [...]}{+8}\\ &= +5.693 + [...]\end{aligned}$$

and similarly

$$\begin{aligned}x_{0-output} &= x_{0-slope} - \frac{\sqrt{\text{Discriminant}}}{2a}\\ &= \frac{+24}{2(+4)} - \frac{\sqrt{+464}}{2(+4)}\\ &= \frac{+24}{+8} - \frac{21.541 + [...]}{+8}\\ &= \frac{2.460 + [...]}{+8}\\ &= +0.307 + [...]\end{aligned}$$

Either way, the reader should check that, indeed,

$$+5.693 \xrightarrow{Rick} 0 + [...]$$

and

$$+0.307 \xrightarrow{Rick} 0 + [...]$$

As a consequence of the 0**-height Location Theorem**, we have:

THEOREM 8.15 Global Height-sign For any quadratic function $QUADRATIC_{a,b,c}$, $\textit{Height-sign } QUADRATIC = (Sign\ a, Sign\ a)$

everywhere except, when $Disc_{QUADRATIC}$ is *positive*, between the two $x_{0\text{-}height}$ inputs where *Height-sign* $QUADRATIC = (-Sign\ a, -Sign\ a)$

As a result, when looking for the inputs for which the output has a given sign, we have two approaches:

i. We can solve the associate equation, one way or the other, and then test each one of the sections determined by the 0-height input(s), if any.

Example 8.17. T o solve the *inequation* $-3x^2 + tx - 11 < 0$, we can begin by looking for its *boundary inputs* by solving the *associated equation* $-3x^2 + tx - 11 = 0$ and then test the resulting intervals.

ii. We can use the **Global Height-sign Theorem**.

=========OK SO FAR=========

Quadratic␣function
cubic␣coefficient
quadratic␣coefficient
linear␣coefficient
constant␣coefficient

Chapter 9

Cubic Functions

Quadratic functions are specified by global input-output rules like the generic global input-output rule:

$$x \xrightarrow{CUBIC} CUBIC(x) = \underbrace{ax^{+3} \oplus bx^{+2} \oplus cx^{+1} \oplus dx^{0}}_{\text{output-specifying code}}$$

which we usually write

$$= \underbrace{ax^3 + bx^2 + cx + d}_{\text{output-specifying code}}$$

where a, called the **cubic coefficient**, b, called the **quadratic coefficient**, c, called the **linear coefficient**, and d, called the **constant coefficient**, are the *bounded* numbers that specify the function $CUBIC$.

Example 9.1. The cubic function $TINA$ specified by the cubic coefficient $+72.55$, the quadratic coefficient -23.04, the linear coefficient -17.39 and the constant coefficient $+5.84$ is the function specified by the global input-output rule

$$x \xrightarrow{RINA} TINA(x) = \underbrace{-72.55}_{\text{cubic coeff.}}\ x^3\ \underbrace{-23.04}_{\text{quadratic coeff.}}\ x^2\ \underbrace{-17.39}_{\text{linear coeff.}}\ x\ \underbrace{+5.84}_{\text{constant coeff.}}$$

It is worth noting again that

term
cubic term
quadratic term
linear term
constant term
quadratic␣part

CAUTIONARY NOTE 9.1 The terms in the global input output rule *need not* be written in order of *descending* exponent. This is just a habit we have.

EXAMPLE 9.2. The function specified by the global input-output rule

$$x \xrightarrow{DIDI} DIDI(x) = -12.06x^3 + 21.03x^2 - 31.39x + 5.34$$

could equally well be specified by the global input-output rule

$$x \xrightarrow{DIDI} DIDI(x) = +5.34 + 21.03x^2 - 31.39x - 12.06x^3$$

or by the global input-output rule

$$x \xrightarrow{DIDI} DIDI(x) = -31.39x + 5.34 - 12.06x^3 + 21.03x^2$$

We now introduce some standard terminology to help us describe very precisely what we will be doing. The output-specifying code of the affine function specified by

$$x \xrightarrow{AFFINE} CUBIC(x) = \underbrace{ax^3 + bx^2 + cx + d}_{\text{output-specifying code}}$$

consists of four **terms**:

- ax^3 which is called the **cubic term**.
- bx^2 which is called the **quadratic term**.
- cx which is called the **linear term**.
- d which is called the **constant term**,

and there is of course also

- $bx^2 + cx + d$ which is called the **quadratic part**

EXAMPLE 9.3. The output-specifying code of the function specified by the global input-output rule

$$x \xrightarrow{TINA} TINA(x) = \underbrace{-71.41}_{\text{cubic coeff.}} x^3 \quad \underbrace{-23.04}_{\text{quadratic coeff.}} x^2 \quad \underbrace{-31.39}_{\text{linear coeff.}} x \quad \underbrace{+5.84}_{\text{constant coeff.}}$$

consists of four terms:

$$= \underbrace{-71.41x^3}_{\text{cubic term}} \quad \underbrace{-23.04x^2}_{\text{quadratic term}} \quad \underbrace{-31.39x}_{\text{linear term}} \quad \underbrace{+5.34}_{\text{constant term}}$$

LANGUAGE NOTE 9.1 Whether we look upon d as the constant *coefficient*, that is as the *coefficient* of x^0 in the constant *term* dx^0 or as the constant *term* dx^0 itself with the power x^0 "going without saying" will be clear from the context.

1 Output *at* x_0

Remember from section 1 that x_0 is a *generic given input*, that is that x_0 is a *bounded* input that has been *given* but whose identity remains *undisclosed* for the time being.

PROCEDURE 9.1

i. *Declare* that x is to be replaced by x_0

$$x\Big|_{x\leftarrow x_0} \xrightarrow{CUBIC} CUBIC(x)\Big|_{x\leftarrow x_0} = ax^3 + bx^2 + cx + d\Big|_{x\leftarrow x_0}$$

which gives:

$$x_0 \xrightarrow{CUBIC} CUBIC(x_0) = \underbrace{a{x_0}^3 + b{x_0}^2 + c x_0 + d}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into an output *number*:

$$= ax_0^3 + bx_0^2 + cx_0 + d$$

which gives the input-output pair

$$\Big(x_0, ax_0^3 + bx_0^2 + cx_0 + d\Big)$$

DEMO 9.1

i. We *declare* that x is to be replaced by -3

$$x\Big|_{x\leftarrow -3} \xrightarrow{ARIA} ARIA(x)\Big|_{x\leftarrow -3} = +17.52x^3 + 21.03x^2 - 32.67x + 71.07\Big|_{x\leftarrow -3}$$

which gives

$$-3 \xrightarrow{ARIA} ARIA(-3) = \underbrace{+17.52(-3)^3 + 21.03(-3)^2 - 32.67(-3) + 71.07}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into an output *number*:

$$= -473.04 \oplus +189.26 \oplus +98.01 \oplus +71.07$$
$$= -114.7$$

which gives the *input-output pair*

$$(-3, -114.7)$$

However, as already discussed in **?? ??** and as has already been the case with *monomial* functions, *affine* functions and *quadratic* functions, instead of getting the output *number* returned by a quadratic function *at* a given input, we will usually want *all* the outputs returned by the quadratic function for inputs *near* that given input. So, instead of getting the single *input-output* *pair at* the given input, we will get the *local input-output rule* with which to get *all* the input-output pairs *near* the given input.

2 Output *near* ∞

As already discussed in subsection 8.2 Output *near* ∞ and in section 3 Output *near* ∞, in order to input a neighborhood of ∞, we will *declare* that "x is near ∞" but write only x after that. This, again, is extremely dangerous as it is easy to forget that what we write may be TRUE *only* because x has been declared to be near ∞.

1. We will *execute* the output-specifying code, namely ax^3+bx^2+cx+d, into an *output jet*, that is with the terms in *descending* order of sizes, which, since here x is *large*, means that here the powers of x must be in *descending* order of exponents. We will then have the *local input-output rule near* ∞:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \underbrace{\text{Powers of } x \text{ in } \textit{descending order of exponents}}_{\text{output jet near } \infty}$$

EXAMPLE 9.4. Given the function specified by the global input-output rule

$$x \xrightarrow{TIBA} TIBA(x) = -61.03 + 37.81x^3 - 82.47x + 45.03x^2$$

To get the output jet near ∞, we first need to get the *order of sizes*.

i. -61.03 is *bounded*

ii. -82.47 is *bounded* and x is *large*. So, since *bounded* $\cdot$ *large* $=$ *large*, $-82.47 \cdot x$ is *large*

iii. $+45.03$ is *bounded* and x is *large*. So, since *bounded* $\cdot$ *large* $=$ *large*, $+45.03 \cdot x$ is *large* too. But *large* $\cdot$ *large* is larger in size than *large* so $+45.03 \cdot x^2$ is even larger than $-82.47 \cdot x$

iv. $+37.81$ is *bounded* and x is *large*. So, since *bounded* $\cdot$ *large* $=$ *large*, $+37.81 \cdot x$ is *large* too. But *large* $\cdot$ *large* $\cdot$ *large* is larger in size than *large* $\cdot$ *large* so $+37.81 \cdot x^3$ is even larger than $+45.03 \cdot x^2$

So, in the output jet near ∞, $+37.81x^3$ must come first, $+45.03x^2$ must come second, $-82.47x$ comes third and -61.03 comes fourth

Then, we get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{RIBA} TIBA(x) = \underbrace{+37.81x^3 + 45.03x^2 - 82.47x - 61.03}_{\text{output jet near } \infty}$$

2. Altogether, then:

PROCEDURE 9.2

i. *Declare* that x is near ∞

$$x\Big|_{x \text{ near } \infty} \xrightarrow{CUBIC} CUBIC(x)\Big|_{x \text{ near } \infty} = ax^3 + bx^2 + cx + d\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \underbrace{ax^3 + bx^2 + cx + d}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into a *jet* near ∞

$$= \underbrace{[a]\, x^3 \oplus [a]\, x^2 \oplus [b]\, x \oplus [c]}_{\text{output jet near } \infty}$$

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \underbrace{[a]\, x^3 \oplus [a]\, x^2 \oplus [b]\, x \oplus [c]}_{\text{output jet near } \infty}$$

(*Here* the jet near ∞ looks the same as the given global input-output rule but that is only because the output-specifying code *happened* to be written in *descending* order of exponents.)

DEMO 9.2

i. We *declare* that x is near ∞

$$x\Big|_{x \text{ near } \infty} \xrightarrow{DINA} DINA(x)\Big|_{x \text{ near } \infty} = -61.03 + 37.81x^3 + 51.32x^2 - 82.47x\Big|_{x \text{ near } \infty}$$

which gives:

$$x \text{ near } \infty \xrightarrow{DINA} DINA(x) = \underbrace{-61.03 + 37.81\, x^3 + 51.32\, x^2 - 82.47\, x}_{\text{output-specifying code}}$$

ii. We *execute* the output-specifying code into a *jet* near ∞:

$$= [+37.81]\, x^3 \oplus [+51.32]\, x^2 \oplus [-82.47]\, x \oplus [-61.03]$$

which gives the *local input-output rule* near ∞:

$$x \text{ near } \infty \xrightarrow{DINA} DINA(x) = \underbrace{[+37.81]\, x^3 \oplus [+51.32]\, x^2 \oplus [-82.47]\, x \oplus [-61.03]}_{\text{output jet near } \infty}$$

(Here the jet near ∞ does *not* look the same as the *global* input-output rule because the output-specifying code happened *not* to be in descending order of exponents.)

3. The reason we use *jets* here is that the term *largest in size* is the *first* term so that to *approximate* the output we need only write the *first* term in the jet and just replace the remaining terms by [...] which stands for "something too small to matter here". In other words,

THEOREM 9.1 Approximate output near ∞. For *cubic* functions, the term in the jet that contributes most to the output near ∞ is the *highest degree term* in the output jet near ∞:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \left[a \right] x^3 + [...]$$

EXAMPLE 9.5. Given the function specified by the global input-output rule

$$x \xrightarrow{DINA} DINA(x) = -61.03 + 37.81x^3 + 51.32x^2 - 82.47x$$

near ∞ we will often just use the *approximation*

$$x \text{ near } \infty \xrightarrow{KINA} KINA(x) = \left[+37.81 \right] x^3 \oplus [...]$$

3 Output *near* x_0

We now deal with the output of the neighborhood of some *given* bounded input x_0.

1. In order to input a neighborhood of a given input x_0 we will declare that $x \leftarrow x_0 \oplus h$ that is that x is to be replaced by $x_0 \oplus h$. As a result, we will have to compute $(x_0 \oplus h)^2$ for which we will have to use an *addition formula* from ALGEBRA, namely **??** in **??** on page **??**.

2. We can then *execute* the input-output specifying phrase into a *jet* that is with the terms in *descending order of sizes* which here, since h is *small*, means that the powers of h will have to be in *ascending* order of exponents. We will then have the local input-output rule near the given input:

$$x_0 \oplus h \xrightarrow{CUBIC} CUBIC(x_0 \oplus h) = \underbrace{\text{Powers of } h \text{ in } \textit{ascending order of exponents}}_{\text{output jet near } \infty}$$

We will therefore use:

PROCEDURE 9.3

i. *Declare* that x is to be replaced by $x_0 + h$

$$x \Big|_{x \leftarrow x_0+h} \xrightarrow{CUBIC} CUBIC(x) \Big|_{x \leftarrow x_0+h} = ax^3 + bx^2 + cx + d \Big|_{x \leftarrow x_0+h}$$

which gives:

jet near x_0

$$x_0+h \xrightarrow{CUBIC} CUBIC(x_0+h) = \underbrace{a(x_0+h)^3+b(x_0+h)^2+c(x_0+h)+d}_{\text{output-specifying code}}$$

ii. *Execute* the output-specifying code into a *jet* near x_0:

$$= a\left(x_0^3+3x_0^2h+3x_0h^2+h^3\right)+b\left(x_0^2+2x_0h+h^2\right)+c\,(x_0+h)+d$$

$$= ax_0^3 \oplus 3ax_0^2\,h \oplus 3ax_0\,h^2 \oplus a\,h^3$$

$$\oplus\ bx_0^2 \oplus 2bx_0\,h \oplus b\,h^2$$

$$\oplus\ cx_0 \oplus c\,h$$

$$\oplus\ d$$

$$= \underbrace{\Big[ax_0^3+bx_0^2+cx_0+d\Big] \oplus \Big[3ax_0^2+2bx_0+c\Big]h \oplus \Big[3ax_0+b\Big]h^2 \oplus \Big[a\Big]h^3}_{\text{output jet near } x_0}$$

which gives the *local input-output rule* near x_0:

$$x_0+h \xrightarrow{CUBIC} CUBIC(x_0+h)$$

$$= \underbrace{\Big[ax_0^3+bx_0^2+cx_0+d\Big] \oplus \Big[3ax_0^2+2bx_0+c\Big]h \oplus \Big[3ax_0+b\Big]h^2 \oplus \Big[a\Big]h^3}_{\text{output jet near } x_0}$$

Demo 9.3

i. We declare that x is to be replaced by $-3+h$

$$x\,\Big|_{x\leftarrow -3+h} \xrightarrow{ARBA} ARBA(x)\,\Big|_{x\leftarrow -3+h} = -32.67x-31.18x^3+71.07+81.26x^2\,\Big|_{x\leftarrow -3+h}$$

which gives

$$-3+h \xrightarrow{ARBA} ARBA(-3+h) = \underbrace{-32.67(-3+h)-31.18(-3+h)^3+71.07+81.26(-3+h)^2}_{\text{output specifying code}}$$

ii. We *execute* the output-specifying code into a *jet* near -3:

$$= -32.67(-3+h)-31.18\left((-3)^3+3(-3)^2h+3(-3)h^2+h^3\right)+71.07+81.26\left((-3)^2+2(-3)h+h^2\right)$$

$$= -32.67(-3)-32.67h$$

$$-31.18(-3)^3-31.18\cdot 3(-3)^2h-31.18\cdot 3(-3)h^2-31.18h^3$$

$$+71.07$$

$$+81.26(-3)^2+81.26\cdot 2(-3)h+81.26h^2$$

$$= \ +98.01 \oplus -32.67\,h$$

$$\oplus \; +841.86 \; \oplus \; -841.86\,h \oplus +280.62\,h^2 \oplus -31.18\,h^3$$
$$\oplus \; +71.07$$
$$\oplus \; +731.34 \; \oplus \; -487.56\,h \; \oplus \; +81.26\,h^2$$
$$= \Big[+98.01 + 841.86 + 71.07 + 731.34\Big]$$
$$\oplus \Big[-32.67 - 841.86 - 487.56\Big]h$$
$$\oplus \Big[+280.62 + 81.26\Big]h^2$$
$$\oplus \Big[-31.18\Big]h^3$$
$$= \underbrace{\Big[+1742.28\Big] \oplus \Big[-1362.09\Big]h \oplus \Big[+361.88\Big]h^2 \oplus \Big[+81.26\Big]h^3}_{\text{output jet near } -3}$$

which gives the *local input-output rule* near -3:

$$-3+h \xrightarrow{ARNA} ARBA(-3+h) = \underbrace{\Big[+1742.28\Big] \oplus \Big[-1362.09\Big]h \oplus \Big[+361.88\Big]h^2 \oplus \Big[+81.}_{\text{output jet near } -3}$$

3. When all we want is a feature-sign, though, the above procedure is very inefficient and we will then use the following procedure based directly on the fact that a *cubic function* is the addition of:

- a *cube function*, (See DEFINITION 6.5 on page 264)
- a *square function*, (See DEFINITION 6.2 on page 262)
- a *linear function*, (See **??** on **??**.)
- a *constant function*. (See **??** on **??**.)

that is:

$$x \xrightarrow{CUBIC} CUBIC(x) = \underbrace{ax^3}_{\text{cube}} \oplus \underbrace{bx^2}_{\text{square}} \oplus \underbrace{cx}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

We declare that x is near x_0 that is that x must be replaced by $x_0 + h$:

$$x \xrightarrow{CUBIC} CUBIC(x) = \underbrace{a\,(x_0+h)^3}_{\text{cube}} \oplus \underbrace{b\,(x_0+h)^2}_{\text{square}} \oplus \underbrace{c\,(x_0+h)}_{\text{linear}} \oplus \underbrace{d}_{\text{constant}}$$

The output of the local input-output rule near x_0 will have to be a *jet*:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = \Big[\quad\Big] \oplus \Big[\quad\Big]h \oplus \Big[\quad\Big]h^2 \oplus \Big[\quad\Big]h^3$$

and we want to be able to get any one of the coefficients of the output jet without having to compute any of the other coefficients. So, what we will do is to get the contribution of each monomial function to the term we want. This requires us to have the *addition formulas* at our finger tips:

a.

$$(x_0+h)^2 = x_0^2 + 2x_0h + h^2$$ (See **??** on page 519)

b.

$$(x_0+h)^3 = x_0^3 + 3x_0^2h + 3x_0h^2 + h^3$$ (See **??** on **??**)

More precisely,

i. If we want the *coefficient* of h^0 in the output jet:

- The cube function contributes ax_0^3
- The square function contributes bx_0^2
- The linear function contributes cx_0
- The constant function contributes d

so we have:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = \left[ax_0^3 + bx_0^2 + cx_0 + d\right] \oplus \left[\quad\right]h \oplus \left[\quad\right]h^2 \oplus \left[\quad\right]h^3$$

ii. If we want the *coefficient* of h^1 in the output jet:

- The cube function contributes $3bx_0^2$
- The square function contributes $2bx_0$
- The linear function contributes c
- The constant function contributes nothing

so we have:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = \left[\quad\right] \oplus \left[3bx_0^2 + 2bx_0 + c\right]h \oplus \left[\quad\right]h^2 \oplus \left[\quad\right]h^3$$

iii. If we want the *coefficient* of h^2 in the output jet:

- The cube function contributes $3bx_0$
- The square function contributes c
- The linear function contributes nothing
- The constant function contributes nothing

so we have:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = \left[\quad\right] \oplus \left[\quad\right]h \oplus \left[3bx_0 + c\right]h^2 \oplus \left[\quad\right]h^3$$

iv. If we want the *coefficient* of h^3 in the output jet:

- The cube function contributes a
- The square function contributes nothing
- The linear function contributes nothing
- The constant function contributes nothing

so we have:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = \left[\quad\right] \oplus \left[\quad\right] h \oplus \left[\quad\right] h^2 \oplus \left[a\right] h^3$$

4 Local graphs

Just as we get a *plot point at* a *bounded* input from the *output at* that input, we get the *local graph near* any input, be it *bounded* or *infinity*, from the *jet near* that input.

PROCEDURE 9.4

1. Get the *output jet* near ∞:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \underbrace{\left[a\right] x^3 \oplus \left[b\right] x^2 \oplus \left[c\right] x \oplus \left[d\right]}_{\text{output jet near } \infty}$$

(See PROCEDURE 9.2 on page 379.)

2. Get the *local graphs*:

a. Of the *cubic term* by graphing near ∞ the monomial function $x \to \left[a\right] x^3$ using ?? ?? on ??.

b. Of the *quadratic term* by graphing near ∞ the monomial function $x \to \left[a\right] x^2$ using ?? ?? on ??.

c. Of the *linear term* by graphing near ∞ the monomial function $x \to \left[b\right] x$ using ?? ?? on ??.

d. Of the *constant term* by graphing near ∞ the monomial function $x \to \left[c\right]$ using ?? ?? on ??.

3. Get the local graph near ∞ of $CUBIC$ using ?? by adding-on to the local graph of the *cubic term* the local graph of the *quadratic term*, the local graph of the the local graph of, and the local graph of the *constant term*.

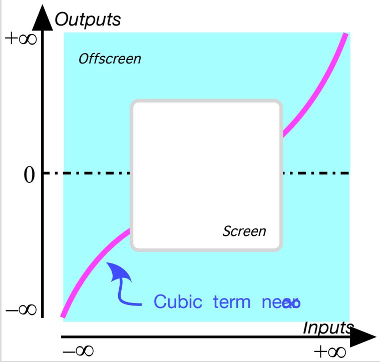
DEMO 9.4

1. We get the output jet near ∞: (See DEMO 9.2 on page 379)

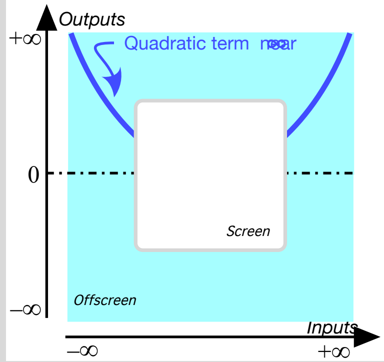
$$x \text{ near } \infty \xrightarrow{DINA} DINA(x) = \underbrace{\left[+37.81\right] x^3 \oplus \left[+51.32\right] x^2 \oplus \left[-82.47\right] x \oplus \left[-61.03\right]}_{\text{output jet near } \infty}$$

2. Get the local graph near ∞ of each term:

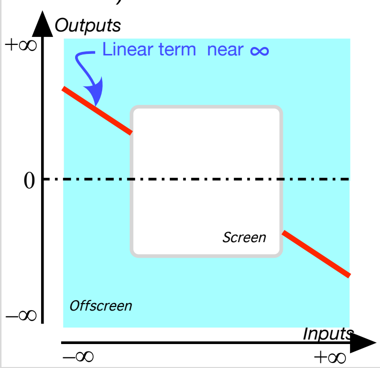
a. We get the graph of the *cubic* term by graphing the monomial function $x \to \left[+37.81\right] x^3$ near ∞ (See ?? on ??)



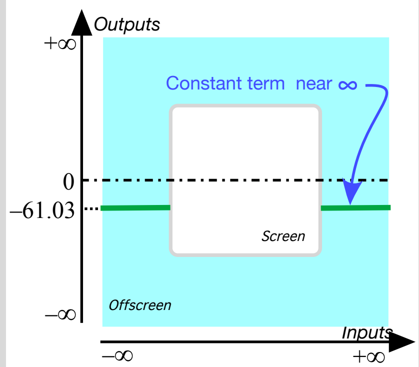
b. We get the graph of the *quadratic* term by graphing the monomial function $x \to \left[+51.32\right] x^2$ near ∞ (See ?? on ??)



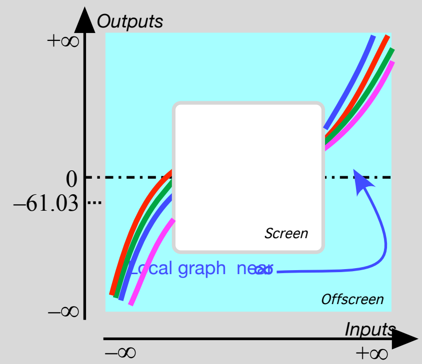
c. We get the graph of the *linear* term by graphing the monomial function $x \to \left[-82.47\right] x$ near ∞ (See ?? on ??)



d. We get the graph of the *constant* term by graphing the monomial function $x \to \left[-61.03\right]$ near ∞ (See ?? on ??)



3. We get the local graph near ∞ of $DINA$ by adding-on to the graph of the *quadratic* term the graph of the *linear* term and the graph of the *constant* term. (See ?? on ??)



PROCEDURE 9.5

1. Get the local *input-output rule* near x_0 of $CUBIC$ using PROCEDURE 9.3 To evaluate near x_0 the function specified by $x \xrightarrow{CUBIC} CUBIC(x) = ax^3 + bx^2 + cx + d$ on page 380

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0 + h)$$
$$= \underbrace{\left[ax_0^3 + bx_0^2 + cx_0 + d\right] \oplus \left[3ax_0^2 + 2bx_0 + c\right]h \oplus \left[3ax_0 + b\right]h^2 \oplus \left[a\right]h^3}_{\text{output jet near } x_0}$$

2. Get the *local graphs*:

a. Of the *constant* term by graphing near 0 the monomial function $x \to \left[ax_0^3 + bx_0^2 + cx_0 + d\right]$

b. Of the *linear* term by graphing near 0 the monomial function $x \to \left[3ax_0^2 + 2bx_0 + c\right]x$

c. Of the *quadratic* term by graphing near 0 the monomial function $x \to \left[3ax_0 + b\right]x^2$

d. Of the *cubic* term by graphing near 0 the monomial function $x \to \left[a\right]x^3$

3. Get the *local graph* of $CUBIC$ near x_0 by adding to the local graph of the *constant term*, the local graph of the *linear term*, the local graph of the *quadratic term*, the local graph of the *cubic term*.

DEMO 9.5

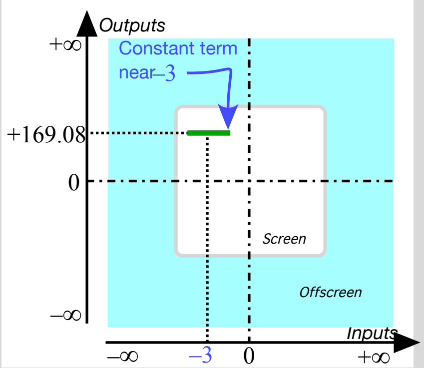
1. We get the *local input-output rule* near -3 of $ARBA$ (See DEMO 9.3

on page 381):

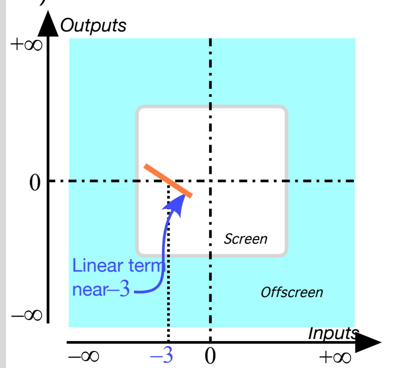
$-3+h \xrightarrow{ARNA} ARBA(-3+h) = \underbrace{\left[+1742.28\right] \oplus \left[-1362.09\right] h \oplus \left[+361.88\right] h^2 \oplus \left[+81.26\right] h^3}_{\text{output jet near } -3}$

2. We get the local graphs

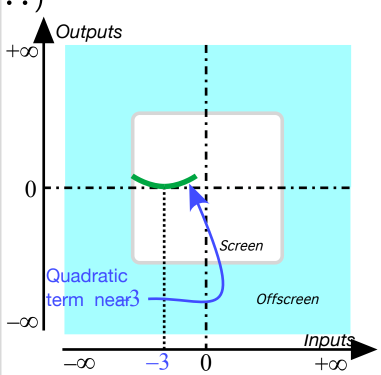
a. We get the graph of the *constant* term near -3 by graphing the monomial function $x \to \left[+1742.28\right]$. : (See ?? on ??)



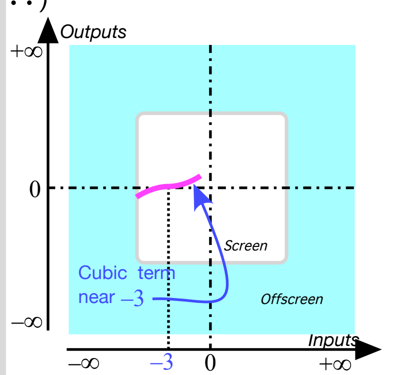
b. We get the graph of the *linear* term near -3 by graphing the monomial function $x \to \left[-1362.09\right] x$ (See ?? on ??)



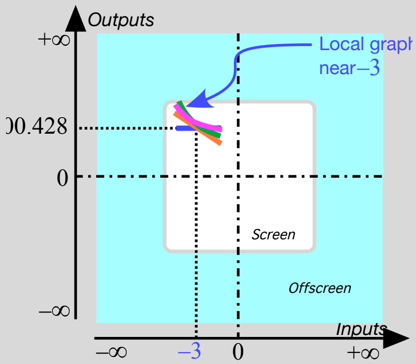
c. We get the graph of the *quadratic* term near -3 by graphing the monomial function $x \to \left[+361.88\right] x^2$: (See ?? on ??)



d. We get the graph of the *cubic* term near -3 by graphing the monomial function $x \to \left[+81.26\right] x^3$: (See ?? on ??)



3. We get the *local graph* near -3 of $ARBA$ by adding to the local graph of the *constant term* the local graph of the *linear term*, the local graph of the *quadratic term*, and the local graph of the *cubic term*. (See **??** on **??**)



5 Local Feature-signs

As we saw in **?? ??**, a feature-sign near a given input, be it near ∞ or near x_0, can be read from the *local graph* and so we could proceed as follows:

i. Get the *local input-output rule* near the given input (See PROCEDURE 9.2 on page 379 when the given input is ∞ or PROCEDURE 9.3 on page 380 when the given input is x_0.)

ii. Get the *local graph* from the local input-output rule (See PROCEDURE 9.4 on page 384.)

iii. Get the *feature-sign* from the *local graph*. (See **?? ??**.)

However, things are in fact much simpler: Given an input, be it ∞ or a bounded input x_0, to get a required feature-sign near that given input, we look for the term in the output jet near that input that

i. Has the required feature.

and

ii. Is the largest-in-size of all those terms with the required feature.

So, as we will now see, we usually need to get only *one* term in the output jet rather than the whole output jet.

1. Near *infinity* things are quite straightforward because, for a cubic function, the first term in the output jet near ∞ is both the *largest-in-size* and a *regular* monomial so that it has *all three features*:

PROCEDURE 9.6

i. Get the *approximate* local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{CUBIC} CUBIC(x) = \underbrace{\big[a\big]x^3 \oplus \big[b\big]x^2 \oplus \big[c\big]x \oplus \big[d\big]}_{\text{output jet near } \infty}$$
$$= \underbrace{\big[a\big]x^3 \oplus [...]}_{\text{approximate output jet near } \infty}$$

ii. Then, in the *approximate output jet* near ∞:

- Get the *Height-sign*, the *Slope-sign* and the *Concavity-sign* all from the *cubic term* $\big[a\big]x^3$ because the next terms, $\big[b\big]x^2$, $\big[c\big]x$ and $\big[d\big]$ are *too small to matter*. (Not to mention the fact that a linear term has no concavity and a constant term has neither concavity nor slope.)

DEMO 9.6 To get the Height-sign near ∞ of the function specified by

$$x \xrightarrow{DELIA} DELIA(x) = +12x^3 - 2x^2 + 63x - 155$$

.

i. We get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{DELIA} DELIA(x) = +12x^3 - 2x^2 + 63x - 155$$
$$= \underbrace{\big[+12\big]x^3 \oplus \big[-2\big]x^2 \oplus \big[+63\big]x \oplus \big[-155\big]}_{\text{output jet near } \infty}$$

ii. We get *Height-sign* from the *cubic* term $\big[+12\big]x^3$. (The *quadratic* term $\big[-2\big]x^2$, the linear term $\big[+63\big]x$ and the *constant* term $\big[-155\big]$ are *too small to matter*)
iii. Since the *cubic coefficient* $\big[+12\big]$ is *positive*, we get that Height-sign $DELIA$ near $\infty = \langle +, - \rangle$. (Seen from ∞.)

DEMO 9.7 Get the *slope-sign* near ∞ of the function specified by the global input-output rule

$$x \xrightarrow{DETER} DETER(x) = -0.45x^3 + 3.03x^2 - 81.67x + 46.92$$

i. We get the local input-output rule near ∞:

$$x \text{ near } \infty \xrightarrow{DETER} DETER(x) = -045x^3 + 3.03x^2 - 81.67x + 46.92$$

critical for the Concavity

$$= \underbrace{\Big[-0.45\Big]x^3 \oplus \Big[+3.03\Big]x^2 \oplus \Big[-81.67\Big]x \oplus \Big[+46.92\Big]}_{\text{output jet near } \infty}$$

ii. We get *Slope-sign* from the *cubic* term $\Big[-0.45\Big]x^3$. (The *quadratic term* $\Big[+3.03\Big]x^2$, the *linear term* $\Big[-81.67\Big]x$ and the *constant term* $\Big[+46.92\Big]$ are too small to matter.)
Since the *cubic coefficient* -0.45 is negative, we get that Slope-sign $DETER$ near $\infty = \langle \diagdown, \diagdown \rangle$. (Seen from ∞.)

2. Near a *bounded input* though, things are a bit more complicated:

i. The *first* term in the output jet is *usually* the *largest-in-size* so that it gives the Height-sign. However, the first term *usually* has neither Slope nor Concavity because the first term is *usually* a constant term.

ii. The *second* term in the output jet is *usually* too smalll-in-size to change the Height-sign as given by the first term but it is *usually* the *largest-in-size* term that can give the Slope-sign. However, the second term has no Concavity because the second term is *usually* a linear term.

iii. The third *term* in the output jet is *usually* too smalll-in-size to change the Height-sign given by the first term and the Slope-sign given by the second term but it is *usually* the *only term* that can give the Concavity-sign.

So we can *usually* read each feature-sign directly from the appropriate term in the output jet - keeping in mind that the exceptional monomial functions do not have all the features.

However, near a *bounded input*, the given bounded input may turn out to be *critical* for the local feature:

i. If the *constant term* in the output jet is 0, then the term which gives the Height-sign can be either the linear term or the quadratic term if the linear term is 0 or even the cubic term if the quadratic term turns out to be 0 too. The bounded input is then again said to be *critical for the Height*.

ii. If the *linear term* in the output jet is 0, then the term which gives the Slope-sign is the *quadratic term* or the *cubic term* is the *quadratic term* turns out to be 0 too. The bounded input is then said to be *critical for the Slope*.

iii. If the *quadratic term* in the output jet is 0, then the term which gives the Concavity-sign is the *cubic term*. The bounded input is then said to be **critical for the Concavity**.

So, we *usually* need to compute only one coefficient in the output jet. But if the given bounded input turns out to be *critical* for that feature, then we need to compute the next coefficient: So we use

PROCEDURE 9.7
i. Get the local input-output rule near x_0:

$$x_0 + h \xrightarrow{CUBIC} CUBIC(x_0+h) = a(x_0+h)^3 + b(x_0+h)^2 + c(x_0+h) + d$$
$$= a\left(x_0^3 + 3x_0^2h + 3x_0h^2 + h^3\right) + b\left(x_0^2 + 2x_0h + h^2\right) + c\left(x_0+h\right) + d$$
$$= \underbrace{\Big[ax_0^3 + bx_0^2 + cx_0 + d\Big] \oplus \Big[3ax_0^2 + 2bx_0 + c\Big]h \oplus \Big[3ax_0 + b\Big]h^2 \oplus \Big[a\Big]h^3}_{\text{output jet near } x_0}$$

ii. Then, in the *output jet* near x_0:

- Get the *Height-sign* from the *constant term* $\Big[ax_0^3 + bx_0^2 + cx_0 + d\Big]$. (The *linear term*, the *quadratic term* and the *cubic term* are *too small to matter*.)
  If the *constant coefficient* is 0, get the *Height-sign* from the *linear term* $\Big[3ax_0^2 + 2bx_0 + c\Big]h$. (The *quadratic term* and the *cubic term* are too small to matter.)
  If the *linear coefficient* is 0 too, get the *Height-sign* from the *quadratic term* $\Big[3ax_0 + b\Big]h^2$. (The *quadratic term* and the *cubic term* are too small to matter.)
  If the *quadratic coefficient* is 0 too, get the *Height-sign* from the *cubic term* $\Big[a\Big]h^3$. (The *quadratic term* and the *cubic term* are too small to matter.)
- Since the *constant term* has no slope, get the *Slope-sign* from the *linear term* $\Big[3ax_0^2 + 2bx_0 + c\Big]h$. (The *quadratic term* and the *cubic term* are too small to matter.)
  If the *linear coefficient* is 0, get the *Slope-sign* from the *quadratic term* $\Big[3ax_0 + b\Big]h^2$. (The *cubic term* is too small to matter.)
  If the *quadratic coefficient* is 0 too, get the *Slope-sign* from the *cubic term* $\Big[a\Big]h^3$.
- Since both the *constant term* and the *linear term* have no concavity, get *Concavity-sign* from the *quadratic term* $\Big[3ax_0 + b\Big]h^2$. (The *cubic term* is too small to matter.)
  If the *quadratic coefficient* is 0, get the *Slope-sign* from the *cubic term* $\Big[a\Big]h^3$.

DEMO 9.8 To get the feature signs near -3 of the function specified by the global input-output rule

$$x \xrightarrow{ARBA} ARBA(x) = -32.67x + 71.07 + 81.26x^2$$

i. We get the local input-output rule near -3 (See DEMO 9.3 on page 381):

$$-3+h \xrightarrow{ARBA} ARBA(-2+h) = \underbrace{-32.67(-3+h)+71.07+81.26(-3+h)^2}_{\text{output specifying code}}$$

$$= \underbrace{\left[+900.428\right] \oplus \left[-519.63\right] h \oplus \left[+81.26\right] h^2}_{\text{output jet near } -3}$$

ii. Then, from the *output jet*:

- Since the *constant coefficient* $\left[+900.428\right]$ is *positive*, we get that Height-sign $ARBA$ near $-3 = \langle +,+ \rangle$.
- Since the *linear coefficient* $\left[-519.63\right] h$ is *negative*. we get that Slope-sign $ARBA$ near $-3 = \langle \setminus, \setminus \rangle$
- Since the *quadratic coefficient* $\left[+81.26\right] h^2$ is *positive*, we get that Concavity-sign $ARBA$ near $-3 = \langle \cup, \cup \rangle$

6 Cubic Functions: Global Analysis

In the case of cubic functions, we will be able to solve *exactly* only a very few global problems because everything begins to be truly computationally complicated.

7 Global Graph

As always, we use

PROCEDURE 9.8

i. Graph the function near ∞, (See PROCEDURE 9.4 on page 384.)

ii. Ask the ESSENTIAL QUESTION:

- Do all *bounded inputs* have *bounded outputs*

or

- Are there *bounded inputs* whose nearby inputs have unbounded outputs? (∞-height inputs.)

iii. Use the local input-output rule near x_0 to get further information.

(See PROCEDURE 9.3 on page 380.)

essential -feature input

But, given a *bounded* input x_0, we have that:
- a being bounded, ax_0^3 is also bounded
- b being bounded, bx_0^2 is also bounded
- c being bounded, cx_0 is also bounded
- and d being bounded

altogether, we have that $ax_0^3 + bx_0x^2 + cx_0 + d$ is bounded and that the answer to the ESSENTIAL QUESTION is:

EXAMPLE 9.6. Bounded Height Under a *cubic* functions, all bounded inputs have *bounded outputs*.

and therefore

EXAMPLE 9.7. Offscreen Graph The *offscreen graph* of a *cubic* function consists of just the *local graph near* ∞.

We now deal in detail with the third step.

EXISTENCE THEOREMS

Since cubic functions have no *bounded* ∞-height input, the only way a feature can change sign near a bounded input is when the feature is 0 near the bounded input. In particular, **essential 0-feature inputs** are bounded inputs
- with a 0 feature,
- whose existence is forced by the *offscreen graph*—which, in the case of cubic functions consists, by EXAMPLE 9.7, of only the *local graph near* ∞.

None of the following tEXAMPLEs, though, will indicate *where* the 0-feature inputs inputs are *located*. The **Location TEXAMPLEs** will be dealt with in the last part of the chapter.

8 Concavity-sign

Given the function specified by the global input-output rule

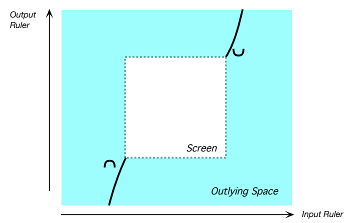
$$x \xrightarrow{CUBIC} CUBIC(x) = ax^3 + b^2x + cx + d$$

recall that when x is near ∞ the **Concavity-sign Near ∞ TEXAMPLE** for cubic functions says that:

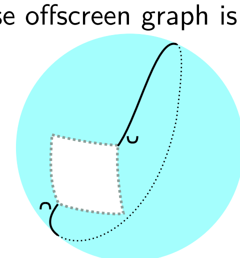
- When a is $+$, Concavity-Sign$|_{x \text{ near } \infty} = (\cup, \cap)$
- When a is $-$, Concavity-Sign$|_{x \text{ near } \infty} = (\cap, \cup)$

1. Since the concavity changes sign as x goes from the left side of ∞ to the right side of ∞ *across* ∞ , the concavity also has to change sign as x goes from the left side of ∞ to the right side of ∞ *across the screen.* In other words, there has to be a *bounded* concavity-sign change input.

Example 9.8. **G** iven a cubic function whose offscreen graph is

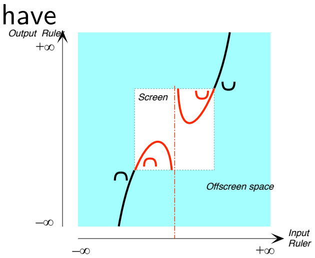


Mercator view

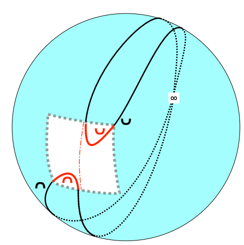


Magellan view

there has therefore to be a bounded concavity-sign change input, $x_{\text{concavity sign-change}}$. But since there cannot be a bounded ∞-height input, we cannot have

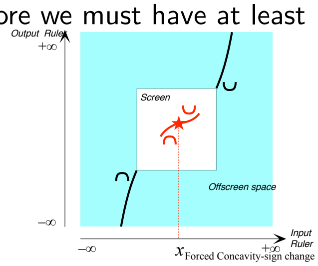


Mercator view

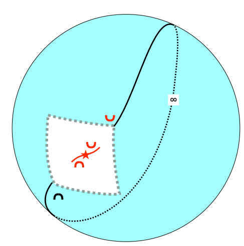


Magellan view

and therefore we must have at least



Mercator view



Magellan view

So, based on the off-screen graph, we have

Example 9.9. **Concavity sign-change** A cubic function must have *at least one* bounded concavity sign-change input.

2. On the other hand, based on the off-screen graph, a cubic function could have any *odd* number of 0-concavity inputs. Based on the *general local input-output rule*, we will see that a cubic function can have *at most one* 0-concavity input. But, at this point, all we know for sure is

EXAMPLE 9.10. 0-**Concavity Existence** A cubic functions must have *at least one* concavity-sign change input:

$$x_{\text{concavity sign-change}} = x_{\text{0-concavity}}$$

9 Slope-sign

Given the cubic function $CUBIC_{a,b,c,d}$, that is the function specified by the global input-output rule

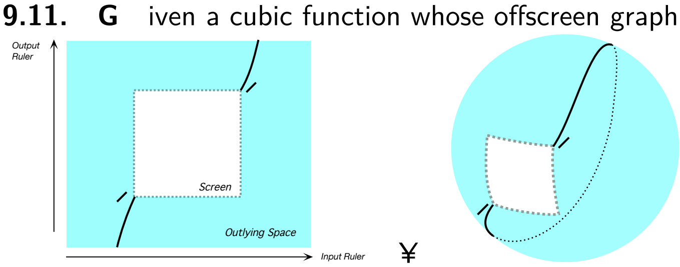
$$x \xrightarrow{\ CUBIC\ } CUBIC(x) = ax^3 + b^2x + cx + d$$

recall that when x is near ∞ the **Slope-sign Near** ∞ **TEXAMPLE** for cubic functions says that:

- When a is $+$, Slope-Sign$|_{x \text{ near } \infty} = (\diagup, \diagup)$
- When a is $-$, Slope-Sign$|_{x \text{ near } \infty} = (\diagdown, \diagdown)$

1. Since the slope does *not* changes sign as x goes through ∞ from the left side of ∞ to the right side of ∞, the slope does not have to change sign as x goes *across the screen* from the left side of ∞ to the right side of ∞ so there does not have to be a *bounded* slope-sign change input:

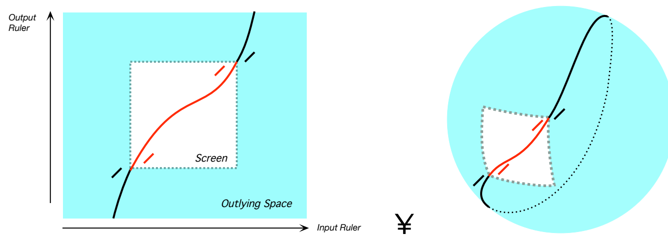
EXAMPLE 9.11. **G** iven a cubic function whose offscreen graph is



¥

Mercator view ¥ Magellan view

there is no need for a bounded slope-sign change input, $x_{\text{Slope-sign change}}$ and therefore we can have



¥

Mercator view ¥ Magellan view

2. On the other hand, based on just *graphic* considerations, a cubic function could have any number of 0-slope inputs. Based on *input-output rule* considerations, we will see that a cubic function can have only zero, one or two 0-slope inputs. But, at this point, all we know for sure is

EXAMPLE 9.12. Slope-Sign Change Existence A cubic function need not have a *Slope-sign change* input.

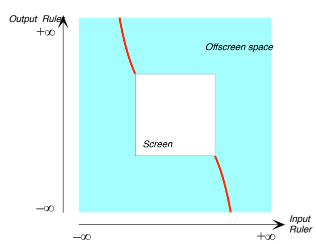
And thus also

EXAMPLE 9.13. 0-**Slope Existence** A cubic function need not have a 0-*Slope* input.

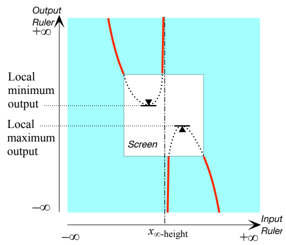
10 Extremum

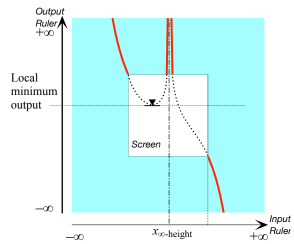
From the *optimization* viewpoint, the most immediately striking feature of an affine function is the absence of a forced extreme input, that is of a bounded input whose output is either larger than the output of nearby inputs or smaller than the output of nearby inputs. On the other hand, at this point we cannot prove that there is no extreme input.

EXAMPLE 9.14. G iven a cubic function with the offscreen graph:

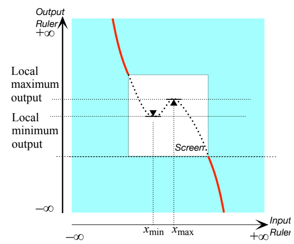


Since there can be no ∞-height input, we cannot have, for instance, either one of the following

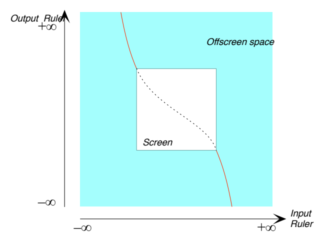




On the other hand, there is nothing to prevent a *fluctuation* such as: ¥



But no extremum input is *forced*: ¥



So, we have

EXAMPLE 9.15. Extremum Existence A cubic function has no *forced* extremum input

11 Height-sign

Given the cubic function $CUBIC_{a,b,c,d}$, that is the function specified by the global input-output rule

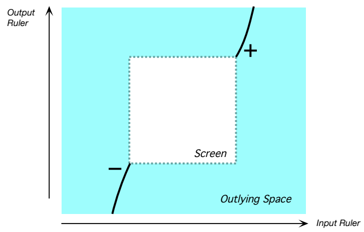
$$x \xrightarrow{CUBIC} CUBIC(x) = ax^3 + b^2x + cx + d$$

recall that when x is near ∞ the **Height-sign Near ∞ TEXAMPLE** for cubic functions says that:

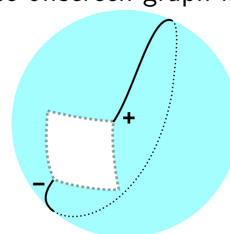
- When a is $+$, Height-Sign$|_{x \text{ near } \infty} = (+, -)$
- When a is $-$, Height-Sign$|_{x \text{ near } \infty} = (-, +)$

1. Since the height changes sign as x goes from the left side of ∞ to the right side of ∞ *across* ∞ , the height has also to change sign as x goes from the left side of ∞ to the right side of ∞ *across the screen*. In other words, there has to be a *bounded* height-sign change input.

EXAMPLE 9.16. G iven a cubic function whose offscreen graph is

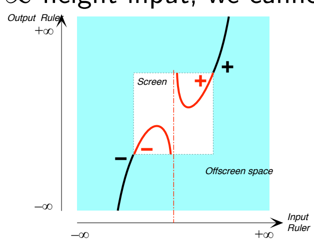


Mercator view

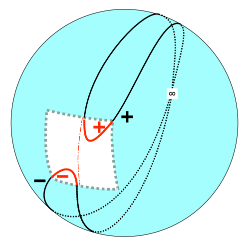


Magellan view

there has therefore to be a height-sign change input But since there cannot be a bounded ∞-height input, we cannot have

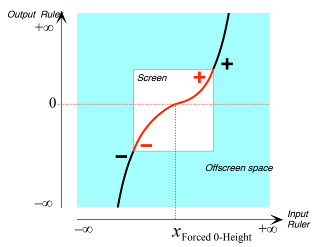


Mercator view

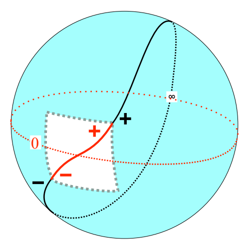


Magellan view

and therefore we must have



Mercator view



Magellan view

2. Moreover, because there is no *bounded ∞-height input* where the height could change sign, $x_{\text{height-sign change}}$ has to be a bounded input where the height is 0. As a result, we have that

EXAMPLE 9.17. Height-Sign Change Existence A cubic functions must have a *Height-sign change* input and

$$x_{\text{Height-sign change}} = x_{\text{0-height}}$$

LOCATION THEOREMS

Previously, we only established the *existence* of certain essential bounded inputs of cubic functions and this investigation was based on *graphic* considerations. Here we will investigate the *location* of the essential bounded inputs and this investigation will be based on the *generic local input-output rule*.

12 0-Concavity Location

Given a cubic function, the global problem of *locating* an input where the local concavity is 0 is still fairly simple.

More precisely, given a cubic function $CUBIC_{a,b,c,d}$, that is the cubic function specified by the global input-output rule

$$x \xrightarrow{CUBIC} CUBIC(x) = ax^3 + b^2x + cx + d$$

since the *concavity* near x_0 is the local square coefficient $3ax_0 + b$, in order to find the input(s) where the local concavity is 0, we need to solve the affine equation

$$3ax + b = 0$$

by reducing it to a basic equation:

$$\begin{aligned} 3ax + b - b &= 0 - b \\ 3ax &= -b \\ \frac{3ax}{3a} &= \frac{-b}{3a} \\ x &= \frac{-b}{3a} \end{aligned}$$

So, we have:

EXAMPLE 9.18. 0-slope Location For any cubic function $CUBIC_{a,b,c,d}$,

$$x_{0-concavity} = \frac{-b}{3a}$$

In fact, we also have:

EXAMPLE 9.19. Global Concavity-sign Given a cubic function $CUBIC_{a,b,c,d}$,

- When a is positive,

$$\begin{aligned} \text{Concavity-sign } CUBIC|_{\text{Everywhere} < \frac{-b}{3a}} &= (\cap, \cap) \\ \text{Concavity-sign } CUBIC|_{\frac{-b}{3a}} &= (\cap, \cup) \\ \text{Concavity-sign } CUBIC|_{\text{Everywhere} > \frac{-b}{3a}} &= (\cup, \cup) \end{aligned}$$

- When a is negative,

$$\begin{aligned}
\text{Concavity-sign } CUBIC|_{\text{Everywhere} < \frac{-b}{3a}} &= (\cup, \cup) \\
\text{Concavity-sign } CUBIC|_{\frac{-b}{3a}} &= (\cup, \cap) \\
\text{Concavity-sign } CUBIC|_{\text{Everywhere} > \frac{-b}{3a}} &= (\cap, \cap)
\end{aligned}$$

The case is easily made by testing near ∞ the intervals for the corresponding inequations.

13 0-Slope Location

In the case of affine functions and of quadratic functions, we were able to prove that there was no shape difference with the principal term near ∞ by showing that there could be no *fluctuation*:

- In the case of *affine functions* we were able to prove that there was no shape difference with *dilation functions*
- In the case of *quadratic functions* we were able to prove that there was no shape difference with *square functions*.

More precisely, given the cubic function $CUBIC_{a,b,c,d}$, that is the function specified by the global input-output rule

$$x \xrightarrow{\quad CUBIC \quad} CUBIC(x) = ax^3 + b^2x + cx + d$$

since the slope near x_0 is the local linear coefficient $3ax^2 + 2bx + c$, in order to find the input(s) where the local slope is 0, we need to solve the *quadratic equation*

$$3ax^2 + 2bx + c$$

which we have seen we cannot solve by reduction to a basic equation and for which we will have to use the 0**-Height TEXAMPLE** for quadratic functions, keeping in mind, though, that

- For a as it appears in 0**-Height TEXAMPLE** for quadratic functions, we have to substitute the *squaring* coefficient of $3ax^2 + 2bx + c$, namely $3a$,
- For b as it appears in 0**-Height TEXAMPLE** for quadratic functions, we have to substitute the *linear coefficient* of $3ax^2 + 2bx + c$ namely $2b$,
- For c as it appears in 0**-Height TEXAMPLE** for quadratic functions, we have to substitute the *constant* coefficient of $3ax^2 + 2bx + c$ namely c.

1. It will be convenient, keeping in mind the above substitutions, first to compute

$$x_{0-slope \text{ for } [3ax^2+2bx+c]} = -\frac{2b}{2 \cdot 3a}$$

$$= -\frac{2b}{6a}$$
$$= -\frac{b}{3a}$$
$$= x_{0-concavity} \text{ for } CUBIC$$

Shape type O

2. Then, still keeping in mind the above substitutions, we compute the discriminant of $3ax^2 + 2bx + c$:

$$\text{Discriminant}[3ax^2 + 2bx + c] = (2b)^2 - 4(3a)(c)$$
$$= 4b^2 - 12ac$$

3. Then we have:

- When Discriminant $[3ax^2+2bx+c] = 4b^2-12ac < 0$, the local linear coefficient of $CUBIC$, $[3ax^2+2bx+c]$, has *no* 0-height input and therefore $CUBIC$ has *no* 0-slope input.
- When Discriminant $[3ax^2 + 2bx + c] = 4b^2 - 12ac = 0$, the local linear coefficient of $CUBIC$, $[3ax^2 + 2bx + c]$, has *one* 0-height input and therefore $CUBIC$ has *one* 0-slope input, namely
  - ▶ $x_{0-\text{slope for } CUBIC} = x_{0-height \text{ for } [3ax^2+2bx+c]} = -\frac{b}{3a}$,
- When Discriminant $[3ax^2 + 2bx + c] = 4b^2 - 12ac > 0$, the local linear coefficient of $CUBIC$, $[3ax^2 + 2bx + c]$, has *two* 0-height inputs and therefore $CUBIC$ has *two* 0-slope inputs., namely:
  - ▶ $x_{0-\text{slope for } CUBIC} = x_{0-height \text{ for } [3ax^2+2bx+c]} = -\frac{b}{3a} + \frac{\sqrt{4b^2-12ac}}{2a}$

  and
  - ▶ $x_{0-\text{slope for } CUBIC} = x_{0-height \text{ for } [3ax^2+2bx+c]} = -\frac{b}{3a} - \frac{\sqrt{4b^2-12ac}}{2a}$

In terms of the function $CUBIC$, this gives us:

EXAMPLE 9.20. 0-**slope Location** Given the cubic function $CUBIC_{a,b,c,d}$, when

- Disc. $[3ax^2 + 2bx + c] = 4b^2 - 12ac < 0$, $CUBIC$ has *no* 0-Slope input
- Disc. $[3ax^2 + 2bx + c] = 4b^2 - 12ac = 0$, $CUBIC$ has *one* 0-Slope input
- Disc. $[3ax^2 + 2bx + c] = 4b^2 - 12ac > 0$, $CUBIC$ has *two* 0-Slope inputs

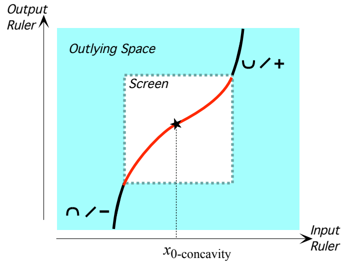
14 Extremum Location

The 0-slope inputs are the only ones which can be extremum inputs. So, there will therefore be three types of cubic functions according to the number of 0-slopes inputs:

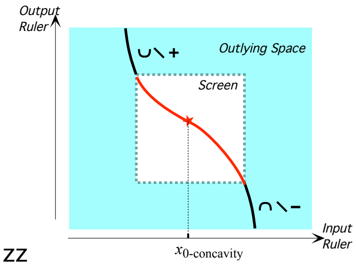
1. When Discriminant $[3ax^2+2bx+c] = 4b^2-12ac < 0$ so that $CUBIC$ has *no* 0-Slope input, there can be no extremum input and we will say that this type of cubic is of **Shape type 0**.

Shape type I
Shape type II

EXAMPLE 9.21.



zz

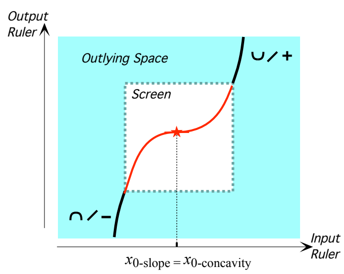


Cube coefficient positive zz Cube coefficient negative

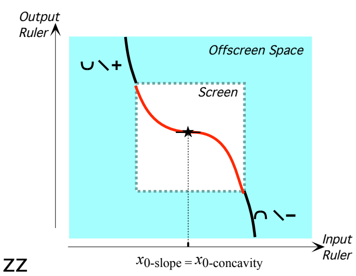
Since cubic function of Shape type O have no 0-Slope input, their shape is *not* like that of *cubing functions*.

2. When Discriminant $[3ax^2+2bx+c] = 4b^2-12ac = 0$ so that $CUBIC$ has *one* 0-Slope input, there will still be no extremum input and we will say that this type of cubic is of **Shape type I**.

EXAMPLE 9.22.



zz

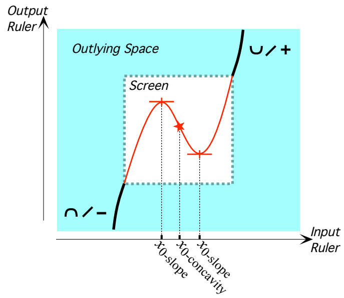


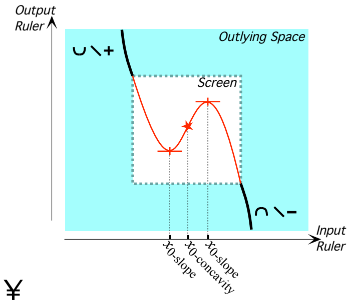
Cube coefficient positive zz Cube coefficient negative

Since cubic function of *Shape type I* do have one 0-Slope input, their shape is very much like that of *cubing functions*.

3. When Discriminant $[3ax^2+2bx+c] = 4b^2-12ac > 0$ so that $CUBIC$ has *two* 0-Slope input, there will be one minimum input and one maximum input and we will say that this type of cubic is of **Shape type II**.

EXAMPLE 9.23.





¥

Cube coefficient positive ¥ Cube coefficient negative

We can thus state:

EXAMPLE 9.24. Extremum Location Given the cubic function $CUBIC_{a,b,c,d}$, when

- Discriminant $[3ax^2 + 2bx + c] = 4b^2 - 12ac < 0$, $CUBIC$ has no locally extremum input.
- Discriminant $[3ax^2 + 2bx + c] = 4b^2 - 12ac = 0$, $CUBIC$ has one locally minimum-maximum input or one locally maximum-minimum input.
- Discriminant $[3ax^2 + 2bx + c] = 4b^2 - 12ac > 0$, $CUBIC$ has both
  - ▶ $x_{\text{locally minimum-output}}$,
  - ▶ $x_{\text{locally maximum-output}}$,

15 0-Height Location

The location of 0-height inputs in the case of a cubic function is usually not easy.

1. So far, the situation has been as follows:

i. The number of 0-height inputs for *affine functions* is always *one*,

ii. The number of 0-height inputs for *quadratic functions* is already more complicated in that, depending on the sign of the extreme-output compared with the sign of the outputs for inputs near ∞, it can be *none*, *one* or *two*.

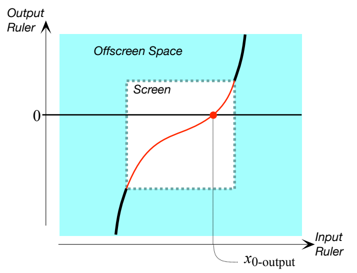
It follows from the **Extremum Location TEXAMPLE** that

iii. The number of 0-height inputs for *cubic functions* depends

a. On the Shape type of the cubic function,

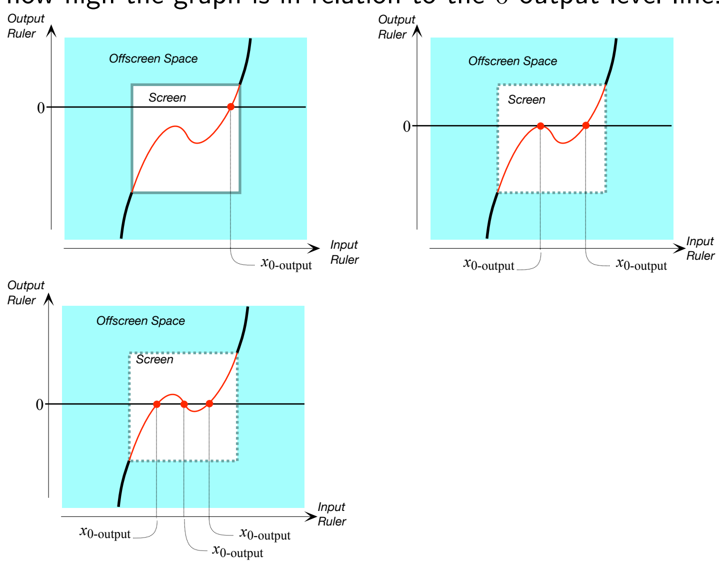
b. In the case of Shape type II, on the sign of the extremum outputs relative to the sign of the cubing coefficient

EXAMPLE 9.25. **T** he cubic function specified by the global graph



¥ is of Shape Type O (No 0-slope input) and always has a single 0-height input.

EXAMPLE 9.26. **T** he cubic function specified by the global graphs are all of the same shape of Type II and the number of 0-height inputs depends on how high the graph is in relation to the 0-output level line.



2. The *obstruction* to computing the solutions that we encountered when trying to solve *quadratic equations*, namely that there was one more term than an equation has sides is even worse here since we have four terms and an equation still has only two sides. See **??** on **??**

Chapter 10

Quartic Functions

nb,nb,mnb,mbm

Chapter 11

Quintic Functions

nb,nb,mnb,mbm

Part IV

Rational Functions

zzzzzzzz

Chapter 12

Rational Degree & Algebra Reviews

Rational functions are functions whose *global input-output rule* is of the form

$$x \xrightarrow{RAT} RAT(x) = \frac{POLY_{Num}(x)}{POLY_{Den}(x)}$$

where $POLY_{Num}(x)$ and $POLY_{Den}(x)$ stand for two *positive-exponent* polynomial expressions.

EXAMPLE 12.1. **T** he function whose global input-output rule is

$$x \xrightarrow{TAB} TAB(x) = \frac{-3x^2 + 4x - 7}{-5x^4 - 8}$$

is a rational function in which:

- $POLY_{Num}(x)$ is $-3x^2 + 4x - 7$
- $POLY_{Den}(x)$ is $-5x^4 - 8$

1 Rational Degree

Because the *upper degree* of polynomial functions is what we used to sort polynomial functions into different *types*, we now try to extend the idea of *upper degree* to the case of rational functions in the hope that this will also help us sort rational functions into different *types*.

rational degree
regular rational function
exceptional rational function

Given a rational function whose global input-output rule is

$$x \xrightarrow{RAT} RAT(x) = \frac{POLY_{Num}(x)}{POLY_{Den}(x)}$$

the **rational degree** of this rational function is the upper degree of $POLY_{Num}(x)$ minus the upper degree of $POLY_{Den}(x)$:

$$\text{Rat.Deg. of } \frac{POLY_{Num}(x)}{POLY_{Den}(x)} = \text{UppDeg. of } POLY_{Num}(x) - \text{UppDeg. of } POLY_{Den}(x)$$

Thus, the *rational degree* of a rational function can well be *negative*.

CAUTIONARY NOTE 12.1 he *rational degree* is to rational function very much what the *size* is to arithmetic fractions in "school arithmetic" which distinguishes fractions according to the *size* of the numerator compared to the *size* of the denominator even though, by now, the distinctions are only an inconsequential remnant of history..
What happened is that, historically, the earliest arithmetic fractions were "unit fractions" , that is reciprocals of whole numbers such as one half, one third, one quarter, etc. Later came "Egyptian fractions", that is combinations of (distinct) unit fractions, such as one third and one fifth and one eleventh, etc. A much later development were the "proper fractions", also called "vulgar fractions", such as two thirds, three fifths etc. Later still, came "improper fractions" such as five thirds, seven halves, etc. And finally "mixed numbers", such as three and two sevenths. Today, none of these distinctions matters inasmuch as we treat all fractions in the same manner.
However, while these "school arithmetic" distinctions are based on the *size* of the numerator versus the *size* of the denominator and make no real differences in the way we handle arithmetic fractions, in the case of rational functions, the above distinction based on the *upper degree* of the numerator versus the *upper degree* of the denominator will make a difference—even though no major one—in the way we will handle rational functions of different types.

In fact, by analogy with what we did with *power functions*, we will say that

- Rational functions whose rational degree is either > 1 or < 0, are **regular rational functions**,
- Rational functions whose rational degree is either $= 0$ or $= 1$, are **exceptional rational functions**.

EXAMPLE 12.2. **F** ind the rational degree of the function $DOUGH$ whose global input-output rule is

$$x \xrightarrow{DOUGH} DOUGH(x) = \frac{+1x^4 - 6x^3 + 8x^2 + 6x - 9}{x^2 - 5x + 6}$$

Since the rational degree is given by

$$\text{Rat.Deg. of } \frac{POLY_{Num}(x)}{POLY_{Den}(x)} = \text{UppDeg. of } POLY_{Num}(x) - \text{UppDeg. of } POLY_{Den}(x)$$

and since, here,

- $POLY_{Num}(x) = +1x^4 - 6x^3 + 8x^2 + 6x - 9$
- $POLY_{Den}(x) = +1x^2 - 5x + 6$

we get from the definition of the upper degree of a polynomial that:

$$\begin{aligned}\text{UppDeg. of } +1x^4 - 6x^3 + 8x^2 + 6x - 9 &= \text{Exponent of Highest Term}\\ &= \text{Exponent of } +1x^4\\ &= 4\\ \text{UppDeg. of } +1x^2 - 5x + 6 &= \text{Exponent of Highest Term}\\ &= \text{Exponent of } +1x^2\\ &= 2\end{aligned}$$

so that the rational degree of the rational function $DOUGH$ is:

$$\begin{aligned}\text{Rat.Deg. of } \frac{+1x^4 - 6x^3 + 8x^2 + 6x - 9}{+1x^2 - 5x + 6} &= \text{Exponent of } +1x^4 - \text{Exponent of } +1x^2\\ &= 4 - 2\\ &= 2\end{aligned}$$

so that $DOUGH$ is an example of a rational function of degree > 1 and therefore of a *regular* rational function.

2 Graphic Difficulties

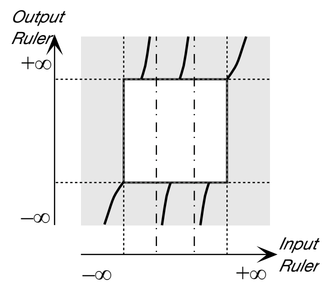
Finally, when there is one or more ∞-height bounded input(s), beginners often encounter difficulties when trying to interpolate smoothly the outlying graph of a rational function.

The difficulties are caused by the fact that, when we draw the local graph near ∞ and the local graphs near the ∞-height inputs from the local input-output rules, we are only concerned with drawing the local graphs themselves from the local input-output rules. In particular, when we draw the local graph near ∞ and the local graphs near the ∞-height inputs, we want to bend them enough to show the concavity but we often end up bending them *too much* to interpolate them.

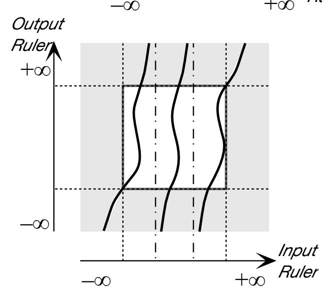
But then, what often happens as a result is that, when we want to interpolate, the local graphs may not line up well enough for us to interpolate them (smoothly).

EXAMPLE 12.3.

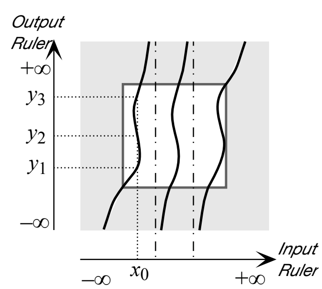
Given the rational function whose offscreen graph was drawn so as to show the concavity. ¥



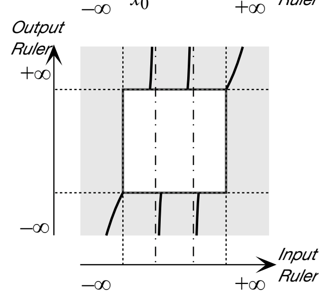
Here is what can happens when we attempt to interpolate ¥



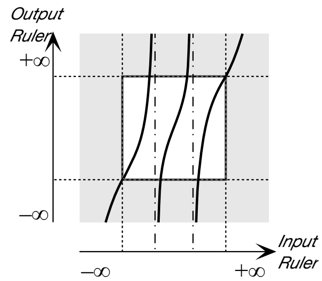
Of course, this is absolutely impossible since, according to this global graph, there would be inputs, such as x_0, with more than one output, y_1, y_2, $\ldots$: ¥



But if we unbend the local graphs just a bit as in ¥

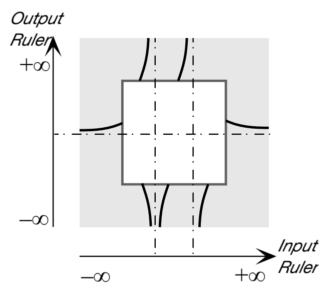


we have no trouble interpolating: ¥

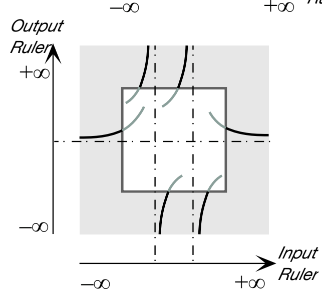


The way to avoid this difficulty is not to wait until we have to interpolate but to catch any problem as we draw the local graphs by mentally extending the local graphs slightly into the transitions.

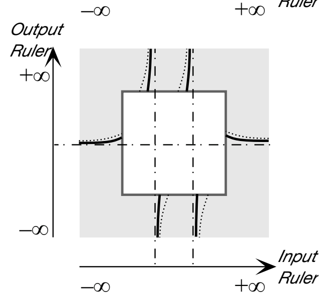
EXAMPLE 12.4.
Given the rational function whose offscreen graph was drawn do as to show the concavity ¥



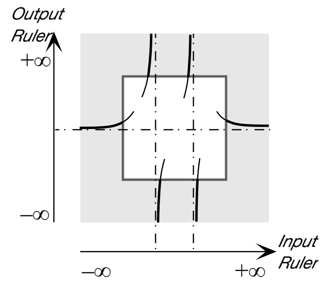
we can already see by extending the local graphs just a little bit into the transitions that this will cause a lot of trouble when we try to interpolate the local graph: ¥



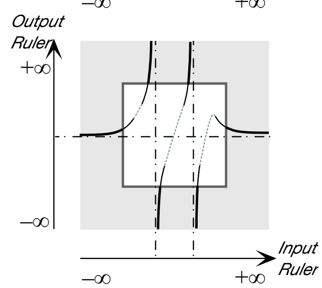
So, here, we bend the local graph near ∞ a little bit more and we unbend the local graphs near the ∞-height inputs a little bit: ¥



We check again by extending the local graphs just a little bit into the transitions: ¥



and indeed now we have no trouble interpolating: ¥



??
??
??
??
??
??
??
??
??
??
??
??
??
??
==================================
??
??
??
??
??
??
??
??

??
??
=================================
??
??
??
??
??
??
??
??
??
??
=================================
??
??
??
??
??
??
??
??
??
??
=================================
??
??
??
??
??
??
??
??
??
??
=================================
??
??
??
??

??
??
??
??
??
??

================================

extract

Chapter 13

Rational Functions: Local Analysis Near ∞

To do local analysis we work in a neighborhood of some given input and thus count inputs from the given input since it is the center of the neighborhood. When the given input is ∞, counting from ∞ means setting $x \leftarrow large$ and computing with powers of *large* in descending order of sizes.

Recall that the *principal term* near ∞ of a given polynomial function *POLY* is simply its highest power term which is therefore easy to **extract** from the global input-output rule. The approximate input-output rule near ∞ of *POLY* is then of the form

$$x|_{x \text{ near } \infty} \xrightarrow{POLY} POLY(x)|_{x \text{ near } \infty} = Highest\,Term\;POLY + [...]$$

However, the complication here is that to get the principal part near ∞ of a rational function we must approximate the two polynomial and divide—or the other way round—and the result need not be a polynomial but can also be a negative-exponent power function and the main issue will be whether to do the approximation before or after the division.

1 Local Input-Output Rule Near ∞

Given a rational function *RAT*, we look for the function whose input-output rule will be simpler than the input-output rule of *RAT* but whose local graph near ∞ will be qualitatively the same as the local graph near ∞ of *RAT*.

More precisely, given a rational function RAT specified by the global input-output rule

$$x \xrightarrow{RAT} RAT(x) = \frac{POLY_{Num}(x)}{POLY_{Den}(x)}$$

what we will want then is an *approximation* for the output of the local input-output rule near ∞

$$x|_{x \text{ near } \infty} \xrightarrow{RAT} RAT(x)|_{x \text{ near } \infty} = \left.\frac{POLY_{Num}(x)}{POLY_{Den}(x)}\right|_{x \text{ near } \infty}$$

from which to *extract* whatever controls the wanted feature.

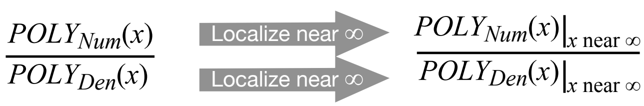
1. Since the center of the neighborhood is ∞, we *localize* both

- $POLY_{Num}(x)$

and

- $POLY_{Den}(x)$

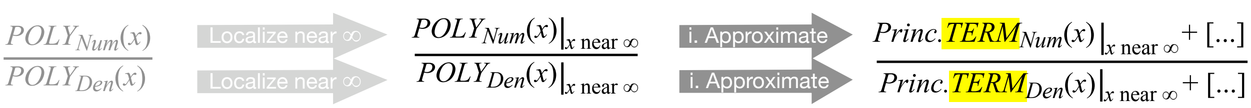
by writing them in *descending* order of exponents.



2. Depending on the circumstances, we will take one of the following two routes to *extract* what controls the wanted feature:

■ The *short route* to $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$, that is:

i. We approximate both $POLY_{Num}(x)|_{x \text{ near } \infty}$ and $POLY_{Den}(x)|_{x \text{ near } \infty}$ to their *principal term*—that is to just their *highest size term*—which, since x is near ∞, is their *highest exponent term*:



ii. In order to divide $Princ.TERM_{Num(x)}\Big|_{x \text{ near } \infty}$, that is the principal term near ∞ of the *numerator* of RAT by $Princ.TERM_{Den(x)}\Big|_{x \text{ near } \infty}$, that is the principal term near ∞ of the *denominator* of RAT we use monomial division

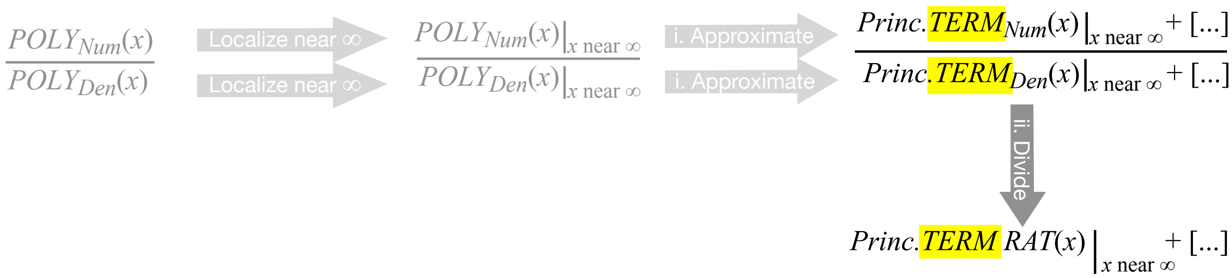
$$\boxed{\frac{ax^{+m}}{bx^{+n}} = \frac{a}{b}x^{+m\ominus+n}}$$ where $+m \ominus +n$ can turn out positive, negative or 0

$$Princ.TERM\ RAT(x)\Big|_{x \text{ near } \infty} = \frac{Princ.TERM_{Num}(x)\Big|_{x \text{ near } \infty}}{Princ.TERM_{Den}(x)\Big|_{x \text{ near } \infty}}$$

$$= \frac{\text{coef. } Princ.TERM_{Num}(x)\Big|_{x \text{ near } \infty}}{\text{coef. } Princ.TERM_{Den}(x)\Big|_{x \text{ near } \infty}} \cdot x^{\text{UppDeg.}POLY_{Num}(x) - \text{UppDeg.}POLY_{Den}(x)}$$

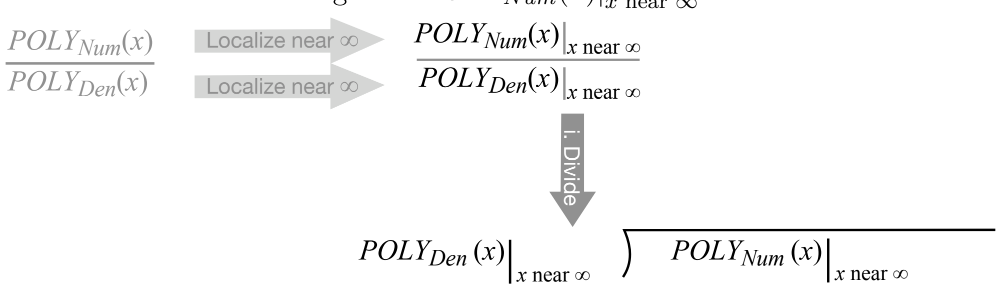
$$= \frac{\text{coef. } Princ.TERM_{Num}(x)\Big|_{x \text{ near } \infty}}{\text{coef. } Princ.TERM_{Den}(x)\Big|_{x \text{ near } \infty}} \cdot x^{\text{RatDeg}.RAT(x)}$$

The resulting monomial is $Princ.TERM\, RAT(x)\Big|_{x \text{ near } \infty}$, that is the *principal term* of the rational function RAT near ∞:

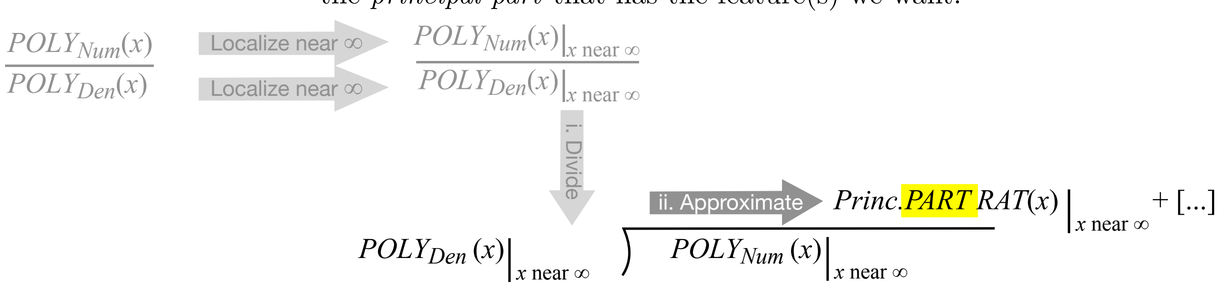


■ The *long route* to $Princ.PART\ RAT(x)\ |_{x \text{ near } \infty}$:

i. In order to divide $POLY_{Num}(x)|_{x \text{ near } \infty}$ *by* $POLY_{Den}(x)|_{x \text{ near } \infty}$, we set up the division as a *long division*, that is $POLY_{Den}(x)|_{x \text{ near } \infty}$ dividing *into* $POLY_{Num}(x)|_{x \text{ near } \infty}$:



ii. We approximate by stopping the long division as soon as we have the *principal part* that has the feature(s) we want:



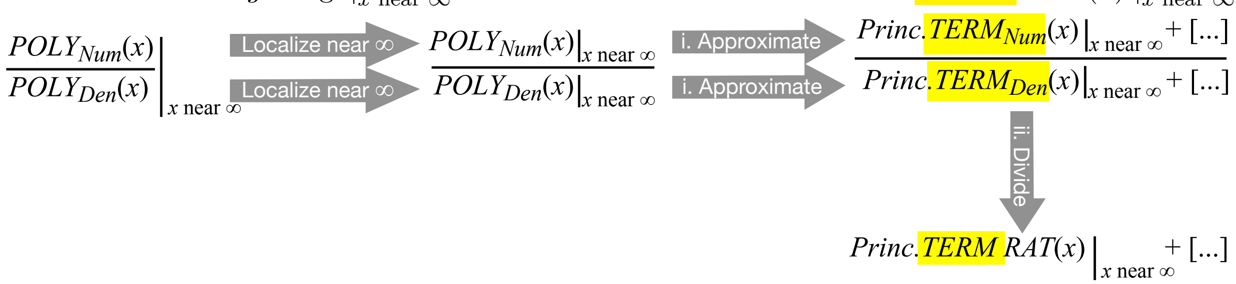
3. Which route we will take in each particular case will depend both on the *wanted feature(s)* near ∞ and on the *rational degree* of RAT and so we will now look separately at how we get $Height$-sign$|_{x \text{ near } \infty}$, $Slope$-sign$|_{x \text{ near } \infty}$ and $Concavity$-sign$|_{x \text{ near } \infty}$

LOCAL ANALYSIS NEAR ∞

When the wanted features are to be found near ∞, the *rational degree* of the rational function tells us up front whether or not the *short route* will allow us to extract the term that controls the wanted feature.

2 Height-sign Near ∞

No matter what the *rational degree* of the given rational function RAT, $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will give us $Height$-sign$|_{x \text{ near } \infty}$ because, no matter what its exponent, *any* power function has $Height$-sign$|_{x \text{ near } \infty}$. So, no matter what the *rational degree* of RAT, to *extract* the term responsible for $Height$-sign$|_{x \text{ near } \infty}$ we can take the *short route* to $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$:

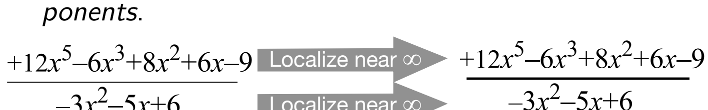


EXAMPLE 13.1. **G** iven the rational function $DOUGH$ specified by the global input-output rule

$$x \xrightarrow{DOUGH} DOUGH(x) = \frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9}{-3x^2 - 5x + 6}$$

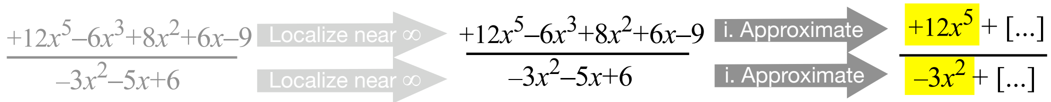
find $Height$-sign $DOUGH|_{x \text{ near } \infty}$.

a. We localize both the numerator and the denominator near ∞—which amounts only to making sure that the terms are in *descending order of exponents*.



b. Inasmuch as $Princ.TERM\ DOUGH(x)\ |_{x \text{ near } \infty}$ has *Height* no matter what the degree, in order to *extract* the term that controls $Height$-sign$|_{x \text{ near } \infty}$ we take the short route to $Princ.TERM\ DOUGH(x)\ |_{x \text{ near } \infty}$:

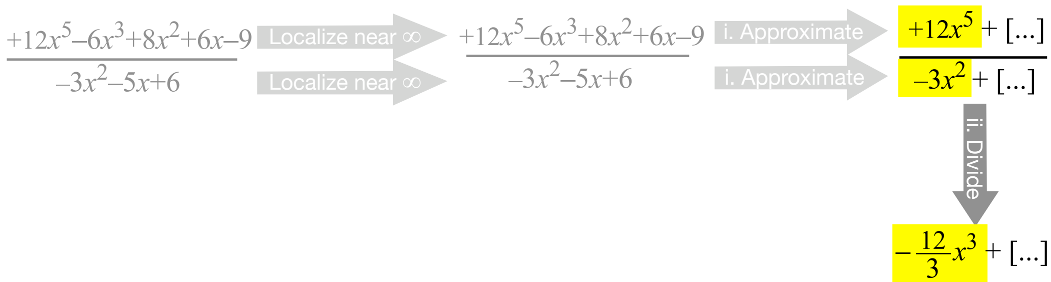
i. We *approximate*



that is we approximate

- the numerator $+12x^5 - 6x^3 + 8x^2 + 6x - 9$ to its *principal term*, $-12x^5$
- the denominator $-3x^2 - 5x + 6$ to its *principal term*, $-3x^2$

ii. And then we *divide*:



where

$$\frac{+12x^5}{-3x^2} = \frac{+12 \cdot x \cdot x \cdot x \cdot x \cdot x}{-3 \cdot x \cdot x} = -\frac{12}{3}x^{5-2}$$

The more usual way to write all this is something as follows:

$$x|_{x \text{ near } \infty} \xrightarrow{DOUGH} DOUGH(x)|_{x \text{ near } \infty} = \left.\frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9}{-3x^2 - 5x + 6}\right|_{x \text{ near } \infty}$$
$$= \frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}{-3x^2 - 5x + 6|_{x \text{ near } \infty}}$$
$$= \frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9}{-3x^2 - 5x + 6}$$
$$= \frac{+12x^5 + [...]}{-3x^2 + [...]}$$
$$= -\frac{12}{3}x^{5-2} + [...]$$

Whatever we write, the *principat term* of $DOUGH$ near ∞ is $-\frac{12}{3}x^3$ and it gives

$$Height\text{-sign } DOUGH|_{x \text{ near } \infty} = (-, +)$$

EXAMPLE 13.2. **G** iven the function PAC specified by the global input-output rule

$$x \xrightarrow{PAC} PAC(x) = \frac{-12x^3 + 7x + 4}{+4x^5 - 6x^4 - 17x^2 - 2x + 10}$$

find Height-sign $PAC|_{x \text{ near } \infty}$.

Inasmuch as $Princ.TERM\ PAC(x)\ |_{x \text{ near } \infty}$ has *Height* no matter what the degree, in order to *extract* the term that controls $Height$-sign$|_{x \text{ near } \infty}$ we take the short route to $Princ.TERM\ DOUGH(x)\ |_{x \text{ near } \infty}$:

$$\begin{aligned} x|_{x \text{ near } \infty} \xrightarrow{PAC} PAC(x)|_{x \text{ near } \infty} &= \left.\frac{-12x^3 + 7x + 4}{+4x^5 - 6x^4 - 17x^2 - 2x + 10}\right|_{x \text{ near } \infty} \\ &= \frac{-12x^3 + 7x + 4|_{x \text{ near } \infty}}{+4x^5 - 6x^4 - 17x^2 - 2x + 10|_{x \text{ near } \infty}} \\ &= \frac{-12x^{+3} + [...]}{+4x^{+5} + [...]} \\ &= \frac{-12}{+4} x^{+3 \ominus +5} + [...] \\ &= -3x^{-2} + [...] \end{aligned}$$

and we get that

$$\text{Height-sign } PAC|_{x \text{ near } \infty} = (-, -)$$

3 Slope-sign Near ∞

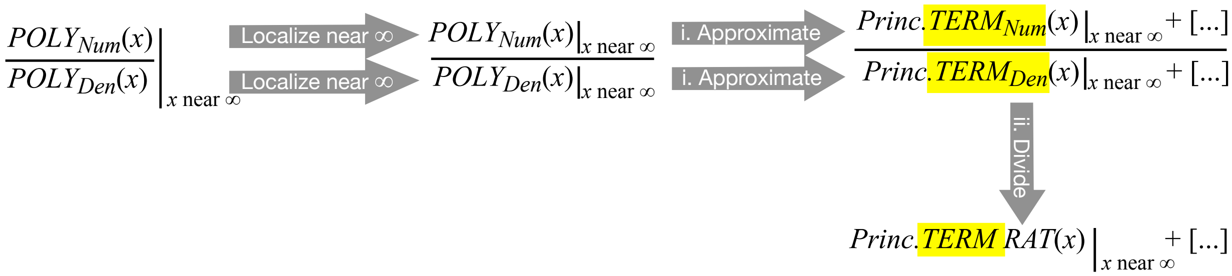
In the case of $Slope$-sign $RAT|_{x \text{ near } \infty}$, there are two cases depending on the *rational degree* of the given rational function:

- If the rational functionRAT is either:
  - A *regular* rational function, that is of rational degree > 1 or < 0

  or

  - An *exceptional* rational function of rational degree $= 1$,

  that is *not* an exceptional rational function of rational degree $= 0$, then $Princ.\ TERM\ RAT(x)\Big|_{x \text{ near } \infty}$ will be a *power function* that will have *Slope* near ∞ and so in order to *extract* the term that controls $Slope$-sign$|_{x \text{ near } \infty}$ we take the short route to $Princ.\ TERM\ RAT(x)\Big|_{x \text{ near } \infty}$:



EXAMPLE 13.3. **G** iven the rational function $SOUTH$ specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{-3x^2 - 5x + 6}{+12x^4 - 6x^3 + 8x^2 + 6x - 9}$$

find Slope-sign of $SOUTH$ near ∞

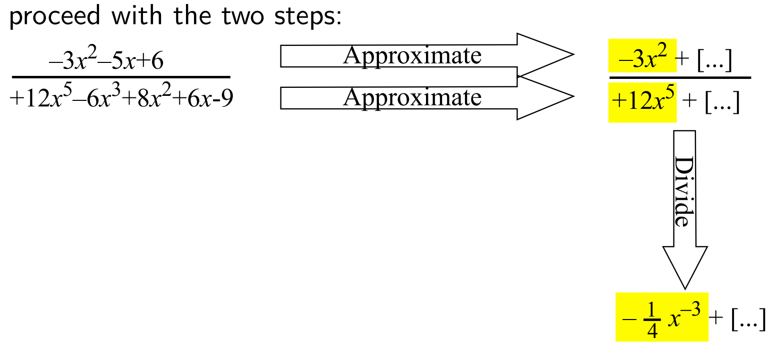
i. We get the local graph near ∞ of $SOUTH$

a. We have

$$x|_{x \text{ near } \infty} \xrightarrow{SOUTH} SOUTH(x)|_{x \text{ near } \infty} = \left.\frac{-3x^2 - 5x + 6}{+12x^5 - 6x^3 + 8x^2 + 6x - 9}\right|_{x \text{ near } \infty}$$

$$= \frac{-3x^2 - 5x + 6|_{x \text{ near } \infty}}{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}$$

We now proceed with the two steps:



b. The more usual presentation is:

$$x|_{x \text{ near } \infty} \xrightarrow{SOUTH} SOUTH(x)|_{x \text{ near } \infty} = \left.\frac{-3x^2 - 5x + 6}{+12x^5 - 6x^3 + 8x^2 + 6x - 9}\right|_{x \text{ near } \infty}$$

$$= \frac{-3x^2 - 5x + 6|_{x \text{ near } \infty}}{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}$$

We *approximate* $-3x^2 - 5x + 6|_{x \text{ near } \infty}$ and $+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}$

$$= \frac{-3x^2 + [...]}{+12x^5 + [...]}$$

and then we *divide*:

$$= \frac{-3}{+12} x^{2-5} + [...]$$
$$= -\frac{1}{4} x^{-3} + [...]$$

c. Since the degree of the power function

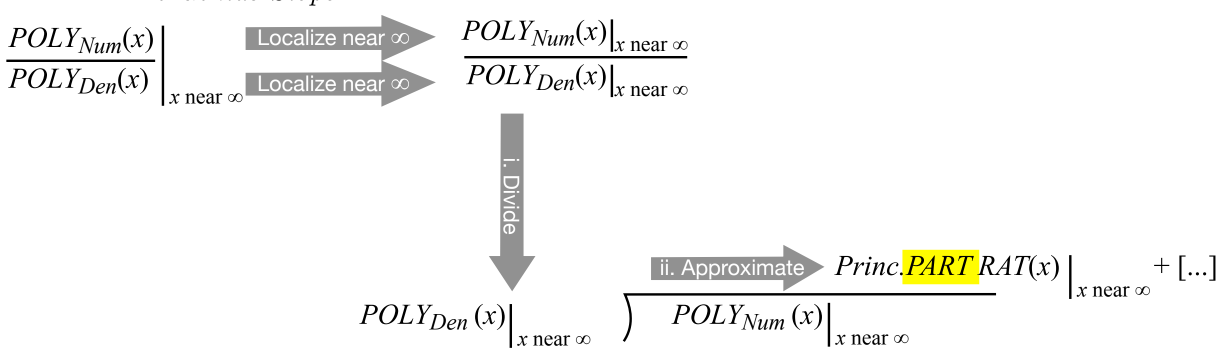
$$x \xrightarrow{POWER} POWER(x) = -\frac{1}{4} x^{-3}$$

which approximates $SOUTH$ near ∞ is < 0, the power function $POWER$ has all three features, *concavity*, *slope* and *height*. (This was of course to be expected from the fact that the *rational degree* of $SOUTH$ is < 0.)

ii. We get

$$\text{Slope-sign of } SOUTH \text{ near } \infty = (\diagup, \diagdown)$$

■ If the rational function RAT *is* an *exceptional rational function* whose rational degree $= 0$, then $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will be an *exceptional power function* with exponent $= 0$ and $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will *not* have *Slope* and so in order to *extract* the term that controls $Slope\text{-sign}|_{x \text{ near } \infty}$ we will have to take the long route to a $Princ.PART\ RAT(x)\ |_{x \text{ near } \infty}$ that *has Slope*:

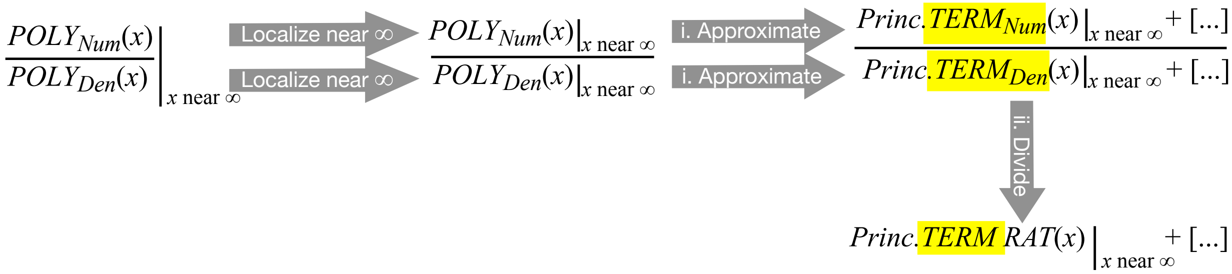


4 Concavity-sign Near ∞

In the case of $Concavity$-sign $RAT|_{x \text{ near } \infty}$, there are *two* cases depending on the rational degree of the given rational function.

■ If the rational function RAT is a *regular* rational function, that is if the rational degree of RAT is either > 1 or < 0, then $Princ.\ TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will be a *regular power function*, that is a power function whose exponent is either > 1 or < 0 and then, in either case, $Princ.\ TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will have *Concavity* and so in order to *extract* the term that controls

$Concavity\text{-sign}|_{x \text{ near } \infty}$ we take the short route to $Princ.TERM_{Den}(x)\,|_{x \text{ near } \infty}$:



EXAMPLE 13.4. G iven the rational function $SOUTH$ specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{-3x^2 - 5x + 6}{+12x^4 - 6x^3 + 8x^2 + 6x - 9}$$

find Concavity-sign of $SOUTH$ near ∞

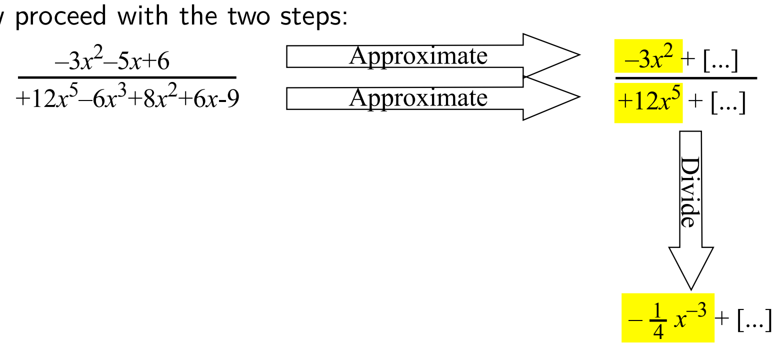
i. We get the local graph near ∞ of $SOUTH$

a. We have

$$x|_{x \text{ near } \infty} \xrightarrow{SOUTH} SOUTH(x)|_{x \text{ near } \infty} = \left.\frac{-3x^2 - 5x + 6}{+12x^5 - 6x^3 + 8x^2 + 6x - 9}\right|_{x \text{ near } \infty}$$

$$= \frac{-3x^2 - 5x + 6|_{x \text{ near } \infty}}{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}$$

We now proceed with the two steps:



b. The more usual presentation is:

$$x|_{x \text{ near } \infty} \xrightarrow{SOUTH} SOUTH(x)|_{x \text{ near } \infty} = \left.\frac{-3x^2 - 5x + 6}{+12x^5 - 6x^3 + 8x^2 + 6x - 9}\right|_{x \text{ near } \infty}$$

$$= \frac{-3x^2 - 5x + 6|_{x \text{ near } \infty}}{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}$$

We *approximate* $-3x^2 - 5x + 6|_{x \text{ near } \infty}$ and $+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}$

$$= \frac{-3x^2 + [...]}{+12x^5 + [...]}$$

and then we *divide*:

$$= \frac{-3}{+12} x^{2-5} + [...]$$
$$= -\frac{1}{4} x^{-3} + [...]$$

c. Since the degree of the power function

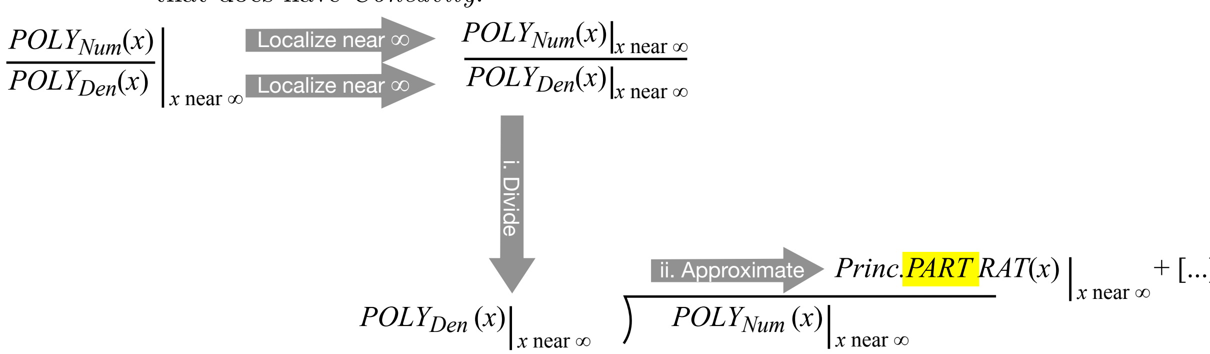
$$x \xrightarrow{POWER} POWER(x) = -\frac{1}{4} x^{-3}$$

which approximates $SOUTH$ near ∞ is < 0, the power function $POWER$ has all three features, *concavity*, *slope* and *height*. (This was of course to be expected from the fact that the *rational degree* of $SOUTH$ is < 0.)

ii. We get

$$\text{Concavity-sign of } SOUTH \text{ near } \infty = (\cap, \cap)$$

■ If the rational function RAT is an *exceptional* rational function that is if the rational degree of RAT is either $= 1$ or $= 0$ then $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will be an *exceptional power function* with exponent either $= 1$ or $= 0$ (**Chapter 7**) and in both cases $Princ.TERM\ RAT(x)\ |_{x \text{ near } \infty}$ will *not* have *Concavity* and in order *extract* the term that controls $Concavity\text{-sign}|_{x \text{ near } \infty}$ we will have to take the long route to a $Princ.PART\ RAT(x)\ |_{x \text{ near } \infty}$ that does have *Concavity*.

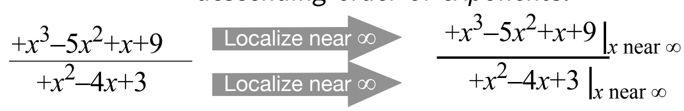


EXAMPLE 13.5. **G** iven the rational function $BATH$ specified by the global input-output rule

$$x \xrightarrow{BATH} BATH(x) = \frac{+x^3 - 5x^2 + x + 6}{+x^2 - 4x + 3}$$

find $Concavity$-sign $BATH|_{x \text{ near } \infty}$.

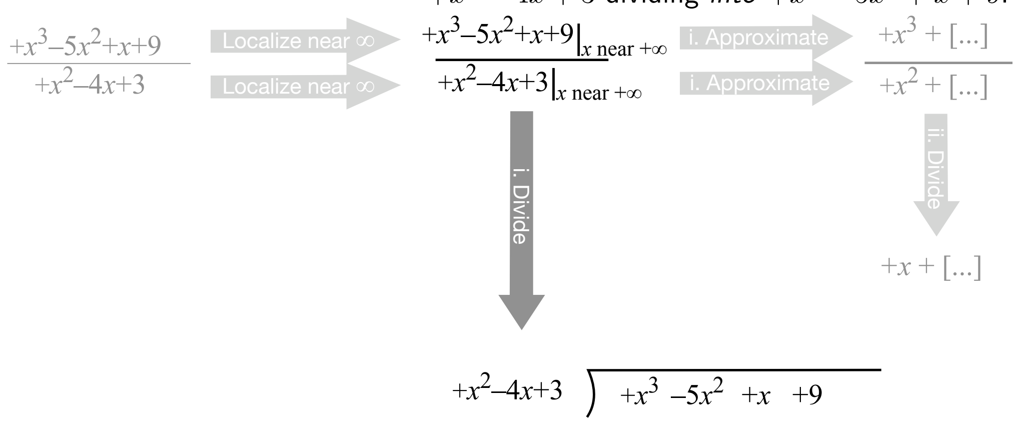
a. The *localization step* is to *localize* both the numerator and the denominator near ∞—which amounts only to making sure that the terms are in *descending order of exponents*.



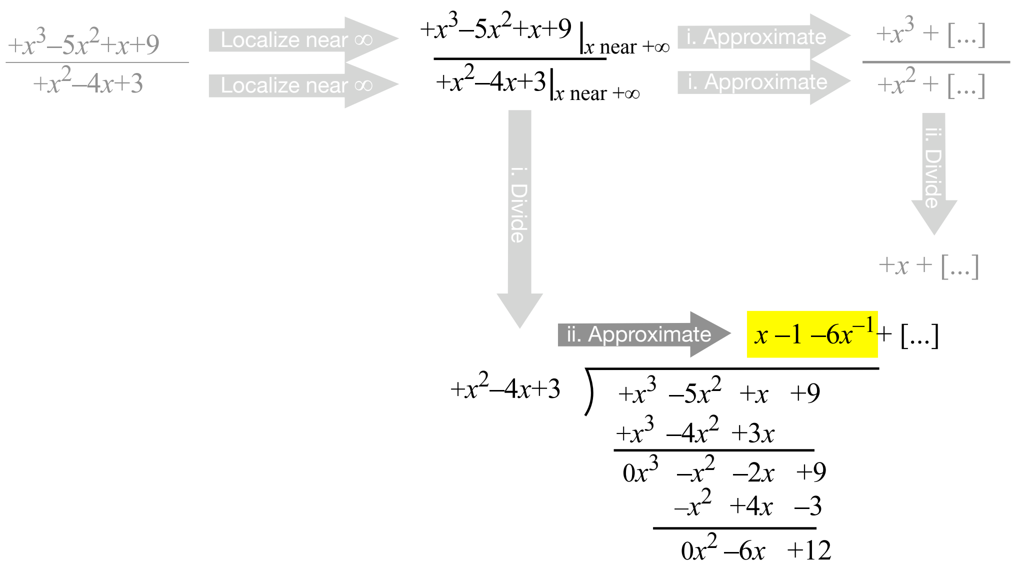
b. Since $Princ.\, TERM\; BATH(x)\,|_{x \text{ near } \infty}$ has no *Concavity*, the *extraction step* to get $Concavity$-sign $BATH|_{x \text{ near } \infty}$ must take the long route to a $Princ.\, PART\; BATH(x)\,|_{x \text{ near } \infty}$ that has *Concavity*:

i. We set up the division as a *long division*:

$$+x^2 - 4x + 3 \text{ dividing } \textit{into } +x^3 - 5x^2 + x + 9.$$



ii. We *approximate* by stopping the long division as soon as we have the *principal part* of the quotient that has *Concavity*:



that is we stop with $-6x^{-1}$ since it is the term responsible for *Concavity*. The more usual way to write all this is:

$$
\begin{aligned}
x|_{x \text{ near } \infty} \xrightarrow{BATH} BATH(x)|_{x \text{ near } \infty} &= \left.\frac{+x^3 - 5x^2 + x + 9}{+x^2 - 4x + 3}\right|_{x \text{ near } \infty} \\
&= \frac{+x^3 - 5x^2 + x + 9|_{x \text{ near } \infty}}{+x^2 - 4x + 3|_{x \text{ near } \infty}} \\
&= \frac{+x^3 - 5x^2 + x + 9}{+x^2 - 4x + 3}
\end{aligned}
$$

and then we *divide* (in the *latin* manner):

$$
\begin{array}{rrrrrr}
 & & +x & -1 & -6x^{-1} & +[...] \\
\hline
+x^2 \; -4x \; +3 &) & +x^3 & -5x^2 & +x & +9 \\
 & & +x^3 & -4x^2 & +3x & \\
\hline
 & & 0x^3 & -x^2 & -2x & +9 \\
 & & & -x^2 & +4x & -3 \\
\hline
 & & & 0x^2 & -6x & +12
\end{array}
$$

Whichever way we write it, $Princ.\ PART\ BATH(x)\ |_{x \text{ near } \infty} = +x - 1 - 6x^{-1}$ and its third term, $-6x^{-1}$, gives

$$Concavity\text{-sign } BATH|_{x \text{ near } \infty} = (\cap, \cup)$$

5 Local Graph Near ∞

In order to get the local graph near ∞, we need a local input-output rule that gives us the *concavity*-signÑand therefore the *slope*-sign and the *height*-sign.

So, the route we must take in order to get the local graph near ∞ is the route that will get us the concavity-sign near ∞.

EXAMPLE 13.6. G iven the rational function $SOUTH$ whose global input-output rule is

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{-3x^2 - 5x + 6}{+12x^4 - 6x^3 + 8x^2 + 6x - 9}$$

find its local graph near ∞.

i. We get the *local input-output rule* near ∞ as in EXAMPLE 1.

We have:

$$x|_{x \text{ near } \infty} \xrightarrow{SOUTH} SOUTH(x)|_{x \text{ near } \infty} = \left.\frac{-3x^2 - 5x + 6}{+12x^5 - 6x^3 + 8x^2 + 6x - 9}\right|_{x \text{ near } \infty}$$

$$= \frac{-3x^2 - 5x + 6|_{x \text{ near } \infty}}{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}$$

We *approximate* separately the *numerator* and the *denominator*:

$$= \frac{-3x^2 + [...]}{+12x^5 + [...]}$$

We *divide* the approximations:

$$= \frac{-3}{+12}x^{2-5} + [...]$$

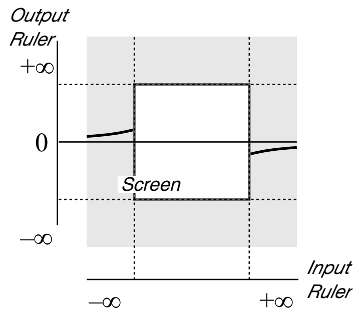
$$= -\frac{1}{4}x^{-3} + [...]$$

ii. Since the degree of the power function

$$x \xrightarrow{POWER} POWER(x) = -\frac{1}{4}x^{-3}$$

is < 0, the power function $POWER$ is *regular* and has both *concavity* and *slope*. So, the local graph of the power function $POWER$ near ∞ will be approximately the graph near ∞ of the rational function $SOUTH$.

The local graph near ∞ of the rational function $SOUTH$ is therefore:



EXAMPLE 13.7. G iven the rational function $DOUGH$ whose global input-output rule is

$$x \xrightarrow{DOUGH} DOUGH(x) = \frac{+12x^4 - 6x^3 + 8x^2 + 6x - 9}{-3x^2 - 5x + 6}$$

find its local graph near ∞.

i. We get the *local input-output rule* near ∞.

We have:

$$x|_{x \text{ near } \infty} \xrightarrow{DOUGH} DOUGH(x)|_{x \text{ near } \infty} = \left.\frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9}{-3x^2 - 5x + 6}\right|_{x \text{ near } \infty}$$

$$= \frac{+12x^5 - 6x^3 + 8x^2 + 6x - 9|_{x \text{ near } \infty}}{-3x^2 - 5x + 6|_{x \text{ near } \infty}}$$

We *approximate* separately the *numerator* and the *denominator*:

$$= \frac{+12x^5 + [...]}{-3x^4 + [...]}$$

We *divide* the approximations:

$$= -\frac{+12}{-3}x^{5-2} + [...]$$

$$= -4x^{+3} + [...]$$

ii. Since the degree of the power function

$$x \xrightarrow{POWER} POWER(x) = -4x^{+3}$$

is > 1, the power function $POWER$ is *regular* and has both *concavity* and *slope*. So, the local graph of the power function $POWER$ near ∞ will be approximately the graph near ∞ of the rational function $DOUGH$.

The local graph near ∞ of the rational function $DOUGH$ is therefore:

EXAMPLE 13.8. **G** iven the rational function $BATH$ specified by the global input-output rule

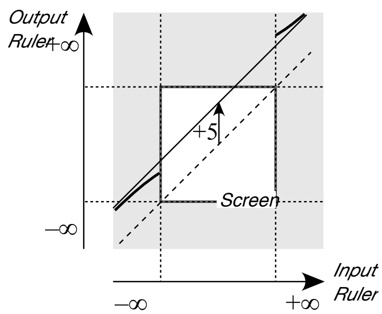
$$x \xrightarrow{BATH} BATH(x) = \frac{+x^3 + x^2 - 5x + 6}{+x^2 - 4x + +3}$$

as in EXAMPLE 1, find the local graph near ∞.

i. We get the *local input-output rule* near ∞ that gives all three features as we did in EXAMPLE 1:

$x|_{x \text{ near } \infty} \xrightarrow{BATH} BATH(x)|_{x \text{ near } \infty} = +x + 5 + 27x^{-1} + [...]$

ii. So the local graph near ∞ of the function $BATH$ is



??
??
??
??
??
??
??
??
??
??
??
??
??
??
==================================
??
??
??
??
??
??

??
??
??
??
==================================
??
??
??
??
??
??
??
??
??
??
==================================
??
??
??
??
??
??
??
??
??
??
==================================
??
??
??
??
??
??
??
??
??
??
==================================
??
??

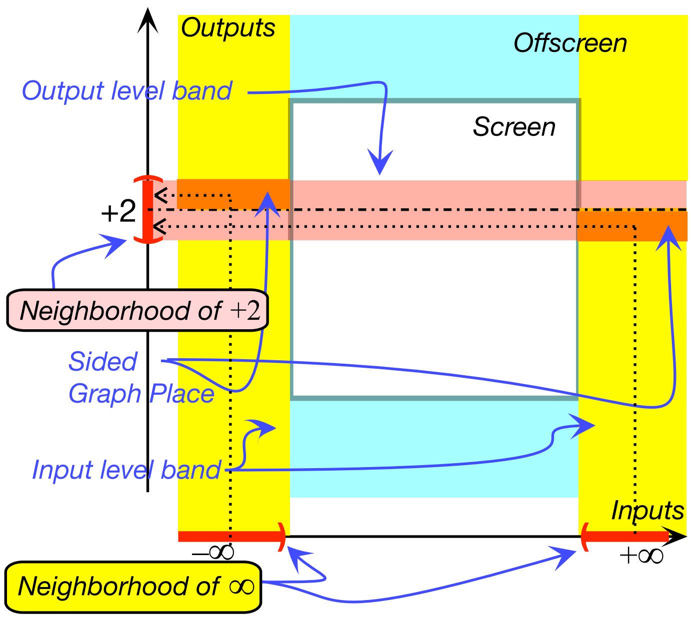
??
??
??
??
??
??
??
??
================================

Chapter 14

Rational Functions: Local Analysis Near x_0

Doing local analysis means working in a neighborhood of some given input and thus counting inputs from the given input since it is the *center* of the neighborhood. When the given input is x_0, we *localize* at x_0, that is we set $x = x_0 + h$ where h is *small* and we compute with powers of h in descending order of sizes.

Example 14.1. **G** iven the input $+2$, then the location of the number $+2.3$ relative to $+2$ is $+0.3$:



Recall that the *principal part* near x_0 of a given polynomial function $POLY$ is the local quadratic part

$$x|_{x \text{ near } x_0} \xrightarrow{POLY} POLY(x)|_{x \text{ near } x_0} = \Big[\quad\Big] + \Big[\quad\Big]h + \Big[\quad\Big]h^2 + [...]$$

However, the complication here is that to get the principal part near x_0 of a rational function we must approximate the two polynomial and divide—or the other way round—and the result need not be a polynomial but can also be a negative-exponent power function and the main issue will be whether to do the approximation before or after the division.

1 Local Input-Output Rule Near x_0

Given a rational function RAT, we look for the function whose input-output rule will be simpler than the input-output rule of RAT but whose local graph near x_0 will be qualitatively the same as the local graph near x_0 of RAT.

More precisely, given a rational function RAT specified by the global input-output rule

$$x \xrightarrow{RAT} RAT(x) = \frac{POLY_{Num}(x)}{POLY_{Den}(x)}$$

what we will want then is an *approximation* for the output of the local input-output rule near x_0

$$x\big|_{x \text{ near } x_0} \xrightarrow{RAT} RAT(x)\big|_{x \text{ near } x_0} = \left.\frac{POLY_{Num}(x)}{POLY_{Den}(x)}\right|_{x \text{ near } x_0}$$

from which to *extract* whatever controls the wanted feature.

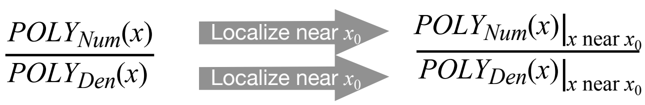
1. Since the center of the neighborhood is x_0, we *localize* both

- $POLY_{Num}(x)$

and

- $POLY_{Den}(x)$

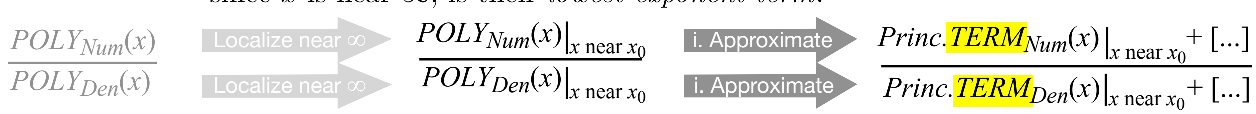
by letting $x \leftarrow x_0+h$ and writing the terms in *ascending* order of exponents.



2. Depending on the circumstances, we will take one of the following two routes to *extract* what controls the wanted feature:

■ The *short route* to $Princ.TERM\ RAT(x)\ |_{x \text{ near } x_0}$, that is:

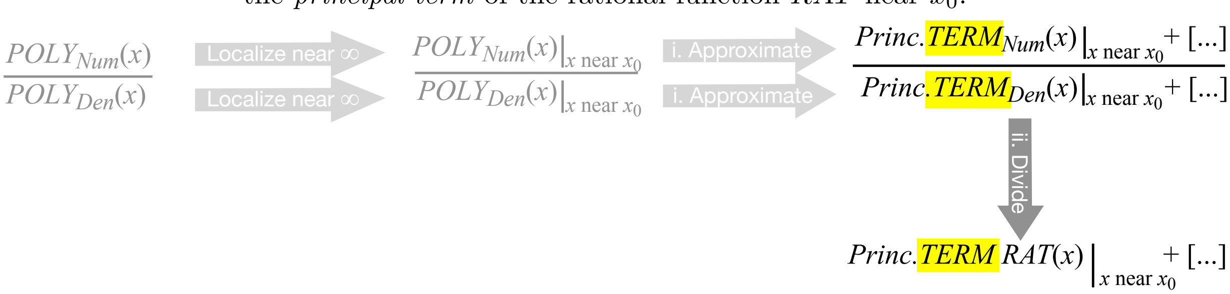
i. We approximate both $POLY_{Num}(x)|_{x \text{ near } x_0}$ and $POLY_{Den}(x)|_{x \text{ near } x_0}$ to their *principal term*—that is to just their *lowest size term*—which, since x is near ∞, is their *lowest exponent term*:



ii. In order to divide $Princ.TERM_{Num(x)}\Big|_{x \text{ near } x_0}$, that is the principal term near x_0 of the *numerator* of RAT by $Princ.TERM_{Den(x)}\Big|_{x \text{ near } x_0}$, that is the principal term near x_0 of the *denominator* of RAT we use monomial division

$$\boxed{\frac{ah^{+m}}{bh^{+n}} = \frac{a}{b}h^{+m\ominus+n}}$$ where $+m \ominus +n$ can turn out positive, negative or 0

The resulting monomial is $Princ.TERM\,RAT(x)\Big|_{x \text{ near } x_0}$, that is the *principal term* of the rational function RAT near x_0.



However, $Princ.TERM\, RAT(x)\Big|_{x \text{ near } x_0}$ is useful only in four cases:

- When it is a constant term *and* what we want is the Height-sign,
- When it is a linear term *and* what we want is the Height-sign or the Slope-sign,
- When it is a square term,
- When it is a negative-exponent term.

■ The *long route* to $Princ.PART\ RAT(x)\ |_{x \text{ near } x_0}$:

i. In order to divide $POLY_{Num}(x)|_{x \text{ near } x_0}$ *by* $POLY_{Den}(x)|_{x \text{ near } x_0}$, we set up the division as a *long division*, that is $POLY_{Den}(x)|_{x \text{ near } x_0}$ dividing *into* $POLY_{Num}(x)|_{x \text{ near } x_0}$ and since these are polynomials in h, in order to be in order of descending sizes, they must be in order of ascending exponents.

ii. We approximate by stopping the long division as soon as we have the *principal part* that has the feature(s) we want:

iii. The difficulty will be that we will have to approximate at two different stages:

- While we localize both the numerator and the denominator,
- When we divide the approximate localization of the numerator by the approximate localization of the denominator

So, we will have to make sure that the approximations in the localizations of the numerator and the denominator do not interfere with the approximation in the division, that is that, as we divide, we do not want to bump into a [...] coming from having approximated the numerator and the denominator too much, that is before we can extract from the division the term that controls the wanted feature.

3. Which route we will take in each particular case will depend both on the *wanted feature*(s) near x_0 and so we will now look separately at how we get *Height*-sign$|_{x \text{ near } \infty}$, *Slope*-sign$|_{x \text{ near } x_0}$ and *Concavity*-sign$|_{x \text{ near } x_0}$

LOCAL ANALYSIS NEAR x_0

When the wanted features are to be found near x_0, the *rational degree* of the rational function does not tell us which of the *short route* or the *long route* will allow us to extract the term that controls the wanted feature.

2 Height-sign Near x_0

If all we want is the Height-sign, then we can always go the short route.

EXAMPLE 14.2. **L** et $SOUTH$ be the function specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}$$

Find the height-sign of $SOUTH$ near $+2$

i. We localize both the numerator of $SOUTH$ and the denominator of $SOUTH$ near $+2$

$$\begin{aligned} h \xrightarrow{SOUTH_{+2}} SOUTH(+2+h) &= \left.\frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}\right|_{x\leftarrow +2+h} \\ &= \frac{x^2+5x+6|_{x\leftarrow +2+h}}{x^4-x^3-10x^2+x-15|_{x\leftarrow +2+h}} \\ &= \frac{(+2+h)^2+5(+2+h)+6}{(+2+h)^4-(+2+h)^3-10(+2+h)^2+(+2+h)-15} \end{aligned}$$

ii. Since we want the local input-output rule that will give us the height-sign, we try to approximate *before* we divide:

$$\begin{aligned} &= \frac{\Big[(+2)^2+5\cdot(+2)+6\Big]+[...]}{\Big[(+2)^4-(+2)^3-10(+2)^2+2-15\Big]+[...]} \\ &= \frac{\Big[+4+10+6\Big]+[...]}{\Big[+16-8-40+2-15\Big]+[...]} \\ &= \frac{+20+[...]}{-45+[...]} \\ &= -\frac{20}{45}+[...] \end{aligned}$$

and since the approximate local input-output rule near $+2$ is

$$h \xrightarrow{SOUTH_{+2}} SOUTH(+2+h) = -\frac{20}{45}+[...]$$

and the local input-output rule includes the term that gives the Height-sign near $+2$

$$-\frac{20}{45}$$

we have:

$$\text{Height-sign } SOUTH \text{ near } +2 = (-.-)$$

EXAMPLE 14.3. **L** et $SOUTH$ be the function specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}$$

Find the height-sign of $SOUTH$ near -3

i. We localize both the numerator of $SOUTH$ and the denominator of $SOUTH$ near -3

$$h \xrightarrow{SOUTH_{-3}} SOUTH(-3+h) = \left.\frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}\right|_{x\leftarrow -3+h}$$

$$= \frac{x^2+5x+6|_{x\leftarrow -3+h}}{x^4-x^3-10x^2+x-15|_{x\leftarrow -3+h}}$$

$$= \frac{(-3+h)^2+5(-3+h)+6}{(-3+h)^4-(-3+h)^3-10(-3+h)^2+(-3+h)-15}$$

ii. Since we want the local input-output rule that will give us the height-sign, we try to approximate to the constant terms:

$$= \frac{\Big[(-3)^2+5\cdot(-3)+6\Big]+[...]}{\Big[(-3)^4-(-3)^3-10(-3)^2-3-15\Big]+[...]}$$

$$= \frac{\Big[+9-15+6\Big]+[...]}{\Big[+81+27-90-3-15\Big]+[...]}$$

$$= \frac{\Big[0\Big]+[...]}{\Big[0\Big]+[...]}$$

We cannot divide as we could get

$$= any\, size$$

iii. We therefore must approximate the localizations at least to h

$$= \frac{\Big[0\Big]+\Big[2\cdot(-3)+5\Big]h+[...]}{\Big[0\Big]+\Big[+4(-3)^3-3(-3)^2-10\cdot 2(-3)+1\Big]h+[...]}$$

$$= \frac{\Big[-6+5\Big]h+[...]}{\Big[-108-27+60+1\Big]h+[...]}$$

$$= \frac{\Big[-1\Big]h+[...]}{\Big[-74\Big]h+[...]}$$

$$= \frac{-h+[...]}{-74h+[...]}$$

We divide

$$= +\frac{1}{74}+[...]$$

and since the approximate local input-output rule near -3 is

$$h \xrightarrow{SOUTH_{-3}} SOUTH(-3+h) = +\frac{1}{74} + [...]$$

and the local input-output rule includes the term that gives the Height-sign near -3

$$+\frac{1}{74}$$

we have:

$$\text{Height-sign } SOUTH \text{ near } -3 = (+,+)$$

3 Slope-sign Near x_0

EXAMPLE 14.4. **L** et $SOUTH$ be the function specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}$$

find the slope-sign of $SOUTH$ near $+2$

i. We localize both the numerator of $SOUTH$ and the denominator of $SOUTH$ near $+2$ and since we want the approximate local input-output rule for the slope-sign, we will approximate to h:

$$+2+h \xrightarrow{SOUTH} SOUTH(+2+h) = \left.\frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}\right|_{x\leftarrow +2+h}$$

$$= \frac{x^2+5x+6|_{x\leftarrow +2+h}}{x^4-x^3-10x^2+x-15|_{x\leftarrow +2+h}}$$

$$= \frac{(+2+h)^2+5(+2+h)+6}{(+2+h)^4-(+2+h)^3-10(+2+h)^2+(+2+h)-15}$$

$$= \frac{\Big[(+2)^2+5\cdot(+2)+6\Big]+\Big[2(+2)+5\Big]h+[...]}{\Big[(+2)^4-(+2)^3-10\cdot(+2)^2+(+2)-15\Big]+\Big[4(+2)^3-3(+2)^2-10\cdot 2(+2)+1\Big]h+[...]}$$

$$= \frac{\Big[+20\Big]+\Big[+9\Big]h+[...]}{\Big[-45\Big]+\Big[-19\Big]h+[...]}$$

ii. We set up the division with

$[-45]+[-19]h+[...]$ dividing *into* $[+20]+[+9]h+[...]$

that is:

$$\begin{array}{rllll}
 & -\frac{20}{45} & -\frac{[9\cdot45]-[19\cdot20]}{45^2}h & +[...] & \\
\hline
-45\ -19h\ +[...]\ \big) & +20 & +9h & +[...] & \\
 & +20 & +\frac{19\cdot20}{45}h & +[...] & \\
\hline
 & 0 & +\frac{[9\cdot45]-[19\cdot20]}{45}h & +[...] &
\end{array}$$

And since $[9 \cdot 45] - [19 \cdot 20] = 405 - 380 = +25$, the approximate local input-output rule near $+2$ is:

$$h \xrightarrow{SOUTH_{+2}} SOUTH(+2+h) = -\frac{20}{45} - \frac{25}{45^2}h + [...]$$

and the term that gives the slope-sign near $+2$ is

$$-\frac{25}{45^2}h$$

so that

$$\text{Slope-sign } SOUTH \text{ near } +2 = (\searrow . \searrow)$$

4 Concavity-sign Near x_0

EXAMPLE 14.5. **L** et $SOUTH$ be the function specified by the global input-output rule

$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}$$

find the concavity-sign of $SOUTH$ near $+2$

i. We localize both the numerator of $SOUTH$ and the denominator of $SOUTH$ near $+2$ and since we want the approximate local input-output rule for the slope-sign, we will approximate to h^2:

$$+2+h \xrightarrow{SOUTH} SOUTH(+2+h) = \left.\frac{x^2+5x+6}{x^4-x^3-10x^2+x-15}\right|_{x\leftarrow+2+h}$$

$$= \frac{x^2+5x+6|_{x\leftarrow+2+h}}{x^4-x^3-10x^2+x-15|_{x\leftarrow+2+h}}$$

$$= \frac{(+2+h)^2+5(+2+h)+6}{(+2+h)^4-(+2+h)^3-10(+2+h)^2+(+2+h)-15}$$

$$= \frac{\Big[(+2)^2+5\cdot(+2)+6\Big]+\Big[2(+2)+5\Big]h+\Big[1\Big]h^2}{\Big[(+2)^4-(+2)^3-10\cdot(+2)^2+(+2)-15\Big]+\Big[4(+2)^3-3(+2)^2-10\cdot2(+2)+1\Big]h+\Big[6(+2)^2-3(+2)-10\Big]h^2+}$$

$$= \frac{\Big[+20\Big]+\Big[+9\Big]h+\Big[1\Big]h^2}{\Big[-45\Big]+\Big[-19\Big]h+\Big[8\Big]h^2+[...]}$$

ii. We set up the division with

$$-45 + -19h + 8h^2 + [...] \quad \text{dividing } \textit{into} \quad +20 + 9h + h^2$$

but carry it out *latin style* (that is, we *write* the *result* of the multiplication as it comes out instead of the *opposite of the result*.)

$$\begin{array}{r|lllll}
 & -\frac{20}{45} & -\frac{[9\cdot45]-[19\cdot20]}{45^2}h & -\left[\frac{+45[45-8\cdot20]-19[[9\cdot45]-[19\cdot20]]}{45^3}\right]h^2 & +[...] \\
\hline
-45 \; -19h \; +8h^2 \; +[...] & +20 & +9h & +h^2 & \\
 & +20 & +\frac{19\cdot20}{45}h & -\frac{8\cdot20}{45}h^2 & +[...] \\
\cline{2-5}
 & 0 & +\frac{[9\cdot45]-[19\cdot20]}{45}h & +\frac{45-8\cdot20}{45}h^2 & +[...] \\
 & & +\frac{[9\cdot45]-[19\cdot20]}{45^2}h & +19\frac{[9\cdot45]-[19\cdot20]}{45^2}h^2 & +[...] \\
\cline{3-5}
 & & +0h & +\left[\frac{+45[45-8\cdot20]-19[[9\cdot45]-[19\cdot20]]}{45^2}\right]h^2 & +[...]
\end{array}$$

And since $\frac{+45[45-8\cdot20]-19[[9\cdot45]-[19\cdot20]]}{45^2} = -\frac{2401}{45^2}$, the local input-output rule near $+2$ is:

$$h \xrightarrow{SOUTH_{+2}} SOUTH(+2+h) = -\frac{20}{45} - \frac{25}{45^2}h - \frac{2401}{45^2}h^2 + [...]$$

and the term that gives the concavity-sign near $+2$ is

$$-\frac{2401}{45^2}h^2$$

so that

$$\text{Concavity-sign } SOUTH \text{ near } +2 = (\cap, \cap)$$

5 Local Graph Near x_0

EXAMPLE 14.6. **L** et $SOUTH$ be the function specified by the global input-output rule

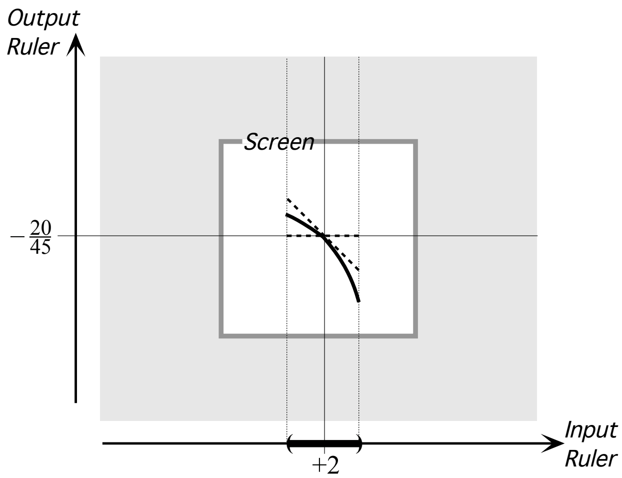
$$x \xrightarrow{SOUTH} SOUTH(x) = \frac{x^2 + 5x + 6}{x^4 - x^3 - 10x^2 + x - 15}$$

find the local graph of $SOUTH$ near $+2$

Since, in order to get the local graph near $+2$ we need all three features near $+2$, height-sign, slope-sign and concavity-sign, we need to get the approximate local input-output rule as we did in the previous example:

$$h \xrightarrow{SOUTH_{+2}} SOUTH(+2+h) = -\frac{20}{45} - \frac{25}{45^2}h - \frac{2401}{45^2}h^2 + [...]$$

from which we get:


Output
Ruler
Screen
$-\frac{20}{45}$
Input
Ruler
+2

==================================
??
??
??
??
??
??
??
??
??
??
??
??
??
??
==================================
??
??
??
??
??
??
??
??
??

??
================================
??
??
??
??
??
??
??
??
??
??
================================
??
??
??
??
??
??
??
??
??
??
================================
??
??
??
??
??
??
??
??
??
??
================================
??
??
??
??
??

??
??
??
??
??
==================================

Chapter 15

Rational Functions: Global Analysis

Contrary to what we were able to do with polynomial functions, with rational functions we will *not* be able to establish global tEXAMPLEs. Of course, we did not really establish global tEXAMPLEs for *all* polynomial functions either but only for polynomial functions *of a given degree*, 0, 1, 2 and 3. But, in the case of rational functions, even the *rational degree* will not separate rational functions into kinds that we can establish global tEXAMPLEs for inasmuch as even rational functions with a given rational degree can be very diverse.

So, what we will do here is to focus on how to get global information about *any given rational function*.

1 The Essential Question

Given a *rational function*, as with any function, the *offscreen graph* will consist:

- certainly of the local graph near ∞. This is because, as soon as the *input* is *large*, the graph point is going to be left or right of the screen and therefore *offscreen* regardless of the size of the *output*,
- possibly of the local graph(s) near certain *bounded input(s)*. This is because, in case the outputs for inputs near certain bounded inputs are

large, the graph points will then be above or below the screen and therefore *offscreen* even though the inputs are *bounded.*

So, as always, we will need to ascertain whether

- There might be *bounded inputs* for which nearby inputs will have a *large* output ,

or, as was the case with all polynomial functions,

- The outputs for any *bounded input* are themselves necessarily *bounded*

In other words, in order to get the *offscreen graph*, we must begin by asking the **Essential Question**:

- Do all *bounded inputs* have *bounded outputs*

or

- Is there one (or more) *bounded input* which is an ∞-height input, that is, a *bounded input* whose nearby inputs have *unbounded outputs*?

And, indeed, we will find that there are two kinds of rational functions:

- rational functions that *do* have ∞-height input(s)
- rational function that *do not* have any ∞-height input as was the case with power functions and polynomial functions.

2 Locating Infinite Height Inputs

However, given a rational function, not only will we need to know whether or not there *exists* ∞-height input(s), if there are any, we will also have to *locate* the ∞-height inputs, if any, because we will need to get the local graph near these ∞-height input(s). More precisely:

1. Given a rational function *RAT* specified by a global input-output rule

$$x \xrightarrow{\ RAT\ } RAT(x) = \frac{NUMERATOR_{RAT}(x)}{DENOMINATOR_{RAT}(x)}$$

we want to find whether or not there can be a *bounded input* x_0 such that the outputs for *nearby* inputs, $x_0 + h$, are *large.* In other words, we want to know if there can be x_0 such that

$$h \xrightarrow{\ RAT\ } RAT(x)|_{x \leftarrow x_0+h} = large$$

But we have

$$RAT(x)|_{x \leftarrow x_0+h} = \left.\frac{NUMERATOR_{RAT}(x)}{DENOMINATOR_{RAT}(x)}\right|_{x \leftarrow x_0+h}$$

$$= \frac{NUMERATOR_{RAT}(x)|_{x \leftarrow x_0+h}}{DENOMINATOR_{RAT}(x)|_{x \leftarrow x_0+h}}$$
$$= \frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)}$$

So, what we want to know is if there can be an x_0 for which

$$\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)} = large$$

2. Since it is a *fraction* that we want to be *large*, we will use the **Division Size TEXAMPLE** from **Chapter 2**:

THEOREM 2 (**Division Size**)

$$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = \boldsymbol{large} \qquad \frac{large}{small} = \boldsymbol{large}$$
$$\frac{medium}{large} = small \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = \boldsymbol{large}$$
$$\frac{small}{large} = small \qquad \frac{small}{medium} = small \qquad \frac{small}{small} = any\ size$$

There are thus two ways that a fraction can be *large*:

- When the numerator is *large*
- When the denominator is *small*

In each case, though, we need to make sure of the other side of the fraction. So, rather than look at the size of both the numerator and the denominator at the same time, we will look separately at: there are two cases where a fraction *could* (but need not) be *large* but in each case we will need to look at the other side of the fraction bar in order to know what the size of the fraction is:

- The first *row*, that is when the *numerator* of the fraction is *large*

  $$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = \boldsymbol{large} \qquad \frac{large}{small} = \boldsymbol{large}$$
  $$\frac{medium}{large} = small \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = \boldsymbol{large}$$
  $$\frac{small}{large} = small \qquad \frac{small}{medium} = small \qquad \frac{small}{small} = any\ size$$

  because in that case all we will then have to do is to make sure that the *denominator* is *not large* too.
- The last *column*, that is when the *denominator* of the fraction is *small*.

$$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = \boldsymbol{large} \qquad \frac{large}{small} = \boldsymbol{large}$$

$$\frac{medium}{large} = small \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = \boldsymbol{large}$$

$$\frac{small}{large} = small \qquad \frac{small}{medium} = small \qquad \frac{small}{small} = any\ size$$

because in that case all we will then have to do is to make sure that the numerator is *not small* too.

3. We now deal with $\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)}$, looking separately at the numerator and the denominator:

- Since the *numerator*, $NUMERATOR_{RAT}(x_0 + h)$, is the output of a *polynomial function*, namely

$$x \xrightarrow{\ NUMERATOR_{RAT}\ } NUMERATOR_{RAT}(x)$$

  and since we have seen that *the only way* the outputs of a *polynomial function* can be *large* is when the inputs are themselves *large*, *there is no way* that $NUMERATOR_{RAT}(x_0+h))$ could be *large* for inputs that are *bounded*. So there is no way that the output of RAT could be large for *bounded* inputs that make the *numerator* large and we need not look any further.
- Since the *denominator*, $DENOMINATOR_{RAT}(x_0 + h)$, is the output of the *polynomial function*

$$x \xrightarrow{\ DENOMINATOR_{RAT}\ } DENOMINATOR_{RAT}(x)$$

  and since we have seen that polynomial functions *can* have *small* outputs if they have 0-height inputs and the inputs are near the 0-height inputs, $DENOMINATOR_{RAT}(x_0+h)$ *can* be *small* for certain bounded inputs and thus so can $\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)}$. However, we will then have to make sure that $NUMERATOR_{RAT}(x_0 + h)$, is *not small* too near these bounded inputs, that is we will have to make sure that x_0 does *not* turn out to be a 0-height input for $NUMERATOR_{RAT}$ as well as for $DENOMINATOR_{RAT}$ so as not to be in the case:

$$\frac{large}{\boldsymbol{small}} = large$$

$$\frac{medium}{\boldsymbol{small}} = large$$

$$\frac{small}{small} = any\ size$$

We will thus refer to a 0-height input for $DENOMINATOR_{RAT}$ as only a **possible ∞-height input**

Altogether, then, we have:

EXAMPLE 15.1. Possible ∞-height Input The 0-height inputs of the *denominator* of a rational function, if any, are the only *possible ∞-height inputs* for the rational function.

4. However, this happens to be one of these very rare situations in which there *is* "an easier way": After we have located the 0-height inputs for $DENOMINATOR_{RAT}$, instead of first making sure that they are not also 0-height inputs for $NUMERATOR_{RAT}$, we will gamble and just get the local input-output rule near each one of the 0-height inputs for $DENOMINATOR_{RAT}$. Then,

- If the local input-ouput rule turns out to start with a *negative-exponent power function*, then we will have determined that x_0 *is* an ∞-height input for RAT and the payoff will be that we will now get the local graph near x_0 for free.
- If the local input-ouput rule turns out *not* to start with a *negative-exponent power function*, then we will have determined that x_0 is *not* a ∞-height input for RAT after all and our loss will be that we will probably have no further use for the local input-output rule.

Overall, then, we will use the following two steps:

Step i. *Locate* the 0-height inputs for the *denominator*, $DENOMINATOR_{RAT}(x)$, by solving the equation $$DENOMINATOR_{RAT}(x) = 0$$ **Step ii.** Compute the *local input-output rule* near each one of the 0-height inputs for the *denominator*, if any.

The advantage is that we need not even refer to the **Division Size TEXAMPLE**: once we have a possible ∞-height input, we just get the local input-output rule near that possible ∞-height input, "for the better or for the worse".

EXAMPLE 15.2. L et $COUGH$ be the function specified by the global input-output rule

$$x \xrightarrow{COUGH} COUGH(x) = \frac{x^4 - x^3 - 10x^2 + x - 15}{x^2 + 5x + 6}$$

locate the ∞-height input(s) of $COUGH$, if any.

Step i. The possible ∞-height input(s) of $COUGH$ are the 0-height input(s) of $DENOMINATOR_{COUGH}(x)$, that is the solution(s), if any, of the equation

$$x^2 + 5x + 6 = 0$$

In general, solving an equation may or may not possible but in this case, the equation is a *quadratic* one and we have learned how to do this in **Chapter 12**. One way or the other, we find that there are two solutions:

$$-3, -2$$

which are the *possible* ∞*-height inputs* of the rational function $COUGH$.

Step ii. We compute the local input-output rules near -3 and near -2:

- Near -3:

$$h \xrightarrow{COUGH\,_{\text{near } -3}} COUGH(-3+h) = \left.\frac{x^4 - x^3 - 10x^2 + x - 15}{x^2 + 5x + 6}\right|_{x \leftarrow -3+h}$$
$$= \frac{x^4 - x^3 - 10x^2 + x - 15|_{x \leftarrow -3+h}}{x^2 + 5x + 6|_{x \leftarrow -3+h}}$$
$$= \frac{(-3+h)^4 - (-3+h)^3 - 10(-3+h)^2 + (-3+h) - 15}{(-3+h)^2 + 5(-3+h) + 6}$$

We try to approximate to the constant terms:

$$= \frac{(-3)^4 + [...] - (-3)^3 + [...] - 10(-3)^2 + [...] - 3 + [...] - 15}{(-3)^2 + [...] + 5(-3) + [...] + 6}$$
$$= \frac{+81 + 27 - 90 - 3 - 15 + [...]}{+9 - 15 + 6 + [...]}$$
$$= \frac{0 + [...]}{0 + [...]}$$
$$= \frac{[...]}{[...]}$$
$$= \textit{any size}$$

So we must go back and try to approximate to the linear terms, ignoring the constant terms since we just saw that they add up to 0 both in the numerator and the denominator:

$$= \frac{4(-3)^3h + [...] - 3(-3)^2h + [...] - 10 \cdot 2(-3)h + [...] + h}{2 \cdot (-3)h + [...] + 5h}$$
$$= \frac{-108h + [...] - 27h + [...] + 60h + [...] + h}{-6h + [...] + 5h}$$
$$= \frac{-74h + [...]}{-h + [...]}$$
$$= +74 + [...]$$

so that -3 is *not* an ∞-heigth input

- Near -2:

$$h \xrightarrow{COUGH_{\text{near } -2}} COUGH(-2+h) = \left.\frac{x^4 - x^3 - 10x^2 + x - 15}{x^2 + 5x + 6}\right|_{x \leftarrow -2+h}$$

$$= \frac{x^4 - x^3 - 10x^2 + x - 15|_{x \leftarrow -2+h}}{x^2 + 5x + 6|_{x \leftarrow -2+h}}$$

$$= \frac{(-2+h)^4 - (-2+h)^3 - 10(-2+h)^2 + (-2+h) - 15}{(-2+h)^2 + 5(-2+h) + 6}$$

We try to approximate to the constant terms:

$$= \frac{(-2)^4 + [...] - (-2)^3 + [...] - 10(-2)^2 + [...] - 2 + [...] - 15}{(-2)^2 + [...] + 5(-2) + [...] + 6}$$

$$= \frac{+16 + 8 - 40 - 2 - 15 + [...]}{+4 - 10 + 6 + [...]}$$

$$= \frac{-33 + [...]}{0 + [...]}$$

$$= \frac{-33}{[...]}$$

$$= large$$

So -2 is an ∞-height input for $COUGH$ and we need only find exactly how small [...] is to get the local input-output rule near -2

$$= \frac{-33 + [...]}{2 \cdot (-2)h + [...] + 5h}$$

$$= \frac{-33 + [...]}{h + [...]}$$

$$= -33h^{-1} + [...]$$

3 Offscreen Graph

Once the Essential Question has been answered, and if we do not already have the local input-output rule near each one of the ∞-height inputs, we need to get them and the corresponding local graphs so that we can then join them smoothly to get the offscreen graph.

Altogether, given a rational function RAT the procedure to obtain the *offscreen graph* is therefore:

i. Get the approximate input-output rule near ∞ and the local graph near ∞

ii. Answer the **Essential Question** and locate the ∞ input(s), if any,

iii. Find the local input-output rule and then the local graphs near each ∞-height inputs

Example 15.3. **L** et $MARA$ be the function specified by the global input-output rule

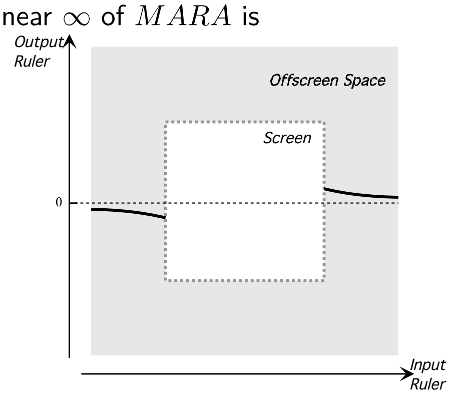
$$x \xrightarrow{MARA} MARA(x) = \frac{x - 15}{x^2 + 5x + 7}$$

Find the offscreen graph.

i. We get the local approximation near ∞:

$$\text{Near } \infty,\ x \xrightarrow{MARA} MARA(x) = \frac{x + [...]}{x^2 + [...]}$$
$$= +x^{-1} + [...]$$

and the local graph near ∞ of $MARA$ is



ii. We locate the ∞-height inputs, if any. The possible ∞-height input(s) of $MARA$ are the 0-height input(s) of $DENOMINATOR_{MARA}(x)$, that is the solution(s), if any, of the equation

$$x^2 + 5x + 7 = 0$$

In general, solving an equation may or may not possible but in this case, the equation is a *quadratic* one and we have learned how to do this in **Chapter 12**. One way or the other, we find that there are no solution. So, the function $MARA$ has no ∞-height input.

iii. The *offscreen graph* therefore consists of only the local graph near ∞.

4 Feature-sign Change Inputs

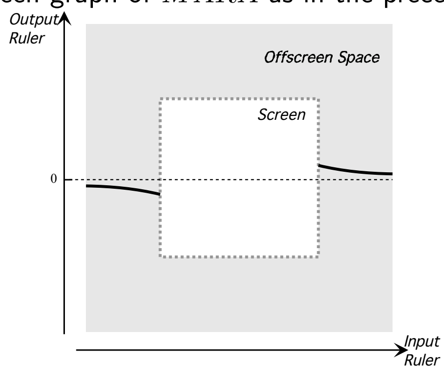
Given a rational function, in order to get the feature-sign change input(s), if any, we need only get the outlying graph and then we proceed as in **Chapter 3** so we need only give an example.

EXAMPLE 15.4. **L** et $MARA$ be the function specified by the global input-output rule

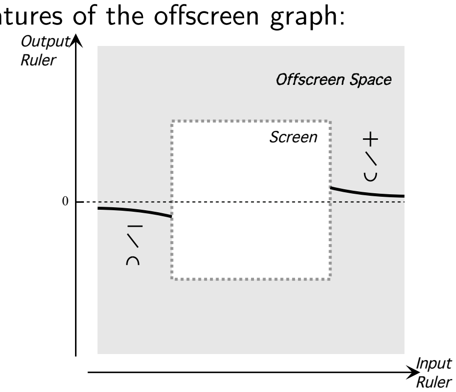
$$x \xrightarrow{MARA} MARA(x) = \frac{x-15}{x^2+5x+7}$$

Find the feature-sign change inputs of $MARA$, if any.

i. We find the offscreen graph of $MARA$ as in the preceding example:



ii. We mark the features of the offscreen graph:



iii. Therefore:

- there must be at least one height-sign change input,
- there does not have to be a slope-sign change input
- there must be at least one concavity-sign change input,

5 Global Graph

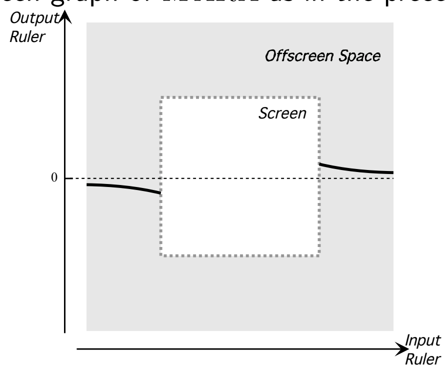
Given a rational function, in order to get the essential global graph, we need only get the outlying graph and then we join smoothly so we need only give an example.

EXAMPLE 15.5. **L** et $MARA$ be the function specified by the global input-output rule

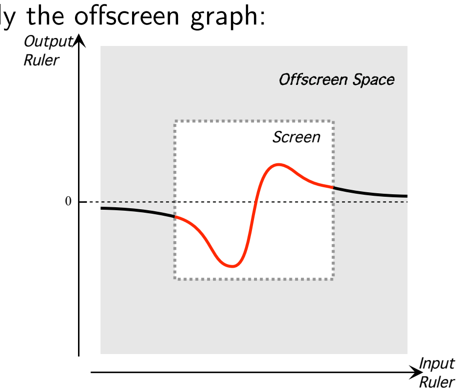
$$x \xrightarrow{MARA} MARA(x) = \frac{x-15}{x^2+5x+7}$$

Find the feature-sign change inputs of $MARA$, if any.

i. We find the offscreen graph of $MARA$ as in the preceding example:

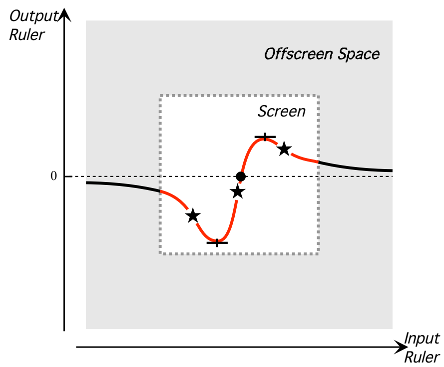


ii. We join smoothly the offscreen graph:



iii. Observe that, in fact,

- there must be at least one height-sign change input,
- there must be at least *two* slope-sign change inputs
- there must be at least *three* concavity-sign change input,



6 Locating 0-Height Inputs

Locating the 0-height inputs of a given rational function is pretty much the mirror image of what we did to locate its ∞-height inputs. More precisely:

1. Given a rational function *RAT* specified by a global input-output rule

$$x \xrightarrow{\ RAT\ } RAT(x) = \frac{NUMERATOR_{RAT}(x)}{DENOMINATOR_{RAT}(x)}$$

we want to find whether or not there can be a *bounded input* x_0 such that the outputs for *nearby* inputs, $x_0 + h$, are *small.* In other words, we want to know if there can be x_0 such that

$$h \xrightarrow{\ RAT\ } RAT(x)|_{x \leftarrow x_0+h} = small$$

But we have

$$\begin{aligned} RAT(x)|_{x \leftarrow x_0+h} &= \left.\frac{NUMERATOR_{RAT}(x)}{DENOMINATOR_{RAT}(x)}\right|_{x \leftarrow x_0+h} \\ &= \frac{NUMERATOR_{RAT}(x)|_{x \leftarrow x_0+h}}{DENOMINATOR_{RAT}(x)|_{x \leftarrow x_0+h}} \\ &= \frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)} \end{aligned}$$

So, what we want to know is if there can be an x_0 for which

$$\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)} = small$$

2. Since it is a *fraction* that we want to be *small*, we will use the **Division Size TEXAMPLE** from **Chapter 2**:

THEOREM 2 (**Division Size**)

$$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = large \qquad \frac{large}{small} = large$$
$$\frac{medium}{large} = \boldsymbol{small} \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = large$$
$$\frac{small}{large} = \boldsymbol{small} \qquad \frac{small}{medium} = \boldsymbol{small} \qquad \frac{small}{small} = any\ size$$

There are thus two ways that a fraction can be *small*:

- When the numerator is *small*
- When the denominator is *large*

In each case, though, we need to make sure of the other side of the fraction. So, rather than look at the size of both the numerator and the denominator at the same time, we will look separately at:

- The third *row*, that is when the *numerator* of the fraction is *small*

$$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = large \qquad \frac{large}{small} = large$$
$$\frac{medium}{large} = small \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = large$$
$$\frac{small}{large} = \boldsymbol{small} \qquad \frac{small}{medium} = \boldsymbol{small} \qquad \frac{small}{small} = any\ size$$

  because in that case all we will then have to do is to make sure that the *denominator* is *not small* too.

- The first *column*, that is when the *denominator* of the fraction is *large*.

$$\frac{large}{large} = any\ size \qquad \frac{large}{medium} = large \qquad \frac{large}{small} = large$$
$$\frac{medium}{large} = \boldsymbol{small} \qquad \frac{medium}{medium} = medium \qquad \frac{medium}{small} = large$$
$$\frac{small}{large} = \boldsymbol{small} \qquad \frac{small}{medium} = \boldsymbol{small} \qquad \frac{small}{small} = any\ size$$

  because in that case all we will then have to do is to make sure that the numerator is *not large* too.

3. We now deal with $\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)}$, looking separately at the numerator and the denominator:

- Since the *numerator*, $NUMERATOR_{RAT}(x_0 + h)$, is the output of a *polynomial function*, namely

$$x \xrightarrow{\quad NUMERATOR_{RAT} \quad} NUMERATOR_{RAT}(x)$$

and since we have seen that polynomial functions *can* have *small* outputs if they have 0-height inputs and the inputs are near the 0-height inputs, $NUMERATOR_{RAT}(x_0 + h)$ *can* be *small* for certain bounded inputs and thus so can $\frac{NUMERATOR_{RAT}(x_0+h)}{DENOMINATOR_{RAT}(x_0+h)}$. However, we will then have to make sure that $DENOMINATOR_{RAT}(x_0+h)$, is *not small* too near these bounded inputs, that is we will have to make sure that x_0 does *not* turn out to be a 0-height input for $DENOMINATOR_{RAT}$ as well as for $NUMERATOR_{RAT}$ so as not to be in the case: 0-height input

$$\frac{small}{small} = any\ size$$

We will thus refer to a 0-height input for $NUMERATOR_{RAT}$ as only a possible **0-height input** for RAT.

- Since the *denominator*, $DENOMINATOR_{RAT}(x_0 + h)$, is the output of a *polynomial function*, namely

$$x \xrightarrow{DENOMINATOR_{RAT}} DENOMINATOR_{RAT}(x)$$

and since we have seen that *the only way* the outputs of a *polynomial function* can be *large* is when the inputs are themselves *large*, *there is no way* that $DENOMINATOR_{RAT}(x_0 + h))$ could be *large* for inputs that are *bounded*. So there is no way that the output of RAT could be *small* for *bounded* inputs that make the *denominator* large and we need not look any further.

Altogether, then, we have:

EXAMPLE 15.6. Possible 0-height Input The 0-height inputs of the *numerator* of a rational function, if any, are the only *possible 0-height inputs* for the rational function.

4. However, this happens to be one of these very rare situations in which there *is* "an easier way": After we have located the 0-height inputs for $NUMERATOR_{RAT}$, instead of first making sure that they are not also 0-height inputs for $DENOMINATOR_{RAT}$, we will gamble and just get the local input-output rule near each one of the 0-height inputs for $NUMERATOR_{RAT}$. Then,

- If the local input-ouput rule turns out to start with a *positive-exponent power function*, then we will have determined that x_0 *is* a 0-height input for RAT and the payoff will be that we will now get the local graph near x_0 for free.
- If the local input-ouput rule turns out to start with a 0-*exponent power function* or a *negative-exponent power function*, then we will have deter-

mined that x_0 is *not* a 0-height input for RAT after all and our loss will be that we will probably have no further use for the local input-output rule.

Overall, then, we will use the following two steps:

Step i. *Locate* the 0-height inputs for the *numerator*, $NUMERATOR_{RAT}(x)$, by solving the equation

$$NUMERATOR_{RAT}(x) = 0$$

Step ii. Compute the *local input-output rule* near each one of the 0-height inputs for the *numerator*, if any.

The advantage is that we need not even refer to the **Division Size TEXAMPLE**: once we have a possible 0-height input, we just get the local input-output rule near that possible 0-height input, "for the better or for the worse".

EXAMPLE 15.7. L et $TARA$ be the function specified by the global input-output rule

$$x \xrightarrow{TARA} TARA(x) = \frac{x^3 - 8}{x^2 + 3x - 10}$$

locate the 0-height input(s) if any.

Step i. The possible 0-height input(s) of $TARA$ are the 0-height input(s) of $NUMERATOR_{TARA}(x)$, that is the solution(s), if any, of the equation

$$x^3 - 8 = 0$$

In general, solving an equation may or may not possible and in this case, the equation is a *cubic* one. Still, here it is a very incomplete one and we can see that the solution is $+2$ which is the possible 0-height input of the rational function $TARA$.

Step ii. We compute the local input-output rule near $+2$.

$$\begin{aligned} h \xrightarrow{TARA_{\text{ near } -3}} TARA(+2+h) &= \left.\frac{x^3 - 8}{x^2 + 3x - 10}\right|_{x \leftarrow +2+h} \\ &= \frac{x^3 - 8|_{x \leftarrow +2+h}}{x^2 + 3x - 10|_{x \leftarrow +2+h}} \\ &= \frac{(+2+h)^3 - 8}{(+2+h)^2 + 3(+2+h) - 10} \end{aligned}$$

We try to approximate to the constant terms:

$$= \frac{(+2)^3 + [...] - 8}{(+2)^2 + [...] + 3(+2) + [...] - 10}$$

$$= \frac{+8 - 8 + [...]}{+4 + 6 - 10 + [...]}$$
$$= \frac{0 + [...]}{0 + [...]} = \frac{[...]}{[...]} = \textit{any size}$$

So we must go back and approximate to the linear terms, ignoring the constant terms since we just saw that they add up to 0 both in the numerator and the denominator:

$$= \frac{3(+2)^2 h + [...]}{2(+2)h + [...] + 3h}$$
$$= \frac{+12h + [...]}{+4h + [...] + 3h} = \frac{+12h + [...]}{+7h + [...]}$$
$$= +\frac{12}{7} + [...]$$

and, since $+\frac{12}{7} \neq 0$, $+2$ is *not* an 0-heigth input for $TARA$.

Chapter 16

Homographic Functions

Part V

Transcendental Functions

pppppp

Part VI

Epilogue

- Derived functions
- Functions defined equationally
- Matters of *size* e.g. the bigger the size of the exponent, the boxier the graph

Check that reciprocity has been moved correctly to **Chapter 7**

1 Looking Back

Until now, the global graph of each new kind of function was qualitatively very different as we moved from one kind of functions to the next.

1. In the case of the *power functions*, we found that the *qualitative features* of the global graphs of

i. regular positive-exponent power functions,
ii. negative-exponent power functions,
iii. exceptional power functions, that is
  - 0-exponent power functions
  - 1-exponent power functions

were very different but the differences among power functions of any particular type were not really that great in that, from the point of view of the *shape* of the global graph, there were really only four types of regular power functions (depending on the *sign* and the *parity* of the *exponent*) and only two types of exceptional power functions (depending on the *parity* of the *exponent*).

2. In the case of the *polynomial functions*, we found that the *qualitative features* of the global graphs changed a lot when we moved from one degree to the next:

i. The global graph of a *constant function* (Degree 0)
  - has no *height*-sign change input, (same *height* everywhere)
  - has no *slope*,
  - has no *concavity*,

ii. The global graph of an *affine function* (Degree 1)
  - always has exactly one *height*-sign change input,
  - has no *slope*-sign change input, (same *slope* everywhere)

- has no *concavity*,

iii. The global graph of a *quadratic function* (Degree 2)
- may or may not have *height*-sign change input(s),
- always has exactly one *slope*-change input,
- has no *concavity*-sign change input, (same *concavity* everywhere)

iv. The global graph of a *cubic function* (Degree 3)
- has at least one *height*-sign change inputs,
- may or may not have *slope*-change input(s),
- has exactly one *concavity*-sign change input,

As for the qualitative differences among the global graphs of polynomial functions of a *same* degree, they are not great—but growing along with the *degree*.

i. The difference among *constant functions* is the *height* of the global graph.

ii. The differences among *affine functions* are the *height* and the slope of the global graph.

iii. The differences among *quadratic functions* are the *height*, the slope and the concavity of the global graph.

iv. The differences among *cubic functions* are not only the *height*, the slope and the *concavity* of the global graph but also whether or not there is a *bounded fluctuation*.

Thus, in terms of content organization, the *degree* of polynomial functions was a very powerful organizer if only because this allowed us introduce the features, *height*, *slope*, *concavity*, one at a time.

The emphasis throughout will be to convince ourselves of the need to proceed very systematically while keeping our eyes open so as to take advantage of whatever might make our life easier and not to do anything that we do not absolutely have to do.

2 Looking Ahead

We will now say a few words about the way rational functions will be dealt with in the rest of this text.

1. While, so far, we have had a very transparent content organization, in contrast, in the case of *rational functions*, the *rational degree* will *not* be such a powerful organizer because the four different types of rational functions will not be markedly different.

Still, in each one of the next four chapters, we will investigate a given type of rational function but this will be mostly in order not to upset the reader with too much variety from the get go. However, we will not be able to develop much of a theory for each type and we will mostly go about gathering experience investigating rational functions without paying too much attention to the type of rational function being dealt with, taking things as they come.

On the other hand, the differences among rational functions of any given type of rational degree, will be quite significant because of the possible ∞-height inputs.

Thus, the other side of the coin will be that, while, until now, once we had a theory of a kind of function, the investigation of this kind of functions quickly became a bit boring in that we knew what the overall global graph was going to look like, in the case of rational functions, there will be a much more interesting *diversity*.

2. Before anything else, it should be stressed that in the investigations of any given *rational function* we will follow essentially the exact same approaches that we used in the investigation of any given *power function* and of any given *polynomial function*: We will thus

i. get its local graph near ∞,
ii. get the answer to the ESSENTIAL QUESTION and find the ∞-*height input(s)*, if any. (This will involve solving an equation.)
iii. get the local graph near the ∞-*height inputs*, if any.
iv. get the global graph by interpolating the local graph near ∞ and the local graphs near the ∞-*height inputs*, if any.

3. As happened each time we investigated a new kind of function, finding the local rule near bounded inputs—and therefore near ∞-*height input(s)*—will require a new algebra tool.

4. As with any function, rational or otherwise, what we will actually do will depend of course on what information we need to find and there are going to be two main kinds of questions:

a. *Local questions*, that is, for instance:

- Find the local concavity-sign near a given input,
- Find the local slope-sign near a given input,
- Find the local height-sign near a given input,
- Find the local graph near a given input,

The given input can of course be *any* input, that is ∞ or any given *bounded* input, for instance an ∞-height input, a concavity-sign change input, a slope-sign change input, a height-sign change input or any *ordinary* input whatsoever.

b. *Global questions*, that is, for instance

reciprocal function
reciprocal

- Find the concavity-sign change input(s), if any
- Find the slope-sign change input(s), if any
- Find the height-sign change input(s), if any
- Find where the output has a given concavity-sign
- Find where the output has a given slope-sign
- Find where the output has a given height-sign
- Find the global graph

In the case of global questions, it will usually be better to start by getting the *bounded graph* and then to get the required information from the bounded graph. But then of course, since the bounded graph is really only the *essential* bounded graph, that is the graph that is interpolated from the *outlying graph*, the global information that we will get will only be about the *essential* features that is the features forced onto the bounded graph by the *outlying graph*.

The curious reader will obviously have at least three questions:

i. How do the various power functions compare among each other?

ii. What of polynomial functions of degree higher than 3?

iii. What of Laurent polynomial functions?

In the "overview", we will discuss the several manners in which *regular positive-power* functions, *negative-power* functions and *exceptional-power* functions all fit together. This will require discussing the *size* of slope.

3 Reciprocity Between 0 and ∞

We will now investigate the relationship between 0 and ∞

1. Reciprocal Function The **reciprocal function** is the power function with exponent -1 and coefficient $+1$, that is the function whose global input-output rule is

$$x \xrightarrow{\ RECIPROCAL\ } RECIPROCAL(x) = (+1)x^{-1} = +\frac{1}{x}$$

so that the output is the **reciprocal** of the input (hence the name).

1. The first thing about the reciprocal function is that it is typical of negative-exponent power functions in terms of what it does to the *size* of the output:

$$+large \xrightarrow{\ RECIPROCAL\ } RECIPROCAL(large) = +small$$

$$-large \xrightarrow{RECIPROCAL} RECIPROCAL(large) = -small$$

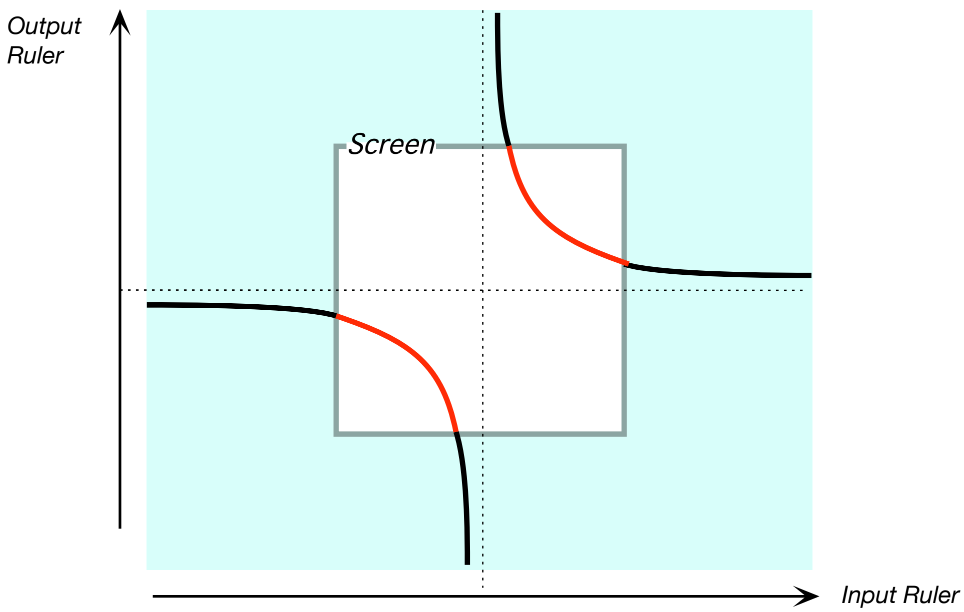
and

$$+small \xrightarrow{RECIPROCAL} RECIPROCAL(small) = +large$$

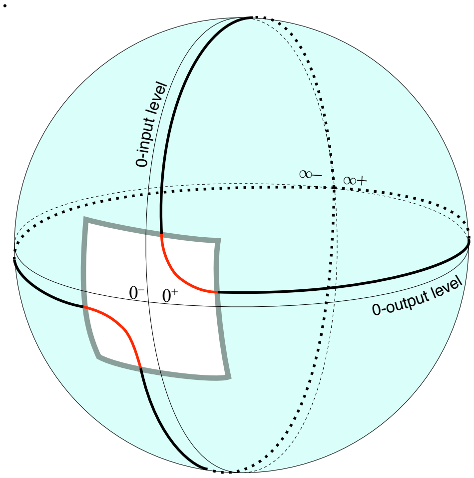
$$-small \xrightarrow{RECIPROCAL} RECIPROCAL(small) = -large$$

2. More generally, the global graph of the reciprocal function is:

- Mercator picture:



- Magellan picture:



3. Although quite different from the *identity function*, the *reciprocal*

family
prototypical

functions does play a role in the **family** of all power functions that is quite similar in some respects to the role played by the *identity function*

For instance, because the size of the exponent in both cases is 1, they are both the "first" of their kind.

However, that is not very important because:

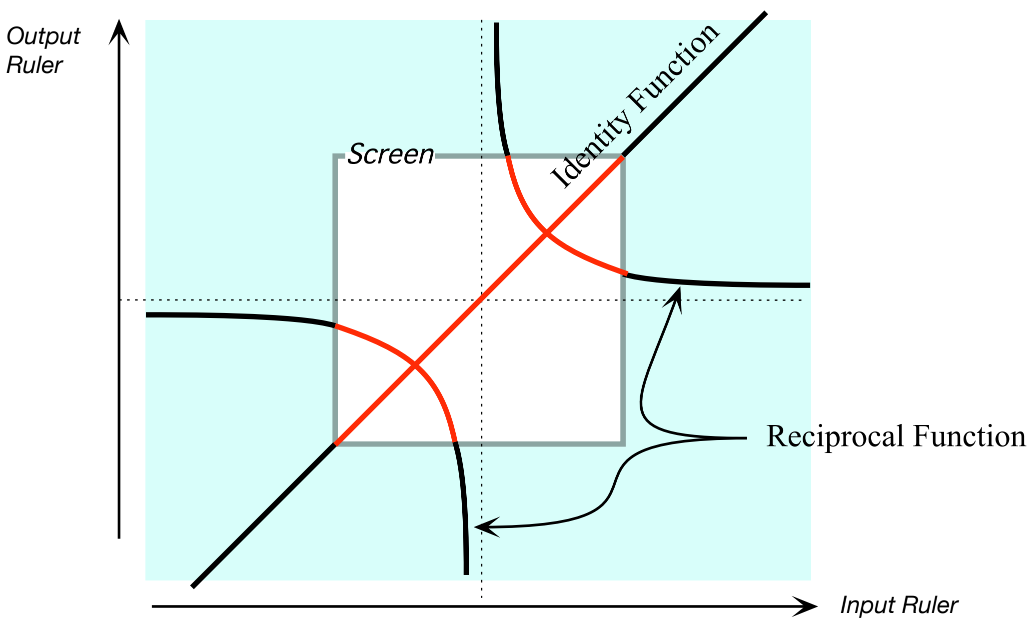
- The *identity function* is *not* **prototypical** of the other power functions because the identity function is a linear function and has no concavity.

  EXAMPLE 16.1. The identity function lack concavity while all regular power function have concavity.

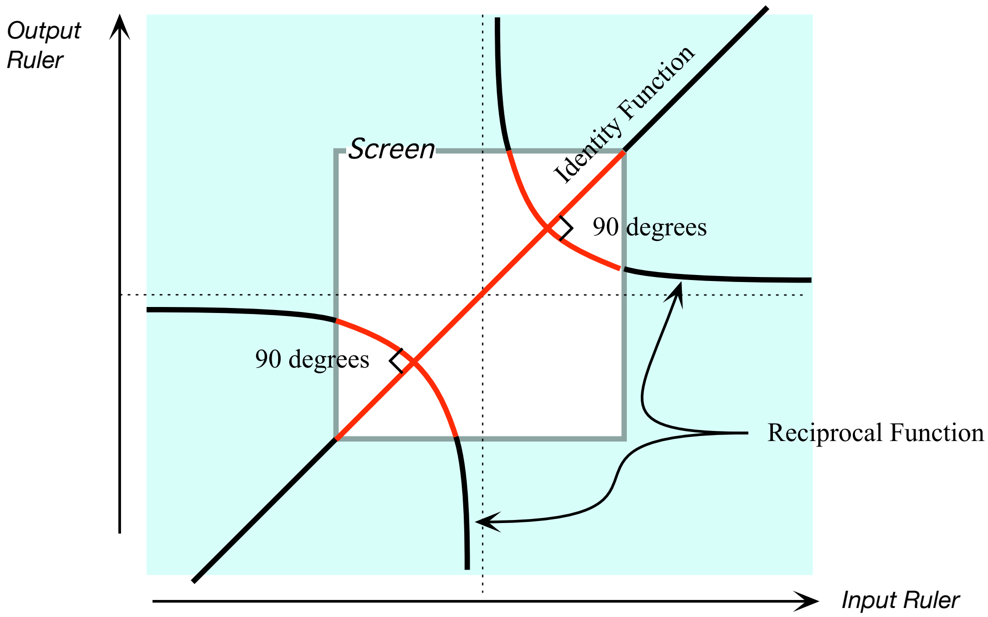
- The *reciprocal function* is *prototypical* of the other *negative power* functions in many ways.

  EXAMPLE 16.2. The shape of the reciprocal function is essentially the same as the shape of all (negative-exponent) power functions of type NOP

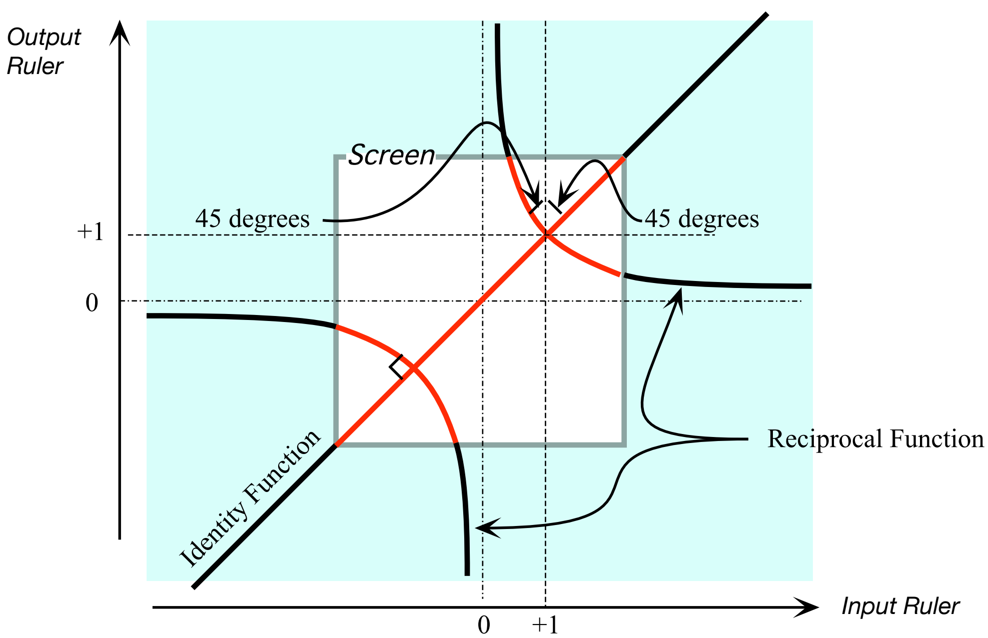
One thing the identity function and the reciprocal function have in common, though and for what it's worth at this time, is that the reciprocal function is the mirror image of itself when the mirror is the identity function.



In particular, they intersect at a 90 degree angle.



Another way way to look at it is that the local graphs near +1 are locally mirror images of each other when the mirror is the input level line for +1:



2. Reciprocity

far
reciprocal of each other

1. It will be convenient to introduce two new terms:

- We introduced the word "*near*" almost from the begining and, with Magellan graphs in mind, we will now introduce the word "**far**". Thus,
  - When an input is *large*, it is *near* ∞ and therefore *far* from 0,
  - When an input is *small*, it is *near* 0 and therefore *far* from ∞.
- More generally, we will say that *two* power functions are **reciprocal of each other** when:
  - their coefficient are *the same*,
  - the *size* of their exponents are *the same*,
  - the *sign* of their exponents are *the opposite*.

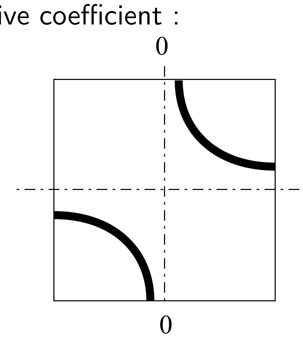
  In other words, two power functions are *reciprocal of each other* whenever they differ only by the *sign* of their *exponents*.

EXAMPLE 16.3. The identity function and the reciprocal function are reciprocal.

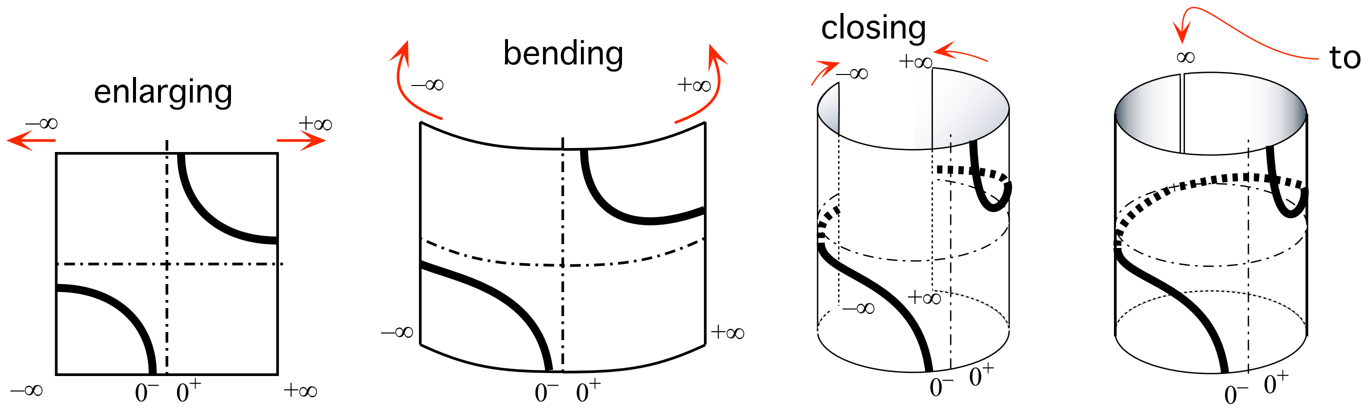
We will see that, when the mirror is the input level line for +1, the local graphs near +1 of two power functions that are *reciprocal of each other* are approximately mirror images of each other. But the angles will not be 45 degrees anymore.

2. The point of all this is that the local graph near ∞ of a regular power function is the same as the local graph near 0 of the power function that it is reciprocal of and, vice versa, the local graph near 0 of a regular power function is the same as the local graph near ∞ of the power function that it is reciprocal of.

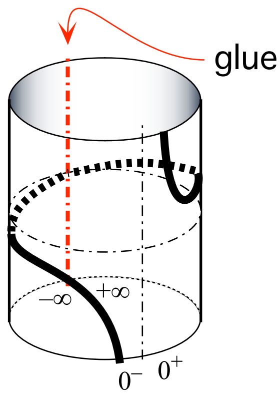
EXAMPLE 16.4. Given the local graph near 0 of $JACK$, an odd *positive* power function with positive coefficient :



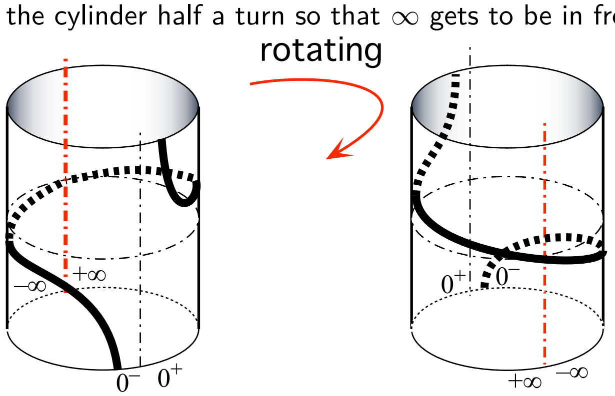
We enlarge the *extent* of the input ruler more and more while shrinking the *scale* by the edges more and more and, as we do so, we bend the screen backward more and more closing down the gap until the edges touch.



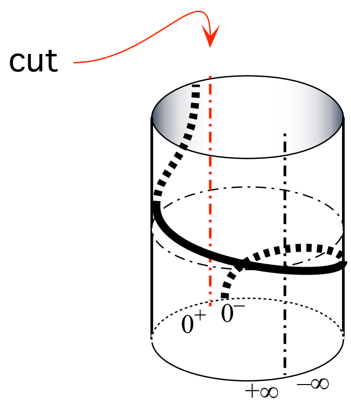
We then glue shut the edges of the screen at ∞ to get a cylinder.



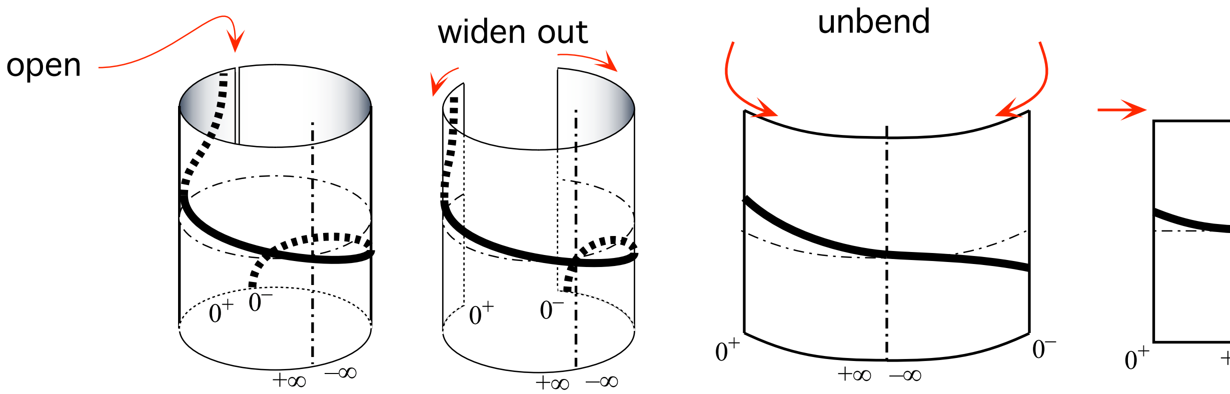
Then we turn the cylinder half a turn so that ∞ gets to be in front of us:



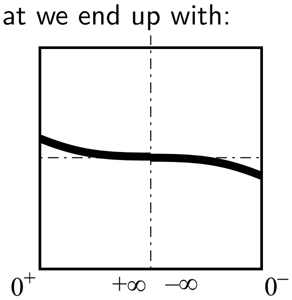
Now we cut open the cylinder along the input level line for 0



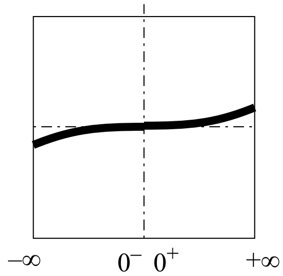
We widen out and unbend the screen forward more and more until it becomes flat.



The local graph near ∞ that we end up with:

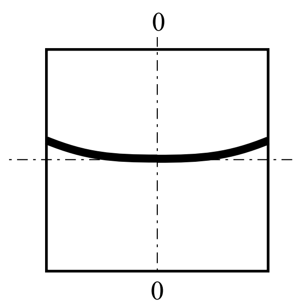


is exactly like the local graph near 0 of $JACK$'s *reciprocal* power function which is an odd *positive* power function with positive coefficient:

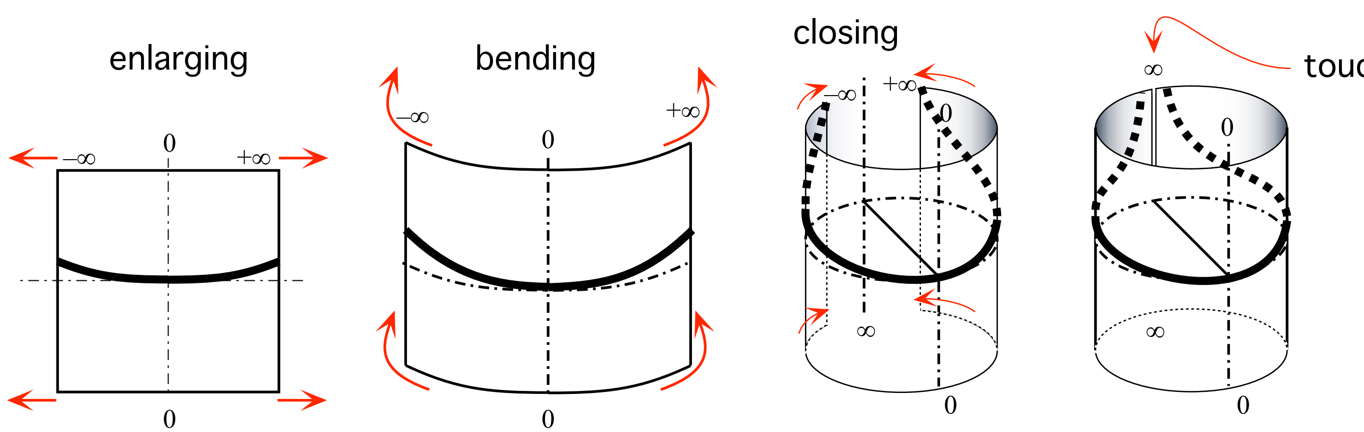


(On both graphs, outputs for negative inputs are negative and outputs for positive inputs are positive.)

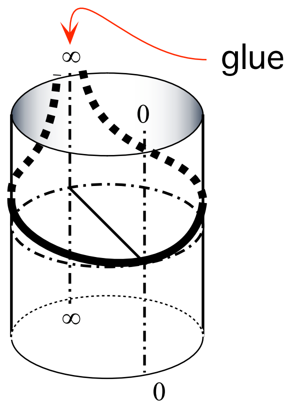
EXAMPLE 16.5. Given the local graph near 0 of the even *positive* power function $JILL$:



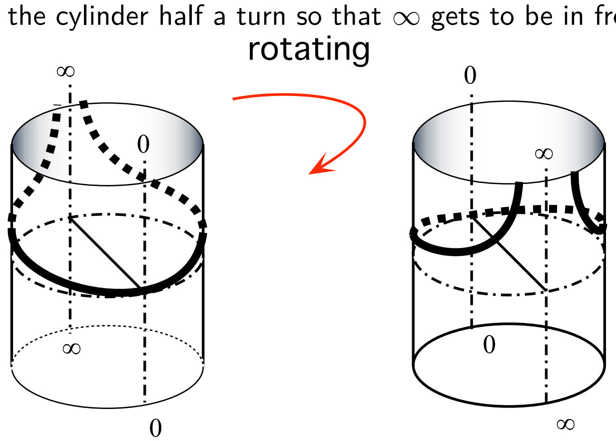
We enlarge the *extent* of the input ruler more and more while shrinking the *scale* by the edges more and more and, as we do so, we bend the screen backward more and more until the edges touch.



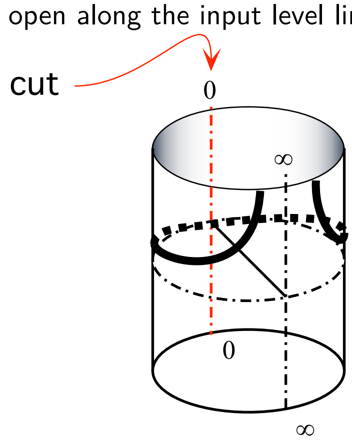
We then glue the edges of the screen at ∞ to get a cylinder.



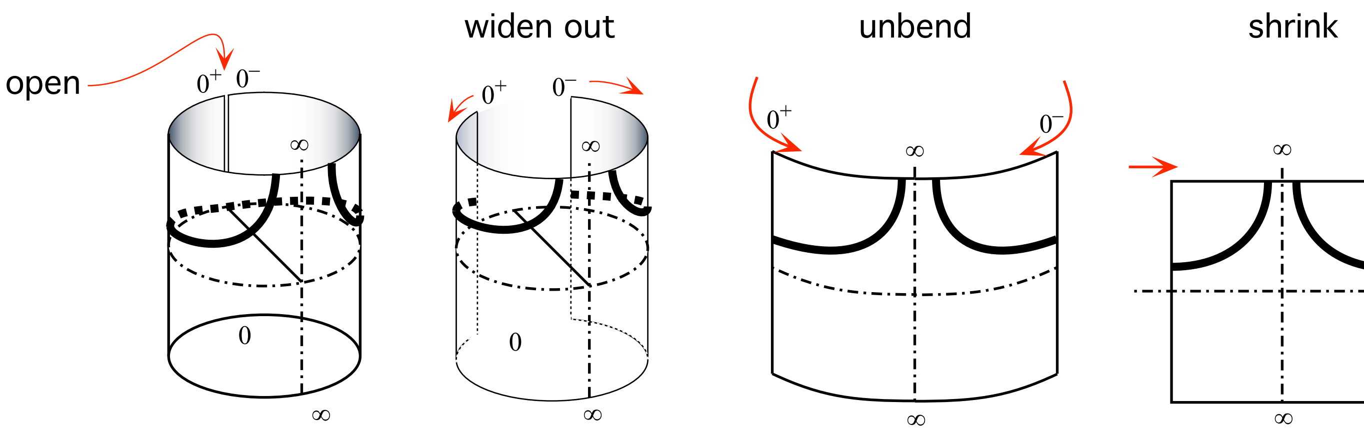
Then we turn the cylinder half a turn so that ∞ gets to be in front of us:



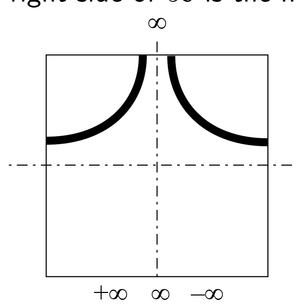
Now we cut the cylinder open along the input level line for 0



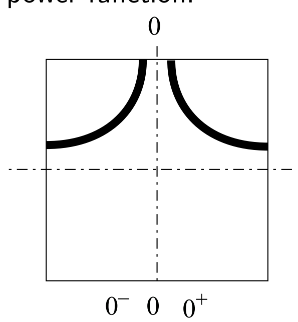
We unbend the screen forward more and more until it becomes flat.



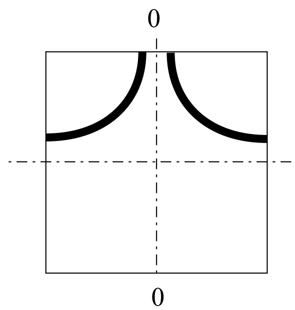
The local graph near ∞ that we get (Remember that the left side of ∞ is the positive side of ∞ and the right side of ∞ is the negative side of ∞):



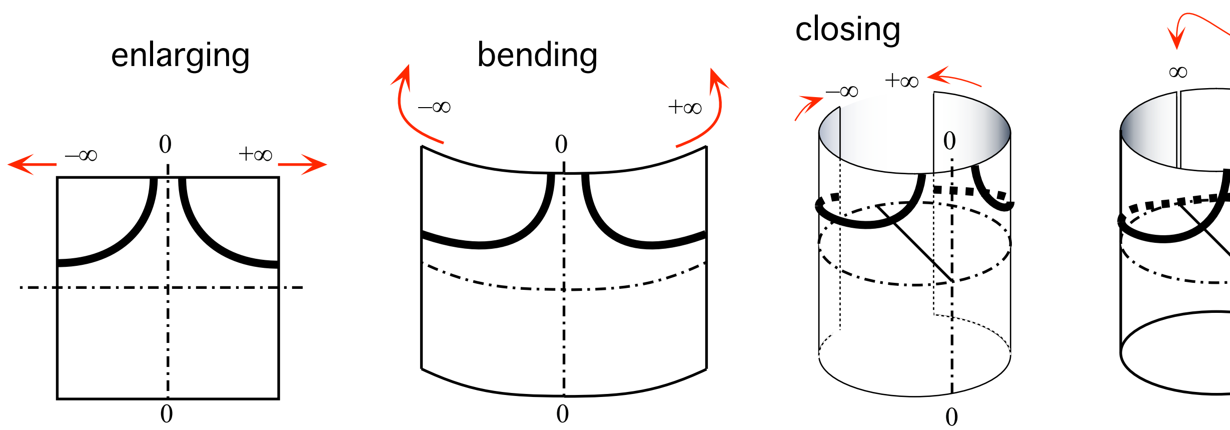
is just like the local graph near 0 of $JILL$'s *reciprocal* power function which is a *negative*, *even-exponent* power function:



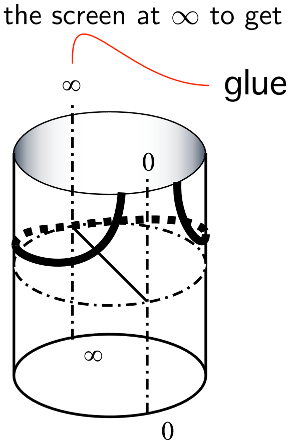
Example 16.6. Given the local graph near 0 of the even *positive* power function $JACK$:



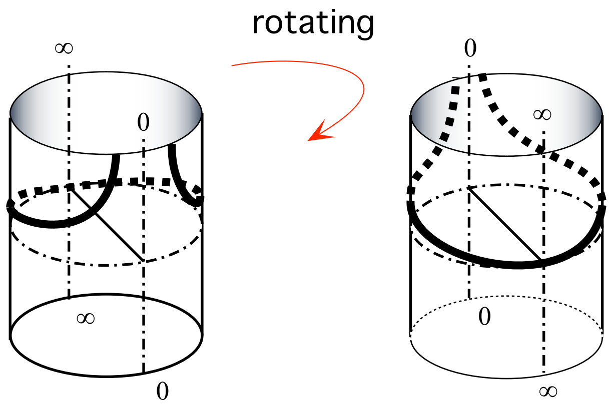
We enlarge the *extent* of the input ruler more and more while shrinking the *scale* by the edges more and more and, as we do so, we bend the screen backward more and more until the edges touch.



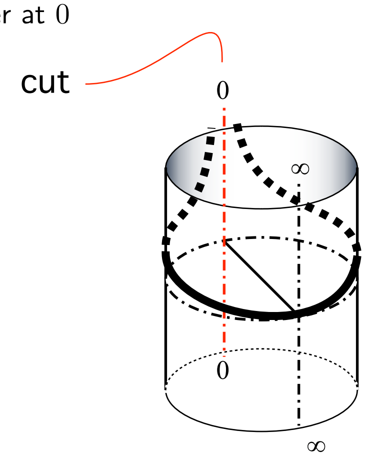
We then glue the edges of the screen at ∞ to get a cylinder.



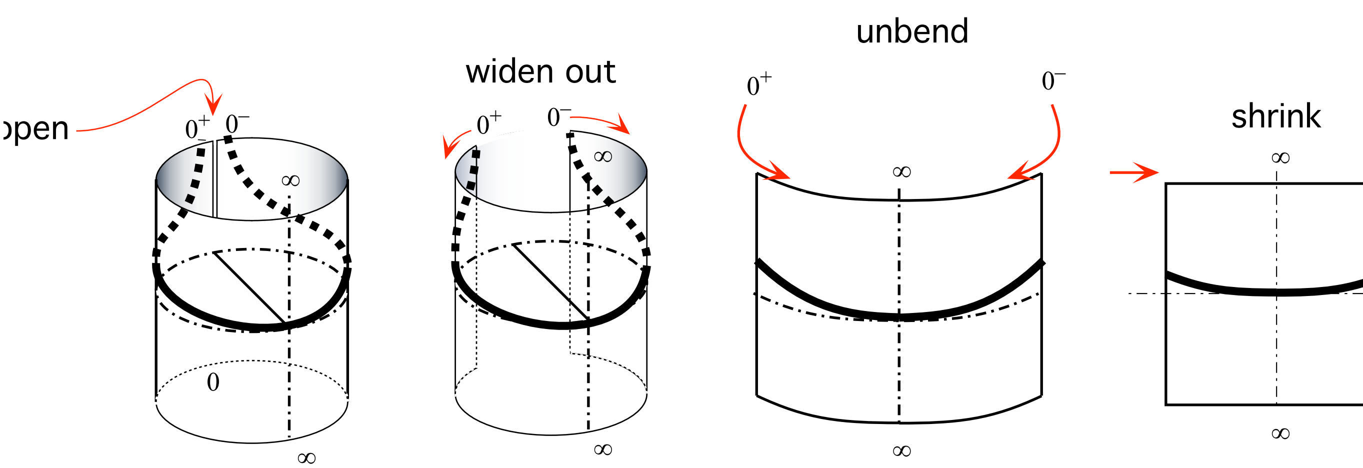
Then we turn the cylinder half a turn so that ∞ gets to be in front of us:



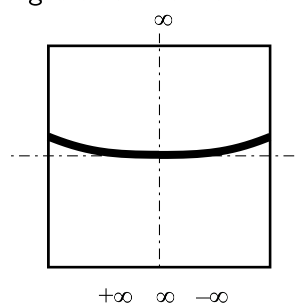
Now we cut the cylinder at 0



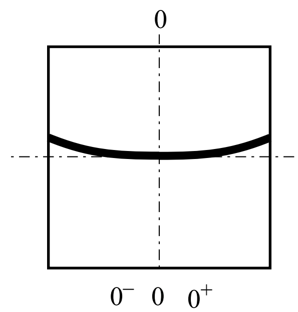
and we unbend the screen forward more and more until it becomes flat.



The local graph near ∞ that we get (Remember that the left side of ∞ is the positive side of ∞ and the right side of ∞ is the negative side of ∞):



is just like the local graph near 0 of $JACK$'s *reciprocal* power function which is an even *positive* power function:



4 The Family of Power Functions

The following is more of an informative nature at this stage than something that we will be building on in this text. The purpose here is mostly to give some coherence to all the power functions by showing various ways in which they fit together. It should help the reader organize her/his vision of power functions.

1. Types of Regular Functions This is just a recapitulation of stuff we saw in the preceding two chapters:

size-preserving
size-inverting
fixed point

Sign exponent	Parity exponent	Sign coefficient	TYPE
+	*Even*	+	PEP
		−	PEN
	Odd	+	POP
		−	PON
−	*Even*	+	NEP
		−	NEN
	Odd	+	NOP
		−	NON

2. What Power Functions Do To Size We will say that a function is **size-preserving** when the size of the output is the same as the size of the input, that is "small gives small" and "large gives large".

EXAMPLE 16.7. Regular positive-exponent power functions are *size-preserving*:

Correspondingly, we will say that a function is **size-inverting** when the size of the output is the *reciprocal* of the size of the input, that is "small gives large" and "large gives small".

EXAMPLE 16.8. Negative-exponent power functions are *size-inverting*:

By contrast, with *exponent-zero* power functions, the output for *small* inputs has size 1 and so is neither *small* nor *large* and so *exponent-zero* power functions are neither *size-preserving* nor *size-inverting*. You might say that they are "size-squashing".

Thus, in a way, constant functions separate *regular positive-exponent* power functions from *negative-exponent* power functions.

On the other hand, even though linear functions are exceptional, they are nevertheless *size-preserving*.

3. Fixed point A **fixed point** for a function is an input whose output is equal to the input.

template

EXAMPLE 16.9. Given the identity function, every input is a *fixed point*. In particular, both 0 and $+1$ are *fixed points*.

EXAMPLE 16.10. 0 is a fixed point for all regular power functions.

EXAMPLE 16.11. $+1$ is a fixed point for all regular power functions.

EXAMPLE 16.12. -1 is a fixed point for all regular even-exponent power functions.

5 The bigger the size of the exponent the boxier the graph

We will call **template** something that looks like it could be the graph of a regular power function except that it is not a function because the inputs -1 and $+1$ both have an unbounded number of outputs. Each type of regular power function has its own template.

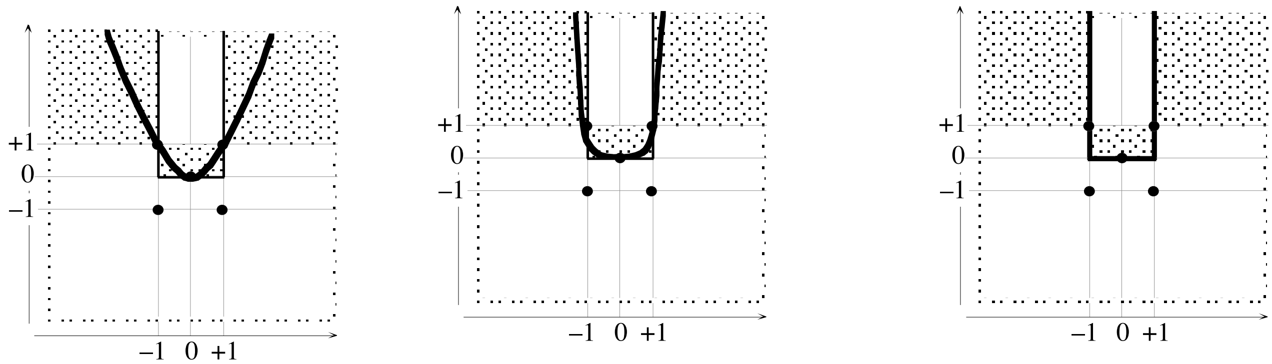
1. We begin by comparing power functions with their template two at a time.

EXAMPLE 16.13. The positive-even-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{+4}} POWER_{+4}(x) = +x^{+4}$$

is much closer to its template than the positive-even-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{+2}} POWER_{+2}(x) = +x^{+2}$$

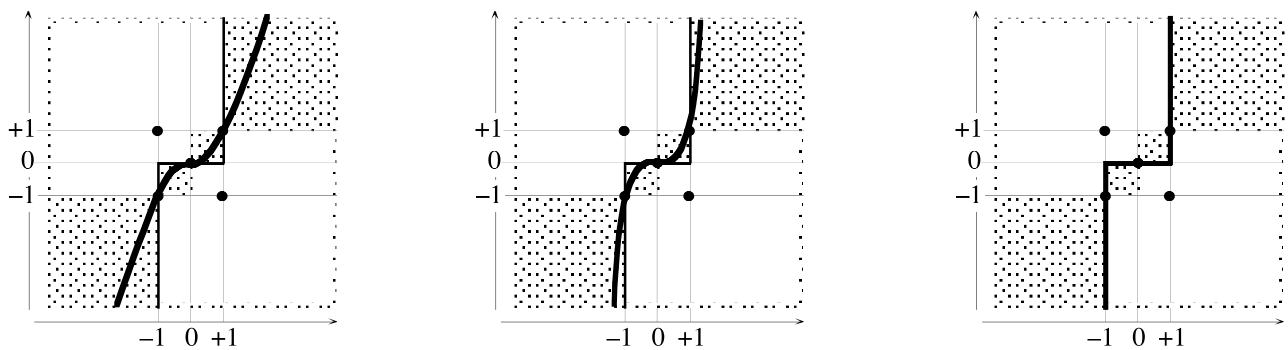


EXAMPLE 16.14. The positive-odd-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{+5}} POWER_{+5}(x) = +x^{+5}$$

is much closer to its template than the positive-odd-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{+3}} POWER_{+3}(x) = +x^{+3}$$

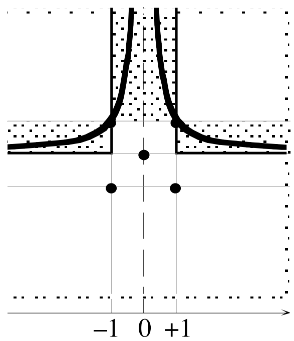


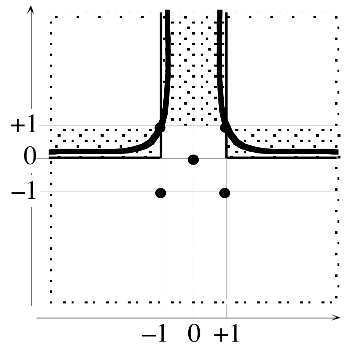
EXAMPLE 16.15. The negative-even-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{-4}} POWER_{+5}(x) = +x^{-4}$$

is much closer to its template than the negative-even-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{-2}} POWER_{-2}(x) = +x^{-2}$$



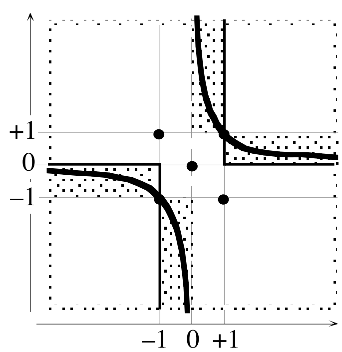


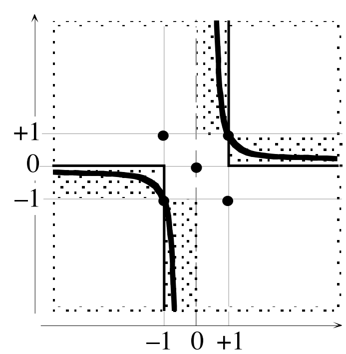
Example 16.16. The negative-odd-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{-3}} POWER_{-3}(x) = +x^{-3}$$

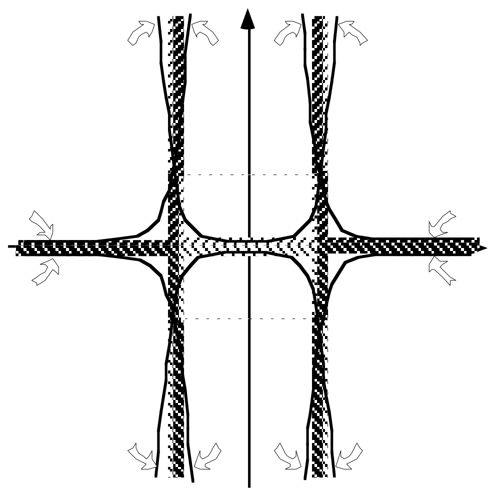
is much closer to its template than the negative-odd-exponent power function whose global input-output rule is

$$x \xrightarrow{POWER_{-1}} POWER_{-1}(x) = +x^{-1}$$



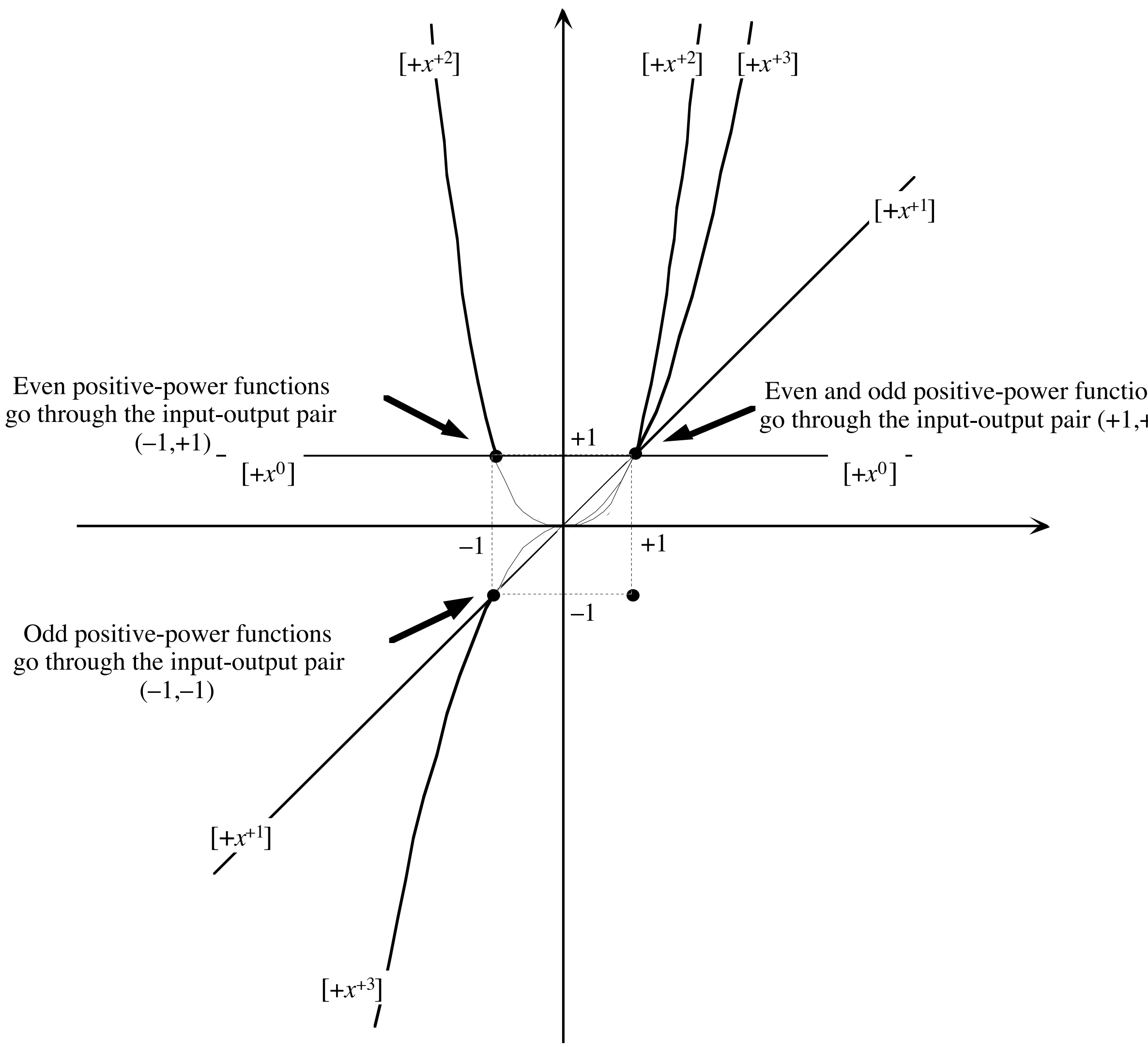


2. Together, power functions make an interesting pattern:

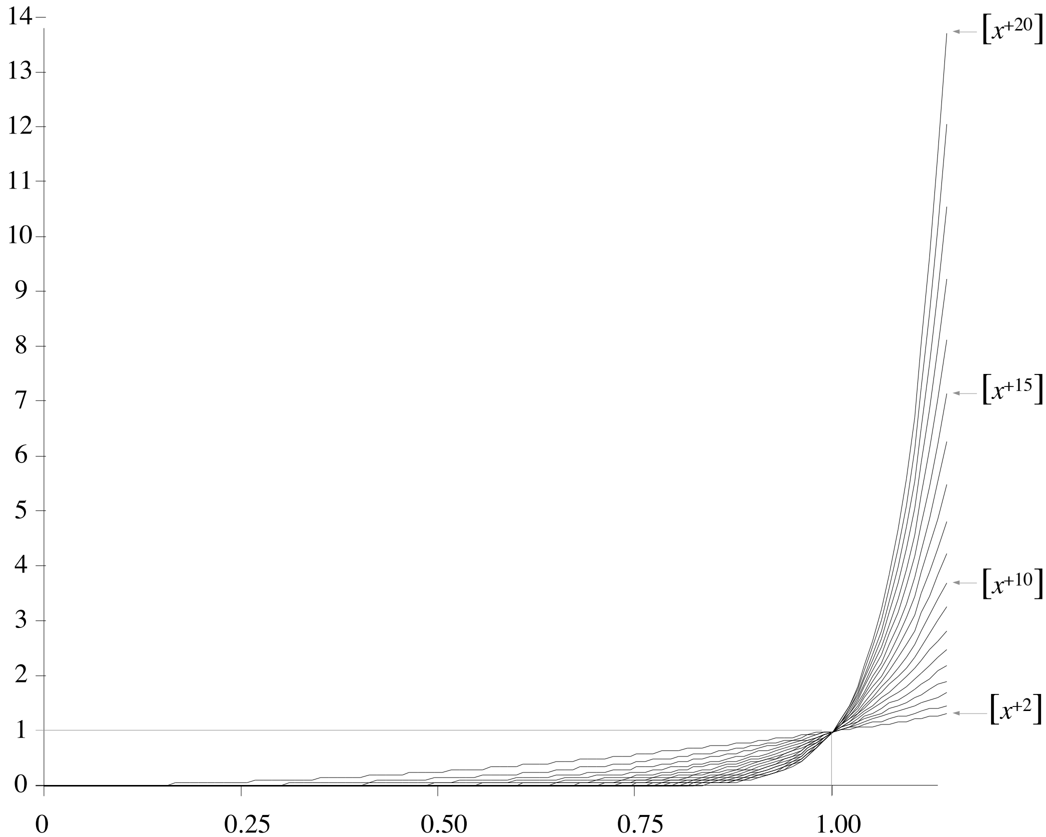


6 Local Quantitative Comparisons

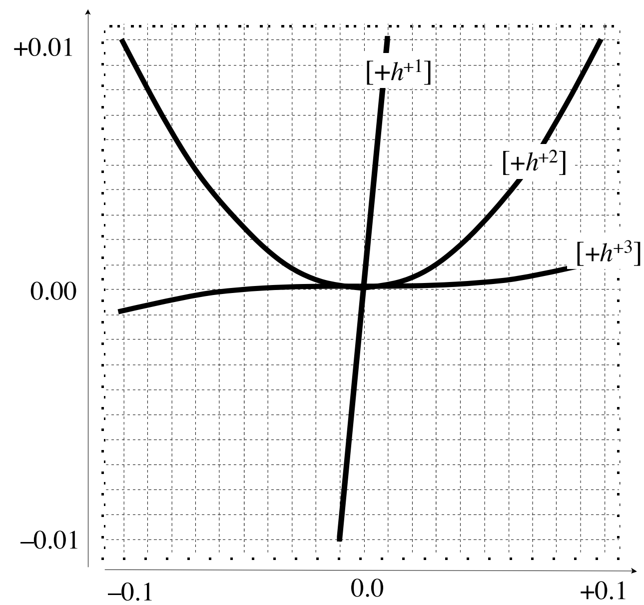
1. Local quantitative comparison near ∞



2. Local quantitative comparison near +1



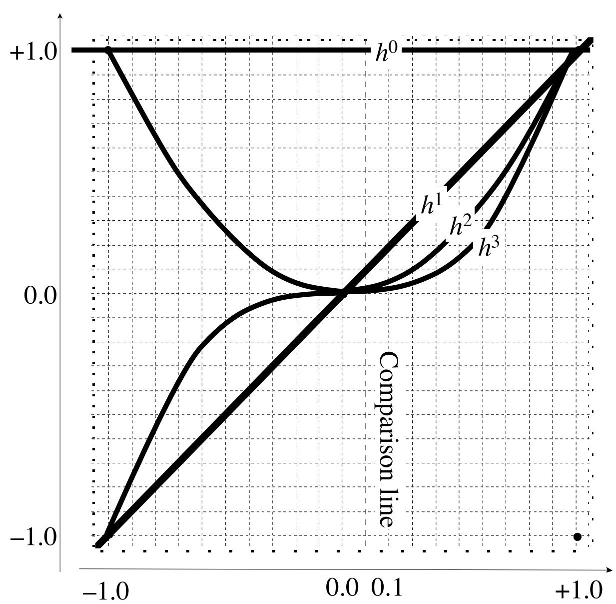
Local quantitative comparison near 0, between −0.1 and +0.1



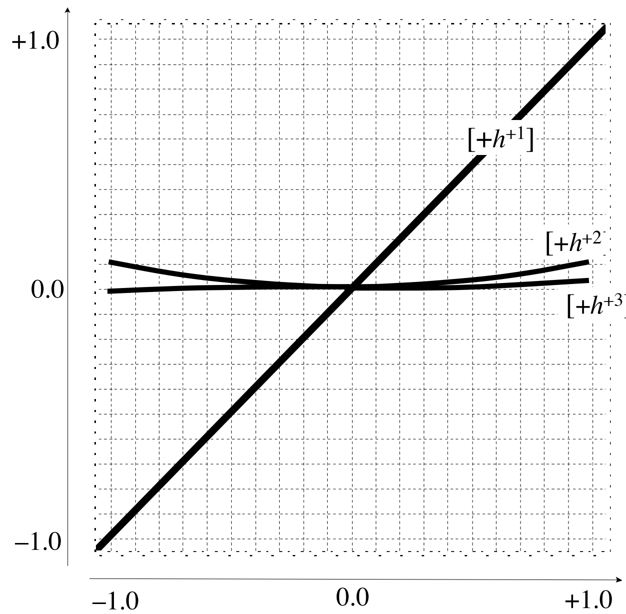
3.

7 Global Quantitative Comparisons

1. Global quantitative comparison between −1 and +1

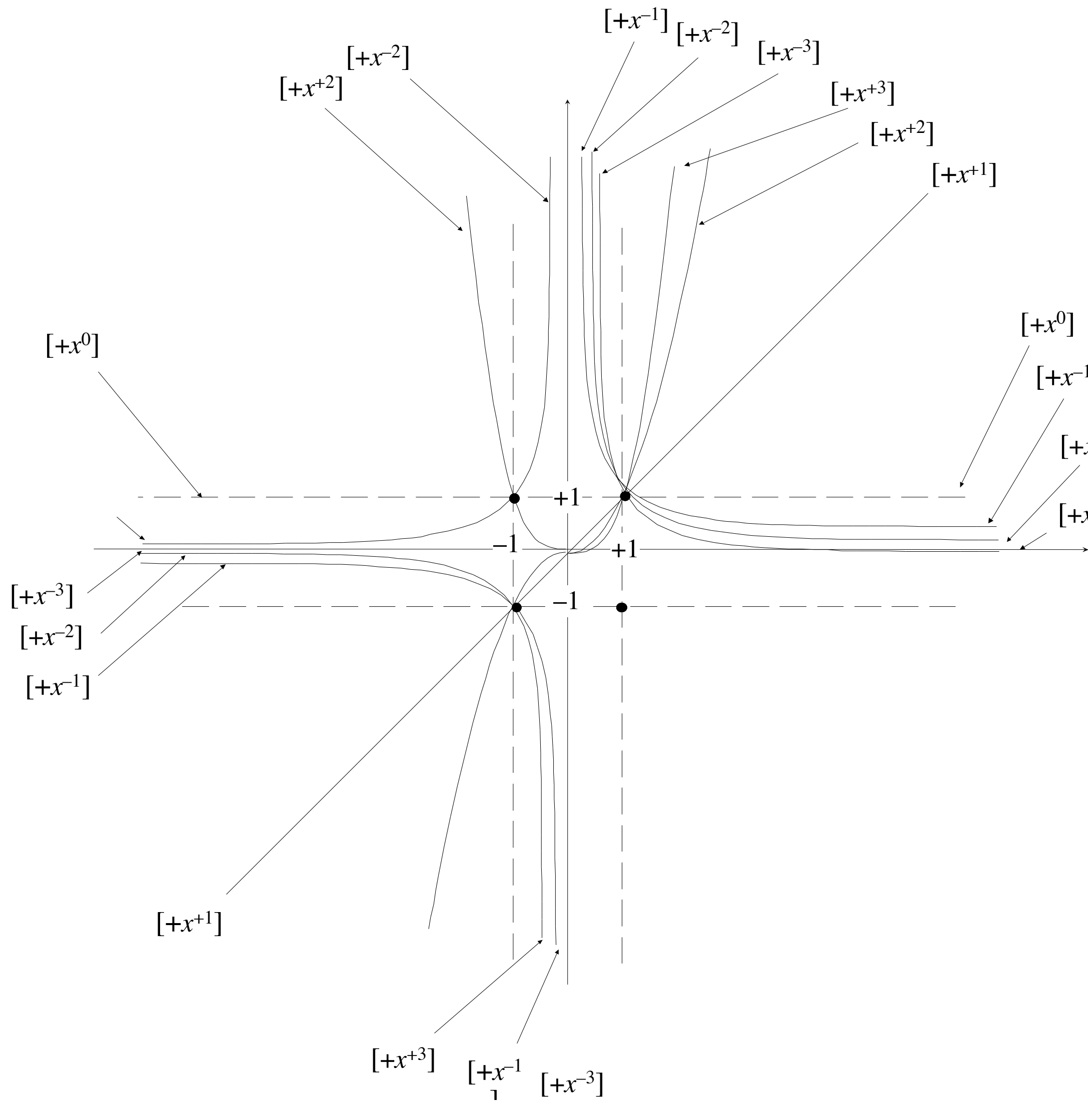


2. Global quantitative comparison between −1 and +1

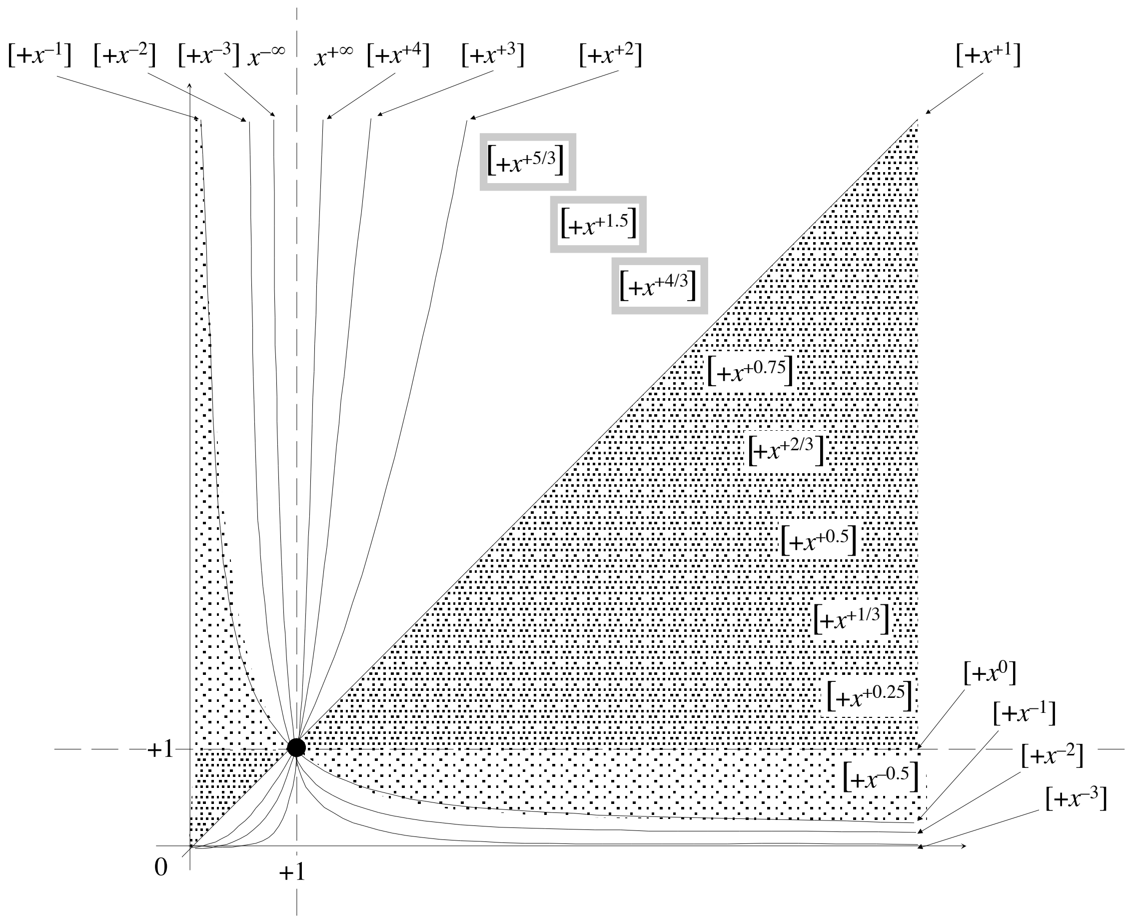


1. Symmetries Of Power Functions

3.



2. Coverage By Power Functions



Observe that there are graphs of power functions whose exponent is a fraction or a decimal number and that these graphs are exactly where we would expect them to be based on the way the fractional or decimal exponent fits with the whole number exponents. This, though, is a something that will be investigated in the next volume: REASONABLE TRANSCENDENTAL FUNCTIONS.

Part VII

Appendices

list
paper world
entity
numeral phrase
numerator
numeral
magnitude
quantitative information
denominator
essence
qualtative information

Appendix A

Dealing With Numbers

=======Begin LOOK UP=======

Collection of objects can be **listed**.

1 Real World *Numbers* - Paper World *Numerals*

With heavy reminders of to which world each word belongs!

Separating what is happening in the real world from what is happening in the **paper world** of a text is not easy so this section will use the terminology used in MODEL THEORY and LINGUISTICS. And since it is impossible to exhibit in the paper world the real world **entities** we will want to calculate about, we will use paper world *drawings* as *stand-ins* for real world entities:

There are two kinds of real world entities which we will both denote with paper world **numeral phrases** consisting of:

- A **numerator** using **numerals** (https://en.wikipedia.org/wiki/Numeral_(linguistics)) to provide the **magnitude** of the entity. (**Quantitative information**.)

and

- A **denominator** using words to provide the **essence** of the entity. (**Qualtative information**.)

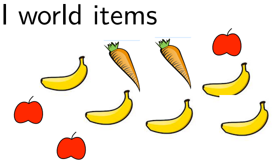
However, the two kinds of real world entities are different enough that we will have to use two different kinds of paper world *numerals* in the numerators.

collection
item
whole number
count
plain whole numeral

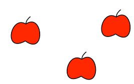
1. Magnitude of collections of items.

i. **Real world.** Since we get a real world **collection** of *identical* real world **items** just by gathering the real world items, determining how many real world items there are in a collection is simple: we get the **whole number** of real world items in the collection just by **counting** the real world items in the collection. .

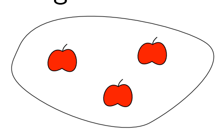
EXAMPLE A.1. The real world items



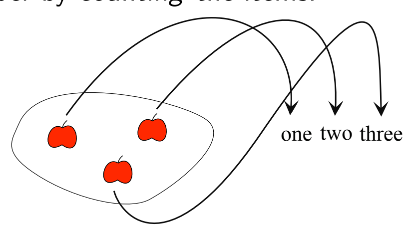
are *not* all the same and so *cannot* be gathered into a real world collection but the real world items



are all the same and so *can* be gathered into a real world collection:



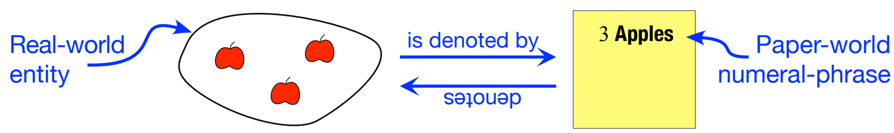
and we get the whole number by *counting* the items:



ii. **Paper world.** Collections of items are then denoted by paper world numeral phrases in which:

- The paper world numerator is the paper world **plain whole numeral** which says *how many* items there are in the collection, that is which denotes the real world whole number of items in the real world collection,
- The paper world denominator is the paper world word which says *what kind* of items are in the collection, that is which denotes the kind of real world items in the real world collection.

EXAMPLE A.2.



unit
decimal number
plain decimal numeral
decimal pointer
digit

where:
· The *numeral* 3 says *how many* items in the collection,
and where
· The *word* **Apple** says *what kind* items in the collection.

2. Magnitude of amounts of stuff.

i. **Real world.** Since *stuff* comes *in bulk*, determining *how much* stuff there is in an amount of stuff is much more complicated than deciding *how many* items there are in a collection of items because, in order to determine how much stuff there is in a real world amount of stuff, we first need a real world **unit** of that stuff. Only then can we determine the **decimal number** of units in the amount of stuff.

Which is why "The Weights and Measures Division promotes uniformity in U.S. weights and measures laws, regulations, and standards to achieve equity between buyers and sellers in the marketplace." *(`https://www.usa.gov/federal-agencies/weights-and-measures-division`)*

EXAMPLE A.3. Milk is *stuff* we drink and before we can say *how much* milk we have or want, we must have a real world *unit* of milk, say *liter* of milk or *pint* of milk.

ii. **Paper world.** Amounts of stuff are then denoted by paper world numeral phrases in which:

- The paper world numerator is the paper world **plain decimal numeral** which says *how much* stuff there is in the amount of stuff, that is, more precisely, the plain decimal numeral in which the **decimal pointer** indicates which **digit** corresponds to the unit of stuff in the denominator, which denotes the real world decimal number of units of stuff in the amount of stuff.
- The paper world denominator is the paper world word which says *what kind* of stuff in the amount of stuff and what unit of stuff.

Which points to its left.

EXAMPLE A.3. (Continued) *Then* we may say we have or want, say, 6.4 *liters* of milk or, say, 3 pints of milk.

It should be noted that decimal numerals work hand in hand with the METRIC SYSTEM of units while US Customary units usually require *fractions*, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, etc and mixed fractions.

Although panels on interstate roads have begun to show such things as 3.7 *Miles.*

3. Orientation of entities. Numerators can provide more information than just the magnitude of the entity, that is about the whole number of

orientation
signed whole numeral
signed decimal numeral
thing
give
qualifier

items or the decimal number of units of stuff, and can also provide information about the **orientation** of the entity by using **signed whole numerals** and **signed decimal numerals** instead of plain whole numerals and plain decimal numeral

4. Concluding remarks.

i. Since decimal numeral denote amounts of stuff while whole numerals denote collections of items, we absolutely need to distinguish *decimal* numerals from *whole* numerals.

EXAMPLE A.4. We need to distinguish the *decimal* numeral 27. which we would denote an amount of *stuff* from the *whole* numeral 27 which would denote a collection of *items*.

Told him it wouldn't! Didn't believe me! Wasted a lot of time trying anyway.

So, it would be tempting to agree that "The decimal point will *never* go without saying in this text." but, unfortunately, this is not really sustainable.
So, like everybody, we will have to agree that

> **AGREEMENT A.1** will often go without saying and we will often leave it to the reader to decide which kind of numeral is intended.

EXAMPLE A.5. When using money, pennies may or may not be beside the point:

Of course, sales people would write $11.99!

- We are more likely to write \$12.00 than \$12

but

- We are more likely to write \$7 000 000 than \$7 000 000.00.

ii. Altogether then, since the kind of numeral used in the numerator depends on:

A. Whether the real world entity we want to denote is:

- A collection of items

or

- An amount of stuff

and also on:

B. Whether the **information** we want about the real world entity is:

- The magnitude of the entity *alone*,

or

- The magnitude *and* the orientation of the entity,

the word numeral should always be used with one of the following **qualifiers**

	Collections	Amounts
Magnitude *only*	plain whole	plain decimal
Magnitude *and* orientation	signed whole	signed decimal

In fact, mathematicians, scientists, and engineers also use many other kinds of 'numbers' for many other kinds of entities. (https://en.wikipedia.org/wiki/Number)

Example A.6.

- 783 043 is a plain whole numeral which may denote a collection of *people*,
- 648.07 is a plain decimal numeral which may denote an amount of *money*,
- $-547\,048\,308$ and $+956\,481$ are signed whole numerals,
- $-137.048\,8$ and $+0.048\,178$ are signed decimal numerals.

And, since, as mentioned almost from the outset of **??** - Preface You Don't *Need* To Read (Page xv), this text assumes that the reader knows how to "compare, add/subtract, multiply/divide" signed decimal 'numbers', we will take the qualifiers plain/signed and whole/decimal to have been defined.

But you can always click on Appendix B - Localization (Page 515)

iii. However,

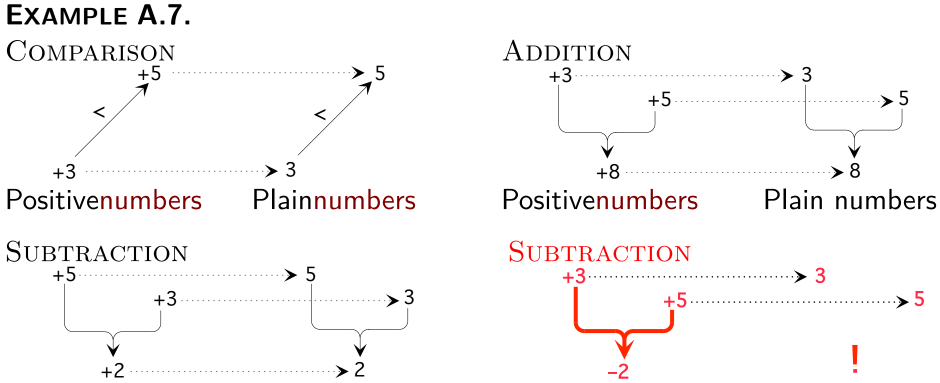
> **Cautionary note A.1** While Discrete Mathematics deals with collections of items, Calculus deals only with amounts of stuff and we will use whole numerals only occasionally and then mostly as an explanatory backdrop for decimal numbers.

2 Things To Keep In Mind

1. *Positive* numbers vs. *plain* numbers. Except for *subtraction*, computing with *positive* numbers goes exactly the same way as computing with the *plain* numbers that are their sizes..

And in only half the cases at that.

Example A.7.



absurd

Positivenumbers Plainnumbers Positive numbers Plain numbers

Positivenumbers Plainnumbers Positivenumbers Plain numbers

So it is tempting to skip the + sign in front of *positive* numbers as "going without saying". But then sentences lose their symmetry.

EXAMPLE A.8. The sentences

- "The opposite of $+5$ is -5" and "The opposite of -3 is $+3$"

are both nicely symmetric while the sentences

- "The opposite of 5 is -5" and "The opposite of -3 is 3"

both lack symmetry.

In fact, negative numbers were called **absurd** *numbers for a long time until "*Calculus made negative numbers necessary.*"(https://en.wikipedia.org/wiki/Negative_number#History)*

In other words, you get exactly what you see, no more, no less.

But then experience shows that skipping the + sign in front of *positive* numbers can lead to *ignoring* the difference between positive numbers and plain numbers and *that* leads to misunderstanding and mistakes because

- while working with *plain* numbers we can just focus on the numbers we are working with,
- when working with *positive* numbers we have to keep constantly in mind that the numbers we are working with have a sign, namely +, and therefore have opposites, namely negative numbers.

And so, in order to help distinguishing signed numbers from plain numbers and more individually positive numbers from their sizes, in this text:

In other words, in this *text, no sign does* NOT *mean* positive *but* plain *and therefore NO opposite.*

> **AGREEMENT A.2** will *never* go without saying.

EXAMPLE A.9. We will always distinguish, for instance,

- The *positive* number $+51.73$ from the *plain* number 51.73 which is the *size* of $+51.73$. (As well as the *size* of -51.73)
- The *positive* number $+64\,300$ from the *plain* number $64\,300$ which is the *size* of $+64\,300$. (As well as the *size* of $-64\,300$)

2. Symbols vs. words. Another issue is that, in everyday language, instead of using signed numbers we still tend to use plain numbers with *everyday words* instead of symbols to denote the orientation.

Yet, even banks, *which used to use plain numbers in two columns, one for debits, one for credits, now use signed numbers in a single column.*

EXAMPLE A.10. We often use *words* like credit and debit, left and right, up and down, income and expense, gain and loss, incoming and outgoing, etc instead of the *symbols* + and − to denote the *orientation* and using *plain* numbers to denote the *size*.

zero
0

3 Zero And Infinity

As simple as numbers may appear, numbers actually present several conceptual difficulties that we need to acknowledge. We also need to make sure that the words we will be using concerning numbers, if only in the PROCEDURES, will be perfectly clear.

Whose existence most CALCULUS texts blissfully omit even to mention.

1. Zero. There are two difficulties with zero that set **0** apart from other numbers and in fact already "*the ancient Greeks* [...] *seemed unsure about the status of zero as a number.*" (https://en.wikipedia.org/wiki/0#Classical_antiquity).

i. In the real world, there is no such thing as zero amount of stuff.

EXAMPLE A.11.
- There is no such thing as a perfect vacuum. (https://en.wikipedia.org/wiki/Vacuum).
- There is no such thing as an absolute zero temperature. (https://en.wikipedia.org/wiki/Absolute_zero)

And when we try to get 0 unit of *any* stuff *all* we get is the error! **??** **??** - **??** (??)

And that's no joke!

EXAMPLE A.12. 0 quart of milk denotes the amount of milk that appears to be in an empty bottle—but it might just be that the amount of milk in the bottle is too small for us to see.

Just how clean is clean?

So the difficulty is that since 0 does not symbolize *any* amount in the real world 0 cannot have a *size* to begin with.

ii. Even though mathematicians do distinguish $\mathbb{N}$, whole numbers *including* 0, and $\mathbb{N}^*$, the whole numbers *excluding* 0, AKA counting numbers, (https://en.wikipedia.org/wiki/Natural_number)

> **LANGUAGE NOTE A.1 Mathematicians accept 0 as a *signed* number** even though ... 0 has *no* sign!

non-zero
After months of waffling!
collection
item
Of course, to say the size is 0 merely moves the issue from signed numbers to plain numbers.
count
plain
whole
natural
positive integer
how many
how much
Isn't that convenient?
decimal
amount
stuff
orientation
magnitude
Remember that words between single quotes will be explained when their time comes.

Since it is standard practice, we will have to accept that

CAUTIONARY NOTE A.2 even if a number with *size* 0—and no sign.

So, what we will do is to distinguish **non-zero** numbers, that is all numbers except 0, from just numbers which include 0. So, all *non-zero* numbers have both a size and a sign.

4 Plain Whole Numbers

Because we can deal with **collection** of **items** *one by one*, describing how many items there are in a collection is easy: just **count** the items in the collection. Then, how many items there are in the collection will be given by a **plain** (as opposed to 'signed') **whole** (as opposed to 'decimal') number.

.

EXAMPLE A.13. Apples are *items*. (We can eat apples one by one.) To say how many are in the collection we *count* them that is we point successively at each while singsonging "one, two, three".

But not in this text.

LANGUAGE NOTE A.2 **Plain whole** numbers are also called **counting** numbers or **natural** numbers (https://en.wikipedia.org/wiki/Natural_number)—and, *incorrectly*, 'positive integers'.

==================================

decimal (as opposed to whole

An **amount** of **stuff** we can deal with only *in bulk*

orientation

magnitude that is how many items in the collection or how much stuff in the amount

LANGUAGE NOTE A.3 The word orientation is *not* too good but the words "*direction*" and "way" aren't either.

==================================

A lot of times, describing *how many* items we have or want in a collection or *how much* stuff we have or want in an amount of stuff is not enough and we also need to describe the *orientation* of the collection of items or of the amount of stuff: up/down, left/right, in/out, etc.

EXAMPLE A.14. How many people are *going into* or *coming out* of a building usually depends on the time of the day.
At least for the rest of us, how much money is *coming into* or *going out* of our bank account usually depends on the day of the month.

signed
size
quantitative
absolute value
numerical value
modulus
norm
| |
sign
qualitative
$+$
$-$
positive
negative
integers
the same

1. Size and sign. So, both **signed** (as opposed to plain) *whole* numbers and signed (as opposed to plain) *decimal* numbers carry *two* kinds of information:

- The **size** of a signed number (whole or decimal) is the **quantitative** information which is given by the plain whole number that describes *how many* items there are in the collection or the plain decimal number that describes *how much* stuff there is in the amount.

  > **LANGUAGE NOTE A.4** **Size** is called **absolute value** in most textbooks but some use **numerical value** or **modulus** or **norm**.

  The standard symbol for size is $|\ \ |$ but we will not use it and just write size of.

  EXAMPLE A.15. Instead of $|-3| = 3$ we will write: size $-3 = 3$.

- The **sign** of a signed-number (whole or decimal) is the **qualitative** information which is given by $+$ or $-$, the symbols that describe the *orientation* of the collection or of the amount, up/down, left/right, in/out, after a decision has been made as to which orientation is to be symbolized by $+$ and therefore which by $-$. Then,
  Positive (whole or decimal) numbers are the signed numbers whose sign is $+$,
  Negative (whole or decimal) numbers are the signed numbers whose sign is $-$.

  EXAMPLE A.16. $+17.43$ **Dollars** specifies a real world transaction:
  - The *size* of $+17.43$, 17.43, describes the *magnitude* of the transaction,
  - The *sign* of $+17.43$, $+$, describes the *orientation* of the transaction.

> **LANGUAGE NOTE A.5** **Signed *whole*** numbers are usually called **integers**.

But how could a plain whole number ever be called a positive integer?

Two signed numbers are:

- **the same** whenever they have the *same* size *and the same* signs. (So, when one is positive, the other has to be positive and vice versa.)

the opposite
opp
<
>
=
≦
≧
comparison (plain)
comparison (signed)
larger-than (plain)
smaller-than (plain)
equal-to (plain)
not-equal-to (plain)
larger-than-or-equal-to (plain)
smaller-than-or-equal-to (plain)
larger-than (signed)
smaller-than (signed)

- **the opposite** whenever they have the *same* size *but different* signs. (So, when one is positive, the other has to be negative and vice versa.)

We will use **opp** as shorthand for opposite of.

EXAMPLE A.17.
opp $(+32.048) = (-32.048)$ opp $(-32.048) = (+32.048)$

=======End LOOK UP =======

As implied by the title, operating on *plain* numbers, whole and decimal, is assumed to be known and this Appendix deals only with the complications brought about by the signs.

.
- **?? ?? - ?? (??)** • **?? ?? - ?? (??)** • **?? ?? - ?? (??)**

5 Comparing.

The symbols, $<, >, =, \leqq, \geqq$, are used for *both* **(plain) comparisons** *and* **(signed) comparisons**

DEFINITION A.1 Given the signed numbers x_1 and x_2,
- When x_1 and x_2 are *both positive*,
$$x_1 > x_2 \text{ iff Size } x_1 > \text{Size } x_2$$
$$x_1 < x_2 \text{ iff Size } x_1 < \text{Size } x_2$$
$$x_1 = x_2 \text{ iff Size } x_1 = \text{Size } x_2$$
- When x_1 and x_2 are *both negative*,
$$x_1 > x_2 \text{ iff Size } x_1 < \text{Size } x_2$$
$$x_1 < x_2 \text{ iff Size } x_1 > \text{Size } x_2$$
$$x_1 = x_2 \text{ iff Size } x_1 = \text{Size } x_2$$
- When x_1 and x_2 have opposite signs,
$x_1 < x_2$ iff x_1 is negative (and therefore x_2 is positive)
$x_1 > x_2$ iff x_1 is positive (and therefore x_2 is negative)

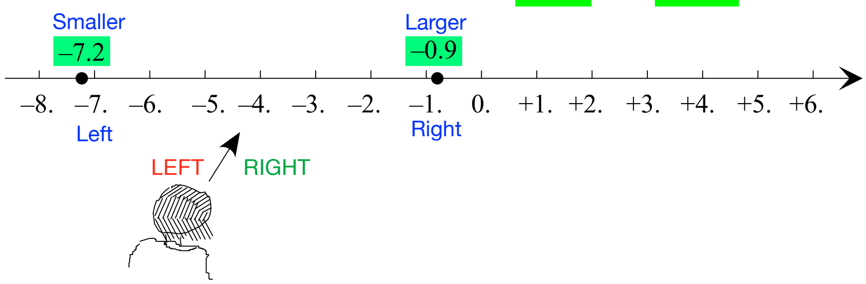
larger-than
smaller-than
equal-to
not-equal-to
larger-than-or-equal-to
smaller-than-or-equal-to
larger-than
smaller-than

equal-to
not-equal-to
larger-than-or-equal-to
smaller-than-or-equal-to

The easiest way is to picture the two numbers on a quantitative ruler and then, because of **?? ?? - ??** (**??**) , the number to *our left* will be **smaller than** the number to *our right* and the number to *our right* will be **larger than** the number to *our left*.

equal-to (signed)
not-equal-to (signed)
larger-than-or-equal-to (signed)
smaller-than-or-equal-to (signed)
smaller than
larger than

EXAMPLE A.18. Given the numbers −7.2 and −0.9. we have

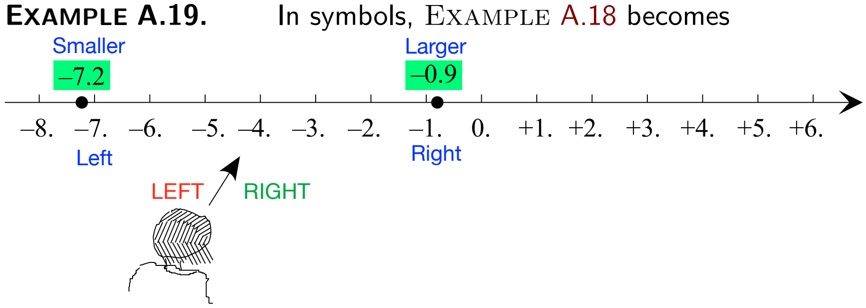


so −7.2 is smaller than −0.9 and −0.9 is larger than −7.2

The *standard* symbols for sign-size-comparisons of *all four kinds* of numbers are:

Sign-size-comparisons	Symbols
equal to	$=$
not equal to	$\neq$
smaller than	$<$
smaller than or equal to	$\leqq$
larger than	$>$
larger than or equal to	$\geqq$

EXAMPLE A.19. In symbols, EXAMPLE A.18 becomes



add
subtract
multiply
divide
reciprocal (plain)

so $-7.2 < -0.9$ as well as $-0.9 > -7.2$

6 Adding and Subtracting

. To **add**

In this text, for reasons explained in Subsection 6.2 - Mercator view (Page 108), when dealing with signed numbers, we will use the word oplus instead of the word add which we will reserve for plain numbers.

we will use the symbol $\oplus$

addition

To **subtract** a number we oplus its opposite instead.

subtraction

7 Multiplying and Dividing

. To **multiply**

MEMORY A.1 Multiplication and Division of Signs

	+	−
+	+	−
−	−	+

To **divide**

1. Reciprocal of a number.

i. The **reciprocal** of a *plain* number is 1. divided by that number. (https://www.mathsisfun.com/reciprocal.html). So:

i. Reciprocal 1. = 1.

ii. The reciprocal of 1 followed or preceded by 0s is easy to get: read the number you want the reciprocal of and insert/remove "th" accordingly,

EXAMPLE A.20.

Reciprocal 1 000. = 1 thousand th = 0.001

Reciprocal 0.000 001 = 1 million ~~th~~ = 1 000 000.

iii. The reciprocal of other numbers needs to be calculated and, for most, we may as well use a calculator.

EXAMPLE A.21.

reciprocal (signed)

Reciprocal 4.00 $= \frac{1.00}{4.00} = +0.25$ (Hopefully by hand.)

Reciprocal 0.89 $= \frac{1.00}{0.89} = 1.13$ (Use a calculator.)

Reciprocal 2.374 $= \frac{1.00}{2.374} = 0.421$ (Use a calculator.)

An important property of reciprocals is that:

> **MEMORY A.2 Sizes of *plain* reciprocal numbers**
> The larger a *plain* number is, the smaller its reciprocal will be,
> The smaller a *plain* number is, the larger its reciprocal will be.

Proof. □

EXAMPLE A.22.

ii. The **reciprocal** of a *signed* number is +1. divided by that number. So, getting the reciprocal of a signed number involves Memory A.1 - Multiplication and Division of Signs (Page 512) which complicates matters:

EXAMPLE A.23.

Reciprocal $+1\,000.$ = +1 thousand th $= +0.001$

Reciprocal $-0.000\,001$ = −1 millionth $= -1\,000\,000.$

Reciprocal +4.00 $= \frac{+1.00}{+4.00} = +0.25$ (Hopefully by hand.)

Reciprocal −0.89 $= \frac{+1.00}{-0.89} = -1.13$ (Use a calculator.)

Reciprocal −2.374 $= \frac{+1.00}{-2.374} = -0.421$ (Use a calculator.)

In particular, even just *stating* the extension of Memory A.2 - Sizes of *plain* reciprocal numbers (Page 513) to signed numbers is a bit complicated and is much easier done in Subsection 8.1 - Input level band (Page 119).

To be specific: **??** **??** - **??** *(??).*

relative

Appendix B

Localization

Inputs are counted from the origin that comes with the ruler. However, rather than counting inputs **relative** to the origin of the ruler, it is often desirable to use some other origin to count inputs from.

Appendix C

Equations - Inequations

The following is essentially lifted from Reasonable Basic Algebra, by *A. Schremmer*, freely downloadable as PDF from (Links live as of 2020-12-31):

- Lulu.com (https://www.lulu.com/en/us/shop/alain-schremmer/reasonable-basic-algebra/ebook/product-1m48r4p5.html?page=1&pageSize=4)

and/or

- ResearchGate.net (https://www.researchgate.net/publication/346084126_Reasonable_Basic_Algebra_Lulu_2009)

Appendix D

Addition Formulas

1 Dimension $n = 2$: $(x_0 + h)^2$ (Squares)

In order to get

Appendix E

Polynomial Divisions

1 Division in Descending Exponents

Since *decimal numbers* are combinations of powers of TEN, it should not be surprising that the procedure for dividing decimal numbers should also work for *polynomials* which are combinations of powers of x.

Appendix F

Systems of Two First Degree Equations in Two Unknowns

1 General case

xxxx xxxxx xxxxx

Appendix G

List of Agreements

Appendix H

List of Cautionary Notes

Appendix I

List of Definitions

Appendix J

List of Language Notes

Appendix K

List of Theorems

Appendix L

List of Procedures

Appendix M

List of Demos

Index