Generalized Arithmetic Operator Parametrized by a Real Number

Daren Scot Wilson

15 July, 2018

Abstract

The familiar operations of addition and multiplication are generalized to a real-number parameterized operation, $\varoast_r$ which bears the same relation to $\varoast_{r-1}$ as multiplication bears to addition. In particular, the distributive law is demanded to hold. This parameterized operation is defined by fractionally iterated exponential functions, of which two particular versions are described in detail. This paper is not intended to be deep or rigorous, but as an introduction and springboard for further exploration.

Generalized Arithmetic Operator Parametrized by a Real Number

Daren Scot Wilson

15 July, 2018

Contents

1. Motivation

When I was young I learned that multiplication is repeated addition. Like any bright curious kid, I asked what happens when one repeats multiplication. Powers and exponents, I was told. Squares, square roots, cubes, et cetera. It was not long before I found this answer deeply unsatisfying.

For behold:

$$A + B = B + A$$

and

$$(A + B) + C = A + (B + C)$$

and there's a magic value, *zero,* that turns the dyadic $+$ into a "do nothing" operator for the other argument. Analogous observations hold for multiplication. These are pleasing symmetries and properties, as useful in the real world of applied math as they are elegant in the world of pure theory.

Now given that multiplication is repeated addition

$$A \times B = \underbrace{A + A + \ldots + A}_{B\ of\ them} \tag{1.1}$$

(note that we use an explicit multiplication symbol) we might expect further glorious symmetry and elegance by defining

$$A \ast B = \underbrace{A \times A \times \ldots \times A}_{B\ of\ them} \tag{1.2}$$

This is normally notated as A^B for which the reader is well aware that

$$A^B \neq B^A$$

(except when $A = B$ and special cases) and that

$$(A^B)^C \neq A^{(B^C)}$$

and that there's no magic value that turns the proposed $\ast$ into a "do nothing" that works equally well leftward and rightward. Extension to a higher order operation – repeated power-taking – does not seem feasible or interesting. (Although it should be noted that as of this writing in 2018, Wikipedia has an article on an ugly operation called "tetration" [8] [2].)

However, there is a way to define $\ast$ such that commutativity, associativity and the other nice properites do hold. Furthermore, this definition allows easy extension to a higher-order operator, one that keeps these nice properties, and bears the same algebraic relations to $\ast$ as $\ast$ does to $\times$.

This process may be continued to create an infinite series of operators. One may also define operators going the other way: some operator ¤ having the same relation to $+$ as $+$ has to $\times$. All these operators may be seen as specific cases of one general operator parameterized by an integer, $❋_n$.

Whenever something is parameterized by an unbounded integer, it may be fruitful to ponder how that parameter could be understood as a real number. Is it possible to define an operator like

❋_n but of fractional order, such as $\text{❋}_{1/2}$? That is the subject of this paper.

2. Definitions and Derivation

We wish to define a parameterized function of two variables ❋_r where r is real having the following familiar group properties:

$$x \,\text{❋}_r\, y = y \,\text{❋}_r\, x \tag{2.1}$$

$$(x \,\text{❋}_r\, y) \,\text{❋}_r\, z = x \,\text{❋}_r\, (y \,\text{❋}_r\, z) \tag{2.2}$$

$$\exists 0_r : \;\; 0_r \,\text{❋}_r\, x = x \,\text{❋}_r\, 0_r = x \tag{2.3}$$

$$\exists \nu_r(x) : \;\; \nu_r(x) \,\text{❋}_r\, x = 0_r \tag{2.4}$$

where the Greek nu symbolizes the generalized inverse operator that we'll usually write as a function. We insist that ❋_r obeys a distributive law familiar from the theory of rings and fields, holding between operators differing in parameter by unity:

$$x \text{❋}_{r+1} (y \,\text{❋}_r\, z) = (x \text{❋}_{r+1} y) \,\text{❋}_r\, (x \text{❋}_{r+1} z) \tag{2.5}$$

We define the special cases

$$x \;\text{¤}\; y = x \text{❋}_{-1} y$$

$$x + y = x \text{❋}_0 y$$

$$x \times y = x \text{❋}_1 y$$

$$x * y = x \text{❋}_2 y$$

$$x \heartsuit y = x \text{❋}_{-1/2} y$$

$$x \spadesuit y = x \text{❋}_{1/2} y$$

$$x \clubsuit y = x \text{❋}_{3/2} y$$

Conventional addition and multiplication provide

$$0_0 = 0 \qquad \nu_0(x) = -x \tag{2.6}$$

$$0_1 = 1 \qquad \nu_1(x) = \frac{1}{x} \tag{2.7}$$

We think of ❋ as a function of two real variables with a parameter, but just as well could view it as a function of three real variables. We use it syntactically the same way as $+$ and will refer to it as an operator. Since it generalizes the well known arithmetic operators $+$ and $\times$, we'll call it the "generalized arithmetic operator".

We attempt to discover a formula for this general operator by first uncovering a formula for $x * y$. Write out the distributive law for regular addition and multiplication but lofted up one level:

$$x * (a \times b) = (x * a) \times (x * b) \tag{2.8}$$

Extend it to an arbitrary number of terms of identical values:

$$x * (\underbrace{a \times a \times \ldots \times a}_{B\ of\ them}) = \underbrace{(x * a) \times \ldots \times (x * a)}_{B\ of\ them} \tag{2.9}$$

We do not assume how many terms there are; we declare there are B of them and will not mind if it's non-integer, and assume the existence of real values a and B such that

$$y = \underbrace{a \times \ldots \times a}_{B\ of\ them} = a^B \tag{2.10}$$

Set a to the identity element $I \equiv 0_2$ for $*$ and assume it to be some value more interesting than zero or one.

$$x * y = x * (\underbrace{I \times I \times \ldots \times I}_{B\ of\ them}) = \underbrace{(x * I) \times \ldots \times (x * I)}_{B\ of\ them} \tag{2.11}$$

The identity element may be made gone of by its own definition:

$$x * y = \underbrace{x \times x \times \ldots \times x}_{B\ of\ them} = x^B \tag{2.12}$$

Solve 2.10 for B and plug in our value for a:

$$B = \log_a(y) = \frac{\ln(y)}{\ln(a)} = \frac{\ln(y)}{\ln(0_2)}$$

thus

$$x * y = x^{\frac{\ln(y)}{\ln(0_2)}} = \exp\left(\ln(x)\frac{\ln(y)}{\ln(0_2)} \right) \tag{2.13}$$

Symmetry is apparent. It seems natural to set $0_2 = e = 2.718\ldots$ and hope we haven't lost any interesting generality.

Let us recall that logarithms were originally invented for making multiplication easy. We observe:

$$x \times y = \exp(\ln(x) + \ln(y)) \tag{2.14}$$

and

$$x * y = \exp(\ln(x) \times \ln(y))) \tag{2.15}$$

and therefore the essence of $*$ may be reduced to addition,

$$x * y = \exp(\exp(\ \ln(\ln(x)) + \ln(\ln(y))\ )) \tag{2.16}$$

With little effort we find:

$$0_2 = \exp(\exp(0)) = e \tag{2.17}$$

and

$$\nu_2(x) = \exp(\exp(\ -\ln(\ln(x))\ )) = \exp({}^1\!/_{\ln(x)}) \tag{2.18}$$

Based on what we just did, it seems reasonable to declare

$$x \circledast_r y = \exp(\ \ln(x) \circledast_{r-1} \ln(y)\ ) \tag{2.19}$$

or, as we are about to define iterated exponentials,

$$x \circledast_r y = \exp_r(\ \ln_r(x) + \ln_r(y)\ ) \tag{2.20}$$

We take this last formula as defining $\circledast_r$. It is a simple exercise to verify that this satisfies the group properties and distributive law as demanded at the start of this section.

For the identity element we have:

$$0_r = \exp_r(0) \tag{2.21}$$

and

$$\nu_r(x) = \exp_r(-\ln_r(x)) \tag{2.22}$$

There is some usefulness of the concept of a "forcer". While the identity element combined with some arbitrary value x returns to us x, the forcer narcissistically does the opposite: it gives to us itself, making gone of its rival x. The obvious example is zero for multiplication. It'll snuff out anything. For addition, it's negative infinity.

$$\Phi_r \circledast_r x = x \circledast_r \Phi_r = \Phi_r \tag{2.23}$$

This has a simple relation to the identity element

$$\Phi_r = \log(0_r) = \exp_{r-1}(0) \tag{2.24}$$

At this point we have a definition for a generalized ‘arithmetic’ operator, but it is dependent on an understanding of iterated exponential functions. We explore this topic next.

3. Fractionally Iterated Exponentials

We have reduced our invention to a matter of iterated exponential functions. We look for general approaches to explore this, including old and new research [5].

First we establish what we mean by an iterated exponential, and describe how we wish to generalize the idea. We define

$$\exp_r(x) = \exp(\exp_{r-1}(x)). \tag{3.1}$$

or more generally, for any real r and s,

$$\exp_r(\exp_s(x)) = \exp_{r+s}(x) \tag{3.2}$$

with

$$\exp_0(x) = x \tag{3.3}$$

and

$$\exp_1(x) = e^x \tag{3.4}$$

Of course, we must have

$$\exp_{-1}(x) = \ln(x) \tag{3.5}$$

For convenience we define an iterated logarithm as an alternative notation for the iterated exponential with a negative order:

$$\ln_r(x) = \exp_{-r}(x) \tag{3.6}$$

At this point, we already have the tools we need to explore the operators $\ast$, ¤ and in general ❋_n for any integer n. The hard but interesting question is determining whether or not there can be such a thing as $\exp_{1/2}$ and its inverse.

How does one calculate iterated exponentials or logarithms with positive integer n? If one using a calculator, one merely punches the ‘exp’ or ‘ln’ key a few times, or not at all when $n = 0$. But there is no way to punch a key half a time! Can we find a pleasing analytic function $f : R \rightarrow R$ with the property

$$f(f(x)) = e^x\ ? \tag{3.7}$$

Luckily this question has been explored in the past. It turns out that there is not one unique answer, but infinite choice. Fractional iteration of real functions is a vast subject [3] best tackled with some basic yet powerful tools.

3.1. Abel's Equation

In the way logarithms help us with multiplication by mapping it to addition, perhaps iterated exponentiation can be mapped onto addition? For an arbitrary function f meeting certain easy requirments, an associated Abel's function is defined by [5] [7]:

$$A(f(x)) = A(x) + 1 \tag{3.8}$$

It immediately follows that

$$A(f(f(x))) = A(f(x)) + 1 = A(x) + 2 \tag{3.9}$$

The arbitrary-order iteration of f can therefore be defined by:

$$f_r(x) = A^{-1}(A(x) + r) \tag{3.10}$$

While the usual convention may be to use superscripts to denote iteration, we have already set out on the road of using subscripts to avoid confusion. We will be needing superscripts for their more mundane purpose of indicating powers. On the other hand, we follow convention in indicating the inverse of Abel's function.

We note that 3.8 only relates f_r with f_{r+1}, and with all values differing from r by an integer. We cannot interpolate. This is analogous to generalizing the factorial $x!$ to a continuous function of real values. Such a generalization $\phi(x)$ would have to obey

$$\phi(x) = x\phi(x-1)$$

for all real x and match $\phi(x) = x!$ for all non-negative integer x. Given any such $\phi(x)$ one may construct another $\phi'(x)$ using an arbitrary function $u(x)$ of period one for which $u(0) = 1$:

$$\phi'(x) = u(x)\phi(x)$$

The widely popular Gamma function $\Gamma(x)$ may be obtained by demanding conditions on smoothness, monotonicity, and a minor change of argument.

We are, of course, interested in the case of Abel's equation where $f(x) = e^x$ but will have occasion to deal with other functions. Assuming it is possible for A to exist, we expect some looseness, to be nailed down by one or more extra conditions. Given one solution A, we may create

$$B(x) = p(A(x)) \tag{3.11}$$

where p is a any continuous, strictly increasing function.[1] Plugging B into Abel's equation and fiddling around a bit, we see that

$$p(A(x) + 1) = p(A(x)) + 1$$

[1]Two of those conditions are actually not necessary. But nice to have.

We may replace $A(x)$ with some letter, say w. By differentiating $p(w+1) = p(w) + 1$ we see that the derivative of p must be periodic in w. We may write

$$p(w) = q(w) + w \tag{3.12}$$

where q is any periodic function with period 1. That's a lot of arbitrariness.

By demanding smoothness or other properties, a unique function may be defined. This paper describes three specific choices for A (or something equivalent) and therefore three versions of $\exp_r$ and ❋$_r$, two of which will be explored in depth.

3.2. Schröder's Equation

An alternative to Abel's Equation is to map iteration onto multiplication instead of addition [6]. While not profoundly different in terms of mathematical logic, Shröder's equation may be more convenient in practical use.

$$L(f(x)) = cL(x) \tag{3.13}$$

where $c > 0$ is some positive constant. The relation between Abel's and Shröder's equation is simple:

$$L(f) = c^{A(f)}$$

For our purposes, f is the exponential function, and we will be interested in performing calcuations using, and doing analysis with, the equation

$$L(\exp_r(x)) = 2^r L(x) \tag{3.14}$$

or one similar.

4. Kneser's Iterated Exponential

Hellmuth Kneser defined a fractionally iterated exponential [4] using a fixed point z_0 in the complex plane

$$z_0 = \exp(z_0)$$

$$z_0 \cong 0.31813 + 1.33723i$$

In a neighborhood of z_0, the effect of an exponential mapping takes on a simple form, and iteration of this mapping is easy to deal with. This, Schröder's equation, and some analysis leads to a method to fractionally iterate exponential functions.

While this is an interesting approach, we will not explore Kneser's iterated exponential in any depth. Among the reasons is that the necessity of dealing with complex numbers makes it awkward for application to real numbers. Yet, the idea of making use of a fixed point is a good one. We will keep this idea.

A side note: Kneser uses ψ for Abel's function and χ for Shröder's in his paper.

5. Szekeres' Iterated Exponential

In 1961 Australian mathematician G.Szekeres published a method [7] [1] of fractionally iterating the exponential function by demanding in some sense the "smoothest" Abel function for a related function g defined as[1]

$$g(x) = e^x - 1 \tag{5.1}$$

The key idea is to realize that this is asymptotically similar to $\exp(x)$,

$$g(x) \approx \exp(x), \qquad \text{for } x >> 1 \tag{5.2}$$

but with the advantage of having $x = 0$ as a fixed point with unit slope. This makes fractional iteration of g easy for small arguments.

We have picked this particular g from infinite possibilites by demanding its Abel function B defined by

$$B(g(x)) = B(x) + 1 \tag{5.3}$$

be maximally smooth in the sense of all orders of derivatives not changing signs along the positive real axis:

$$(-1)^k \frac{d^k}{dx^k} B(x) < 0 \qquad \text{for } k > 2,\ x > 0 \tag{5.4}$$

The function B relates to the Abel function A for the iterated exponential by

$$\lim_{x \to \infty} \frac{B'(x)}{A'(x)} = 1 \tag{5.5}$$

We will not belabor how these relations lead to the choice of $g(x)$ among all alternatives; the reader will find adequate discussion in Szekeres' paper.

The iterated g obeys:

$$g_{r+s}(x) = g_r(g_s(x)) = g_s(g_r(x)) \tag{5.6}$$

$$g_0(x) = x \tag{5.7}$$

$$g_1(x) = g(x) = e^x - 1 \tag{5.8}$$

$$g_{-1}(x) = \overline{g}(x) = \ln(z + 1) \tag{5.9}$$

[1]Szekeres uses 'e' for this function, but the author feels there are already too many 'e's running around, so we'll use 'g'.

where r and s are real. The overbar notation is sometimes nicer to use in the heat of algebraic battle. We'll call g_r the Iterated Szekeres Function.

5.1. Power Series for $g_r(x)$ for small x

The power series for $g(x)$ is barely different from the well known series for e^x:

$$g(x) = g_1(x) = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \tag{5.10}$$

When r and x are small we would like to use a truncatable power series

$$g_u(x) = S_1 x + S_2 x^2 + S_3 x^3 + \ldots \tag{5.11}$$

where we use u in place of r when it is "small" (without any strict definition of what "small" means.) The coefficients S_n are polynomials in u. We can determine these polynomials by jamming this series into the equality

$$g_u(g_1(x)) = g_1(g_u(x)) \tag{5.12}$$

and equating terms of equal powers of x. The reader will certainly enjoy spending all afternoon carefully working out the algebra; we will skip the details in this paper. We find the first few of what we'll call the Szekeres Polynomials to be:

$$S_0(u) = 0$$

$$S_1(u) = 1$$

$$S_2(u) = \frac{1}{2}u$$

$$S_3(u) = \frac{1}{4}u^2 - \frac{1}{12}u$$

$$S_4(u) = \frac{1}{8}u^3 - \frac{5}{48}u^2 + \frac{1}{48}u$$

$$S_5(u) = \frac{1}{16}u^4 - \frac{13}{144}u^3 + \frac{1}{24}u^2 - \frac{1}{180}u$$

$$S_6(u) = \frac{1}{32}u^5 - \frac{77}{1152}u^4 + \frac{89}{1728}u^3 - \frac{91}{5760}u^2 + \frac{11}{8640}u$$

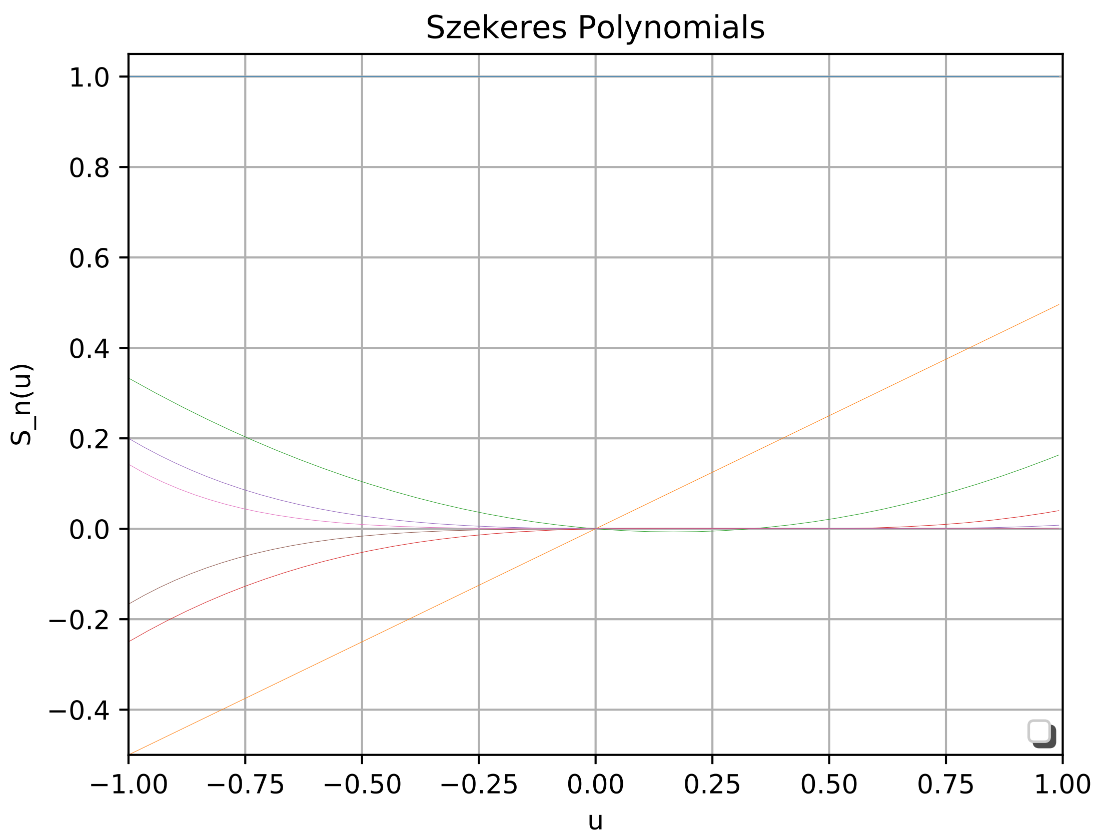
Substitution of special values $u = 0$, $u = 1$, and $u = -1$ give the expected values for power series for the functions x, $e^x - 1$, and $\ln(x + 1)$.

Of special note, we have

$$g_{1/2}(x) = x + 0.25x^2 + 0.0208333x^3 + 0.0002604x^5 - \ldots$$

$$g_{-1/2}(x) = x - 0.25x^2 + 0.1041667x^3 - 0.0520833x^4 + 0.0283854x^5 - \ldots$$

For $g_{1/2}$ there is no fourth power term.



In a practical calculation, if one does not wish to use the higher-order polynomials, which become more necessary for accuracy as u departs farther from zero, or when great accuracy is desired, one may use tricks such as

$$g_r(x) = g_{r/2}(g_{r/2}(x)) \tag{5.13}$$

at the risk of numerical truncation errors, and hoping $r/2$ and $g_{r/2}(x)$ are "small" enough.

As a practical note, if we truncate the series 5.11 after the S_4 term, we may expect about seven digit accuracy when $0 \le u \le .05$ and $0 \le x \le .05$

5.2. Techniques for Computing the Iterated Szekeres Function

What if x is not small? Indeed, we want to handle large x so that we may make the asymptotic approximation of Eq. 5.2. We use the identity

$$g_r(x) = g_H(g_r(g_{-H}(x))) \tag{5.14}$$

with integer H choosen so that $g_{-H}(x)$ is small enough to let us be comfortable with using the power series for the iterated Szekeres function for g_r. This is used for positive large x.

Then what if r is large? Any real number r may be decomposed into the sum of a limited range real value u and an integer n.

$$r = u + N \qquad \text{where } -\tfrac{1}{2} \le u \le \tfrac{1}{2}, \text{ integer } N$$

Then

$$g_r(x) = g_{H+N}(g_u(g_{-H}(x))) \tag{5.15}$$

While 5.15 is universally applicable to all x and r, we have in mind to use it for cases of large x and arbitrary r.

We can now handle fractional iteration of g to any order, for any given real number x, by using repeated application of the simple functions g and $\overline{g}$ (5.8 and 5.9) to bring our work into a comfortable range and handling the funky part with the Szekeres Function power series 5.11.

We were originally interested in iterating exp for some probably moderate value of x. We make use of

$$\exp_r(x) = \exp_{-M}(\exp_r(\exp_M(x))) \tag{5.16}$$

or, if r is not sufficiently near zero but we wish it were, using $r = N + u$ as before

$$\exp_r(x) = \exp_{N-M}(\exp_u(\exp_M(x))) \tag{5.17}$$

We see a strategy for calculating $\exp_r(x)$ for any given x and r. The outline is:

1. Apply exp several times to x until you have a “large” value.
2. Apply $\overline{g}$ several times until you have a “small enough” value z.
3. Compute $g_u(z)$ using the power series and Szekeres Polynomials.
4. Apply g to that result several times to undo the accomplishments of step 2. Apply an extra, if needed, to be sure the result is “large”.
5. Compute natural logarithms repeatedly to undo the accomplishments of step 1, plus or minus a few to account for the N, the integer part of r, and to undo any extra g of the previous step.

To put it another way, renaming our integer iteration counts, one computes $\exp_r(x)$ by the formula

$$\exp_r(x) \;=\; \ln_A(g_B(g_u(\bar{g}_C(\exp_D(x))))) \tag{5.18}$$

where

$$r \;=\; A + B + C + D + u \tag{5.19}$$

and D and B are chosen to produce "large" values, C is chosen to produce a "small" value, and A cleans up.

5.3. Numerical Examples

As an example, let us calculate $\exp_{1/3}(4)$ by Szekeres' method. Start with the 4. Only two exponentiations gives us 5.1484E23, which seems large enough. Applying $\bar{g}$ four times to that result takes us below 1.0. One more for good luck, and we have

$$z = \bar{g}_5(5.148E23) = 0.673211$$

Now the funky part. For $u = 1/3$ the power series is

$$g_{1/3}(z) = z + 0.166667z^2 + 0.000206z^5 - 0.000120z^6$$

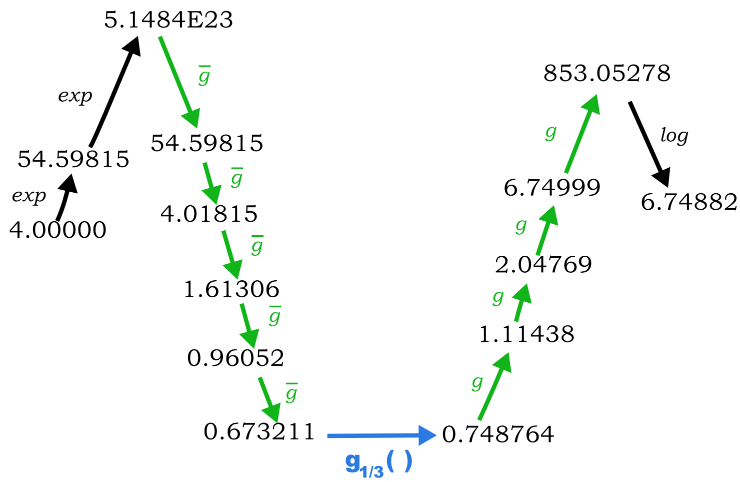
(We have gotten lucky; two of the low-order Szekeres polynomials have zeros at $u = 1/3$.) Let's name the result w, so we have $w = 0.748764$

As a quick check, we could pump w through that same power series, and put the result through it once more, that is to say, compute $g_{1/3}(g_{1/3}(g_{1/3}(z)))$, then compare that with $g(z) = \exp(z) - 1$. These come out to be 0.960502 an 0.960522 respectively, which are close enough for our purposes.

Now we apply g to w a few times until we reach "large" values. Four applications results in 853.05, and five results in overflow[1] This large-ish value is now to be taken down to the final result. We had used five $\bar{g}$ but only four g, so we must use one less log coming down than the two exp we had used at the start of this computation. We don't have any integer N part to the desired index of iteration; $r = u$ here. So one logarithm gives us the final result:

$$\exp_{1/3}(4) = 6.74882$$

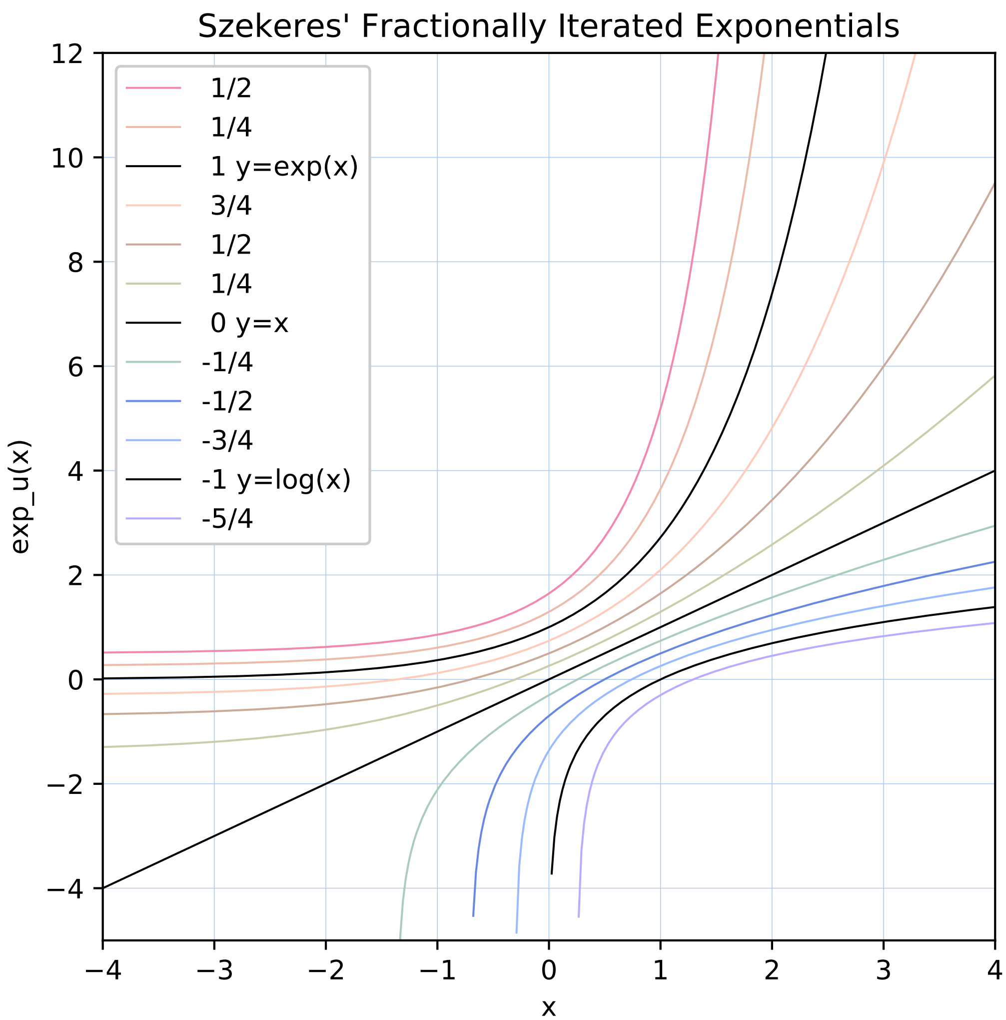
[1]The author is using Python 3.5 on 64-bit Arch Linux, which internally uses 64-bit IEEE floating point.



This diagram shows the overall process. We could have stopped with one exp during the initial step; 54 was plenty "large" enough unless great accuracy was needed. There are no guarantees on accuracy for this simple exercise. One might get a sense of how trustworthy this result is by computing a nearby value such as $\exp_{1/3}(4.001)$. Possibly, we could have done one or two more downward $\bar{g}$ steps and matching additional upward g steps for better accuracy when using the Szekeres power series.

5.4. Plots

The computation described above is easily implemented in software in any language capable of handling IEEE double precision floating point or better. Plotting is no longer a tedious chore as it was during the early part of the author's research decades ago. Here is a plot of iterated exponentials from the ordinary logarithm, to the zero-order identity function, to the ordinary exponential and beyond, with a few fractional orders in between. The curves come out in a pleasing arrangement.



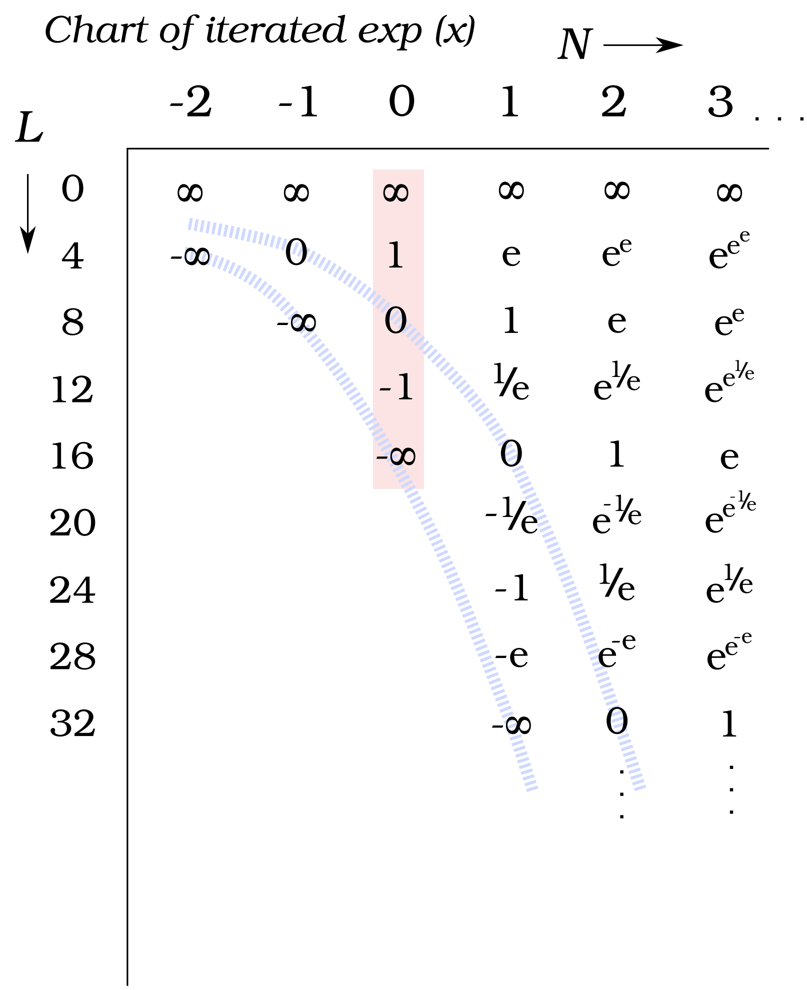
5.5. Extension to Complex Numbers?

Nothing stops us from feeding complex numbers (or worse) to the Iterated Szekeres Function power series or the Szekeres Polynomials. The mathematics of iteration is no different. We did not at any point in our derivations assume any properties special to real numbers, or at least, do anything that appears to prevent analytic continuation into the complex plane for either x or r. But we did assume that repeated application of exp, log, g and $\overline{g}$ would bring arbitrary values to "large" or "small" extremes. If the starting value has an imaginary component, then computation of exp can result in a value with a positive or negative real component. We may not reach "large" or be able to make the approximation 5.2 without some clever effort. The many-to-one mapping of exp with period $2\pi i$ and related multi-valuedness of logarithms suggest that we should expect trouble when following the above algorithm.

6. Symmetry-Based Iterated Exponential

For a completely different approach to iterated exponentials, let us draw on paper the entire range of real values $-\infty$ to ∞ in a finite vertical segment. Being Humans with a fondness for symmetry, we put zero in the middle, the infinities at either end, and 1 and -1 halfway between zero and either end. This is the pink $N = 0$ column in the figure below.

Next to this, in a column to the right, we mark those values as mapped by exp. To the left, log. We continue putting more columns outward both ways.



The columns are identical, except each is stretched to twice the height of the one to its left. The exp function maps the entire set of real numbers onto half of that set. We've extended ranges following our notion of symmetry.

Blue shaded curves show how any given value traces an exponential curve as we go from column to column. There's nothing to stop us from considering points partway between columns, such as where zero would be found in an imagined column at $N = \frac{1}{2}$.

We define a vertical coordinate, L, set arbitrarily to a maximum value of $\Lambda = 16$ at the location of $x = -\infty$ in the original (pink) column. The L coordinate of any value x, halved, gives the L coordinate of its image $\exp(x)$. We have summoned the spirit of Schröder's Equation[1] and have already, by intuitively sketching domains and ranges of exp, determined a specific

Schröder's function.

Without loss of generality, we could set $\Lambda = 1$ though we may choose other values for exploring and describing algorithms.

6.1. Generalized Negations

Negation of any value x corresponds to mirroring vertically about a point at the center of the column. Reciprocals are positioned symmetrically about 1 or -1; the transform is still a mirroring but within each half of the domain. The higher order inverses $\nu_2, \nu_3, \ldots$ act upon ever smaller segments of the whole domain. These segments form an infinitely deep hierarchy, shown on the left side of the illustration below.

Let us call the segment including $+\infty$ the "primary segment" of the operator ν_n wherein eq. 2.22 holds. All other segments would entail logarithms of negative numbers, were we to apply 2.22 directly. However, the generalized inverses commute freely, and each is its own inverse, so if we wish to compute inverses of values not in the primary segment we may make use of conjugations such as

$$\nu_3 = \nu_0 \nu_3 \nu_0 \tag{6.1}$$

or

$$\nu_4 = \nu_1 \nu_3 \nu_4 \nu_3 \nu_1.$$

The illustration below depicts with solid brackets the primary segments of each inverse, and with dotted line brackets the segments requiring conjugations.

6.2. Connecting Any Two Values

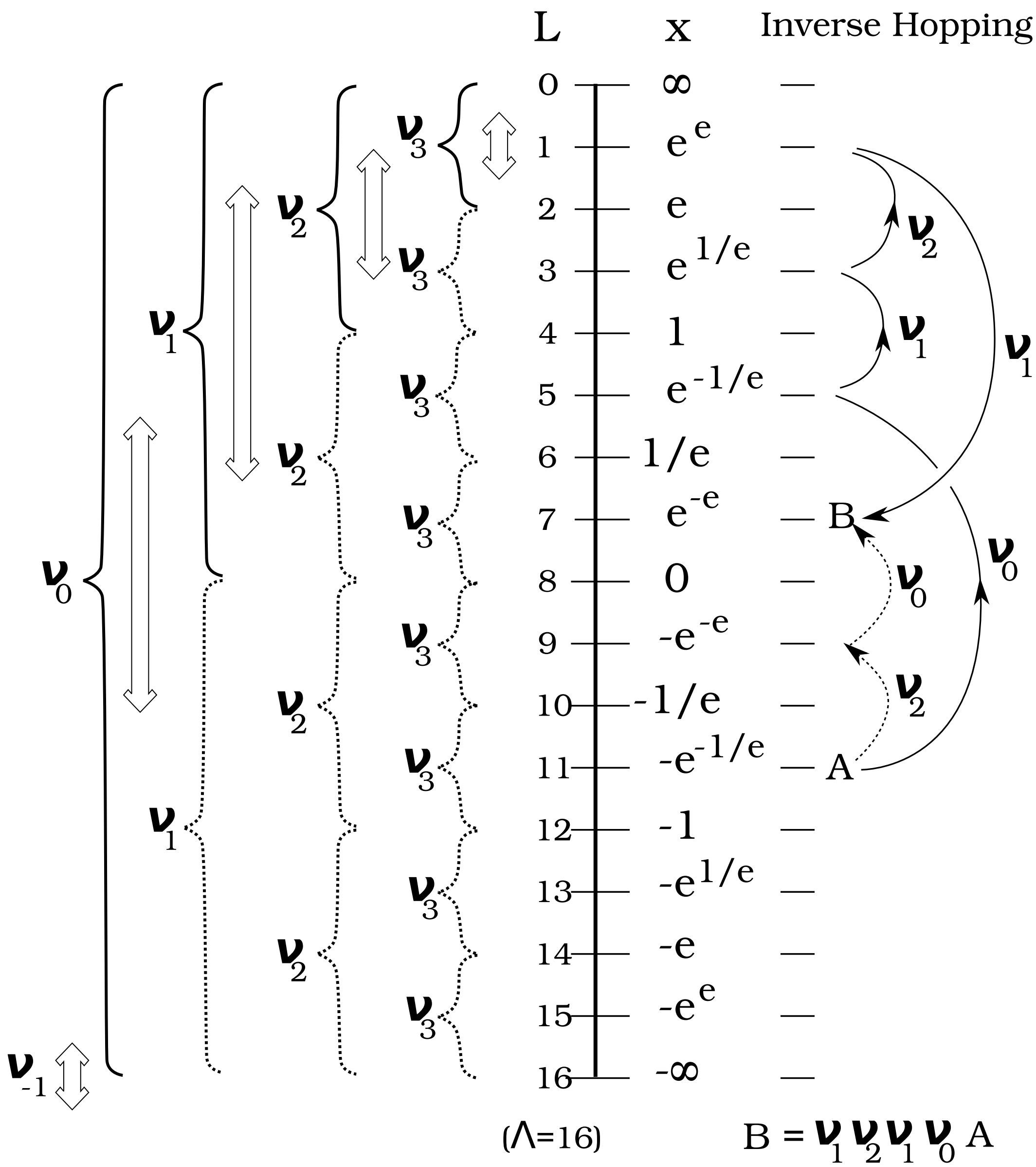
We may use a series of generalized inverses to map any given point **A** to any other given target point **B.** The illustration shows **A** being mapped to **B** by first applying ν_2 then ν_0 (dotted arrows). The first transform does not act on the primary segment, but by use of conjugations we may transform **A** to **B** staying entirely within primary segments, as shown by the solid arrows.

We will find it useful to have a strategy for determining a unique series of inverses, relying on only primary segments, to map any given **A** and **B.** In the example illustrated, first we see the two points are on opposite sides of zero, the fixpoint 0_0 of ν_0. We apply ν_0 to **A.** Negated **A** is then on the same side as **B** for the operator ν_1, so we do nothing. But they are on opposite sides of $^1\!/_e$, an equivalent of the fixpoint 0_2 for ν_2, so we apply that operator, but surrounded by a pair of ν_1. We have arrived at **B,** so we're done.

If **A** and **B** had not been sitting at integer L locations, we would have continued with ν_3, comparing to 0_3 or any of its images under the lower order inverses. There is no limit to how small the segments can get. We may reach arbitrarily close to any **B** with enough inverses.

---

[1]With the trivial replacement of $-r$ for r for the power of two.



If our target point **B** were fixed at, say, zero where we have $L = 8$, then we can relate the series of inverses to map given value x to zero, to a series of reflections of L within various segments. We know the final value $L = 8$. We can work backward from that to find the value of L for the original x. Computing a series of generalized inverses upon some starting value will involve applying several exponentiations followed immediately by several logarithms. We can reduce useless effort by working directly with exp and ln rather than inverses. With this, we concoct a practical algorithm to compute the symmetry-based Schröder's function, and its inverse. We provide this next, without detailed derivation.

6.3. Computation of Symmetry-Based Schröder's Function

First, for clarity let us use L as the name of Schröder's Function, and use λ for values resulting from it, $\lambda = L(x)$. These values range from zero to Λ.

Given any real number x, we compute its Schröder's function value $\lambda = L(x)$ by the following algorithm. We'll set $\Lambda = 1$. First we build a sequence of operations, then follow it in reverse to compute λ from an initial value based on a given Λ. In terms of computer science, this sequence behaves as a stack.

1. Start with an empty stack $S = []$
2. Compute $x \leftarrow \log(x)$ until x goes negative. Keep count (it'll be zero if the given x is already negative.) Push this count onto the stack S.
3. Negate the value: $x \leftarrow -x$
4. Go to step 2. Repeat until a stopping condition is reached:
   (a) x becomes zero, or sufficiently close to zero.
   (b) The information content of S, in terms of bits, is greater than the bits of precision one is calculating with.
5. Set $\lambda = {}^1\!/_2$.
6. Pop the most recent count off S. Update $\lambda \leftarrow {}^1\!/_2 \lambda$ that number of times.
7. Reflect $\lambda \leftarrow 1 - \lambda$
8. Pop off the next count to repeat the previous two steps, until S becomes empty, .

6.4. Algorithm for the Inverse Symmetry-Based Schröder's Function

Given λ, presumably after doing a multiplication according to eq. 3.14, here's how to calculate x:

1. Start with an empty stack $S = []$
2. Double $\lambda \leftarrow 2\lambda$ until it exceeds ${}^1\!/_2$. Keep count. Push that count onto the S.
3. Update $\lambda \leftarrow 1 - \lambda$
4. Repeat steps 2 and 3 until a stopping condition is reached: λ becomes 0.5 or very close, or when the sequence S becomes too long.
5. Set $x = 0$.
6. Pop off the top count from the stack. Replace $x \leftarrow \exp(x)$ that many times
7. Negate: $x \leftarrow -x$
8. Repeat with next count on stack, until stack is empty.

After a little practice, the reader will find that λ may be written in binary from inspection of S, and visa versa.

6.5. Numerical Example

We'll run the same numerical example as for the Szekeres case, $\exp_{1/3}(4)$. There are fewer decisions to make along the way, but to decide when to stop appending logarithm counts to the stack. Starting with $x = 4$ we find the sequence of counts

$$S = (3, 2, 2, 1, 2, 2, 2, 1, 2, 1, 3, 2, 2, 2, 1, \ldots)$$

From this we find

$$L = (binary)0.000110010011\ldots = (dec)0.0984234\ldots$$

The fancy "funky part" in the Szekeres isn't so fancy here. Exponentiating x to the $1/3$ order is just multiplying L by $\frac{1}{2}$ raised to that power:

$$L_{result} = L * (\frac{1}{2})^{\frac{1}{3}} = 0.0781187$$

The sequence of counts for this is:

$$S_{result} = (3, 1, 2, 11, 2, 1, 1, 2, 3, 2, 2, 2, 2, 1, \ldots)$$

Note the eleven only a few places in. If we were doing this work on a calculator, we would tap the "exp" key eleven times. Even if we started with $-\infty$ we'd overflow halfway through. The rest of the sequence won't matter. So chop it at the eleven, and decrement the previous count. We replace stack of operations with:

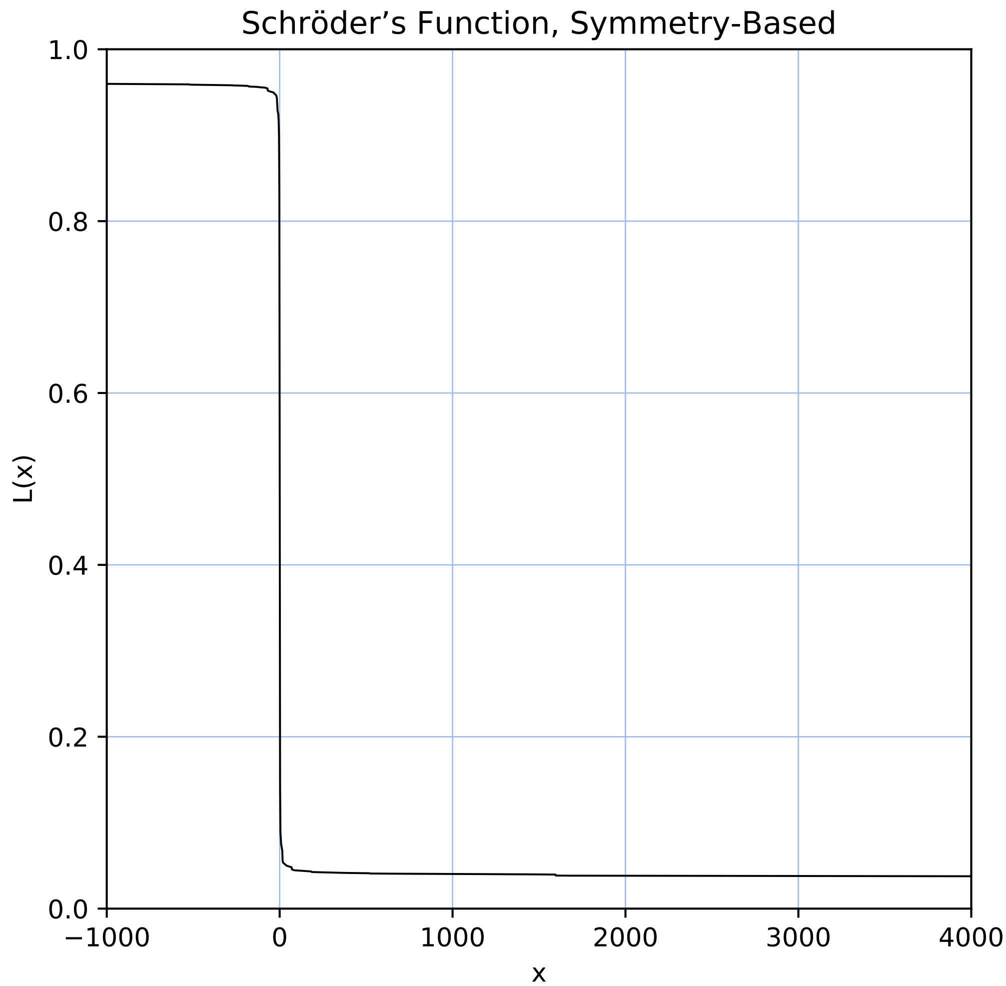
$$S = (3, 1, 1)$$

The result:

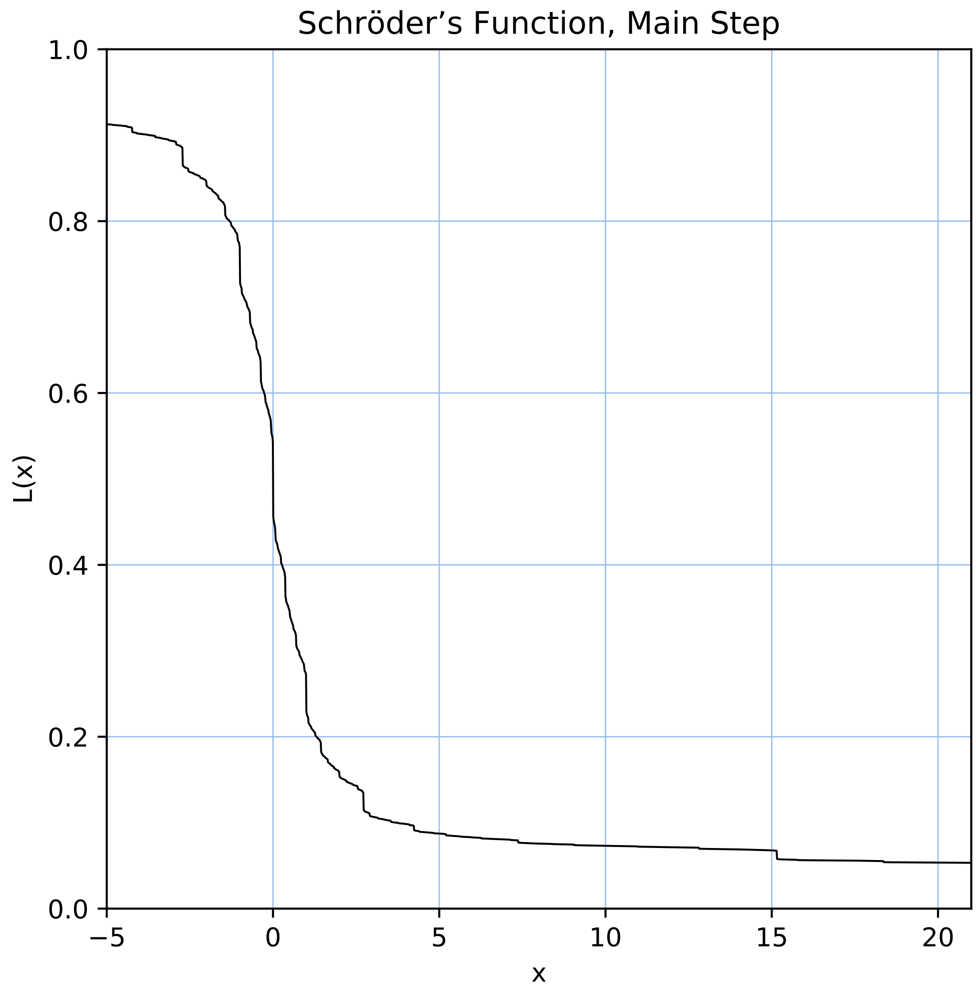
$$x = \exp_{1/3}(4) = 7.37509\ldots$$

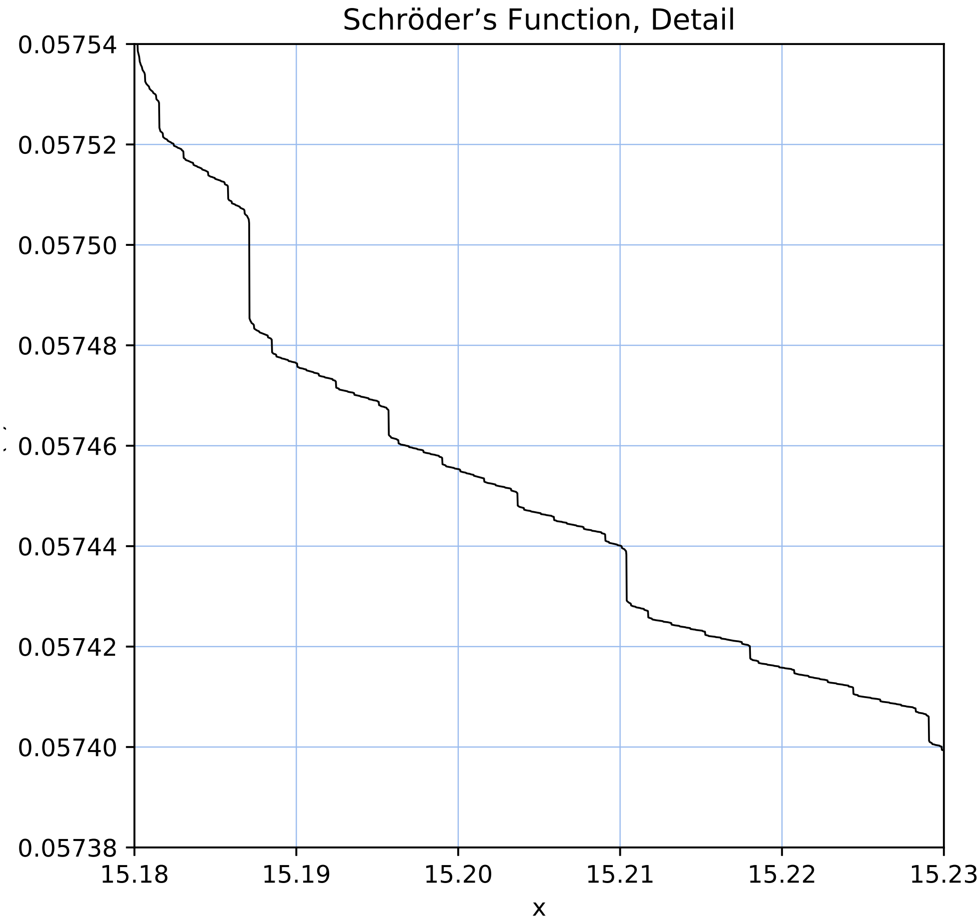
6.6. Plots

The symmetry-based iterated exponential is almost as easy to implement in software as Szekeres, but one must pay more attention to risks of overflow. Plots of the Schröder's Function show some strange properties. The reader may suspect sloppy plotting graphics in the following plot, where $-1000 < x < 4000$, in that the zero of the vertical axis is aligned with the black bottom line of the overall rectangle, but the function $L(x)$, which plunges mightily as x departs only a little from zero heading positive, seems to flatten out at some level definitely above the rectangle's bottom side. The plot as shown is accurate. For $x \approx$ a few thousand, we find $\lambda \approx$ 0.038 to 0.037. We must have x way bigger for λ to be visibly closer to zero.

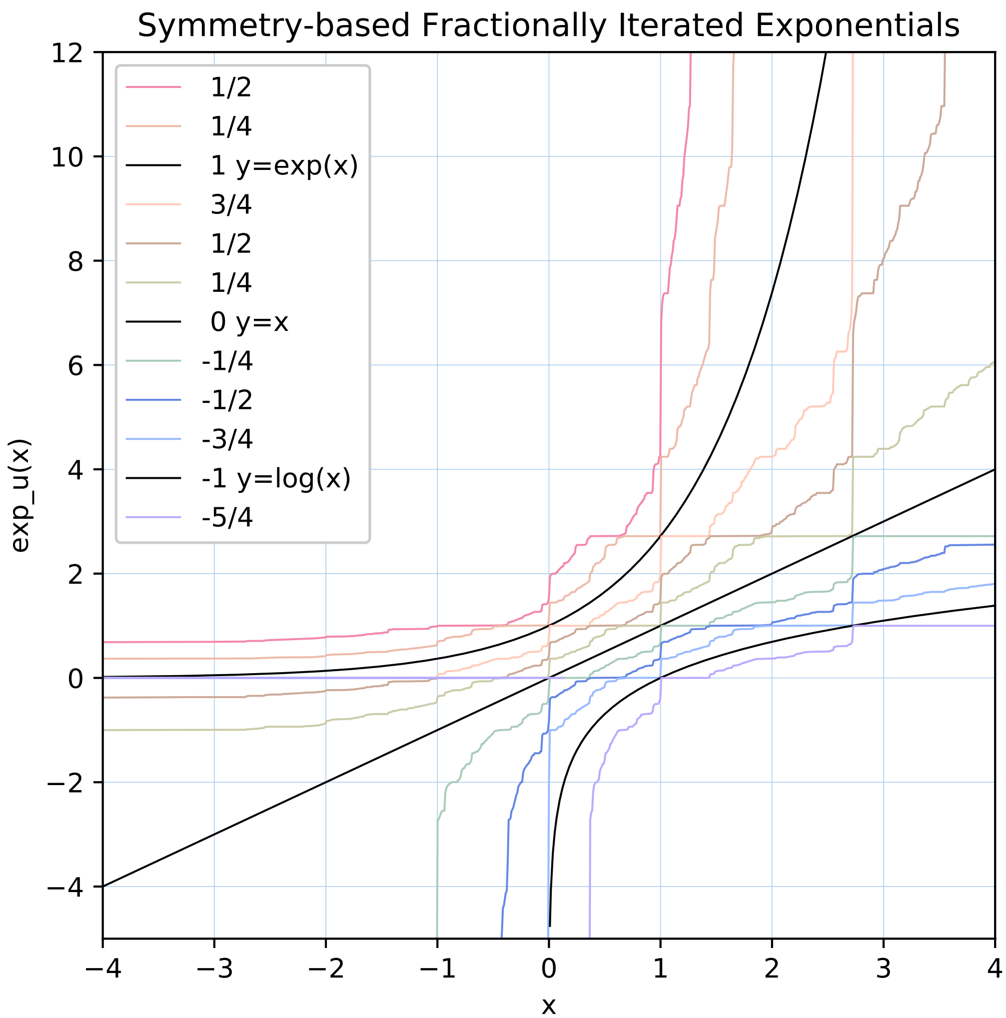


On the largest scale, in the first plot, we have just a somewhat sloppy downward step function. A closer look shows jaggies. The last plot shows jaggies on a finer scale resembling larger-scale jaggies. Jaggies remain visible on the finest scales plottable. This fractal-like appearance is not actually a fractal; one multiplies along one dimension while exponentiating along the other.

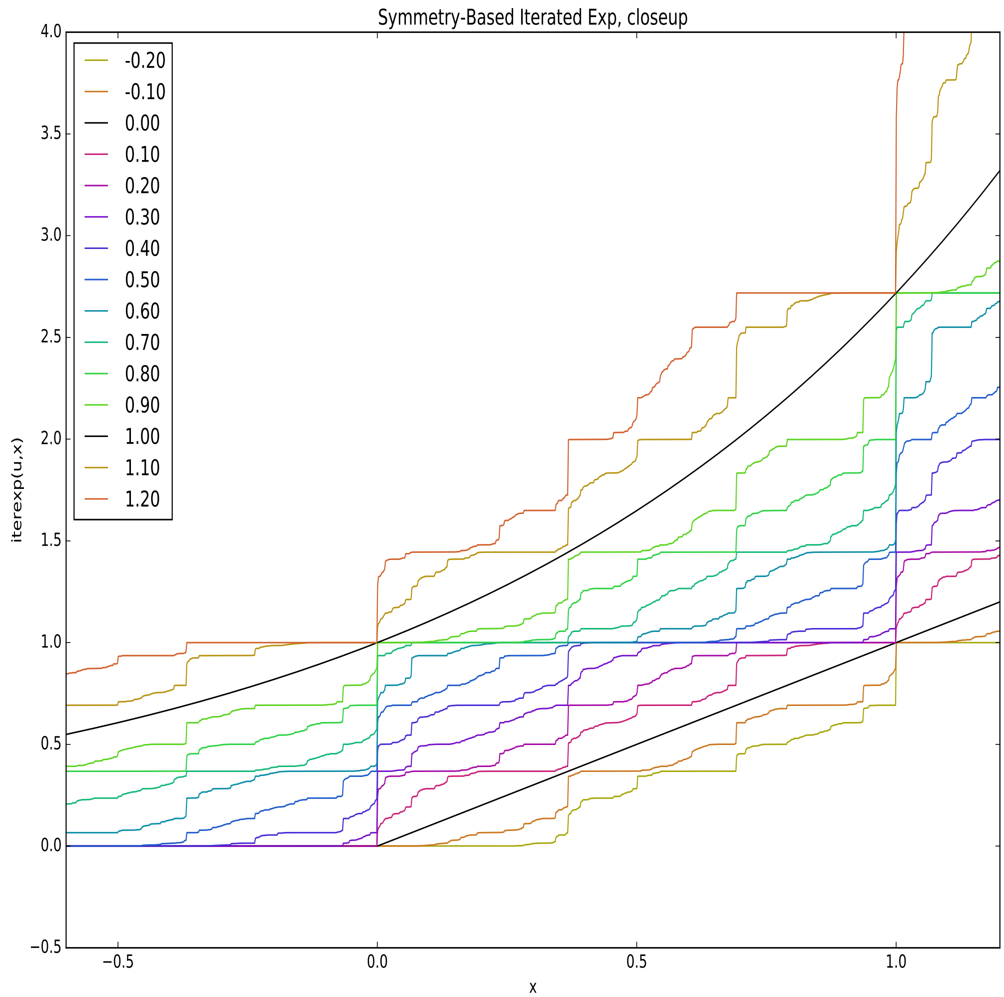

Schröder’s Function, Main Step
L(x)
x
1.0
0.8
0.6
0.4
0.2
0.0
−5
0
5
10
15
20



The next plot shows our symmetry-based fractionally iterated exponentials. For non-integer orders of iteration, these are very jagged. There are flat portions with no discernable slope, and steep rises. These features tend to align for differing orders of iteration, creating an appearance of horizontal and vertical lines crossing the coarsely meandering curves. If not for the use of different colors for different orders of iteration, one might suspect a problem in the software or plotting. All curves are, in fact, monotonically decreasing and non-intersecting.



A close-up of limited region shows the roughness of these functions in better detail:



6.7. Continutity and Smoothness

Claim 1: $L(x)$ and its inverse $L^{-1}(\lambda)$ are continuous, and monotonically decreasing.

Claim 2: … and so are their derivatives to all orders.

The author suspects a proof of the first claim would not be greatly difficult, but has neither the time nor expertise as of this writing to work on the second claim. The reader is invited to have a go at it.

6.8. Extreme Flatness or Steepness

Claim 3a: Any finite interval of x contains an infinite number of finite intervals over which the slope of $L(x)$ is less than any arbitrarily large negative number.

This could be turned around into a statement of flatness for $x = L^{-1}(\lambda)$. We'll phrase it more mathematically.

Claim 3b: For any given finite interval of λ contained within $(0, \Lambda)$, and any arbitrarily small real value ε, there exists an infinite number of finite intervals of λ over which $-\varepsilon < \frac{d}{d\lambda} L^{-1}(\lambda) < 0$

The explanation for these very flat portions can be seen in the procedures for computing $L(x)$ and its inverse. The step where one repeatedly exponentiates a value can lead to some wildly huge numbers. Then $x \leftarrow -x$ followed by at least one more exponentiation, leads to an fantastically small number, potentially smaller than any given ε. The next exponentiation, if the count on the stack says to do one, will produce a number fantastically close to unity. Whatever the subsequent operations, any small variation of the original value has had its influence on subsequent intermediate results compressed to a vanishingly small degree.

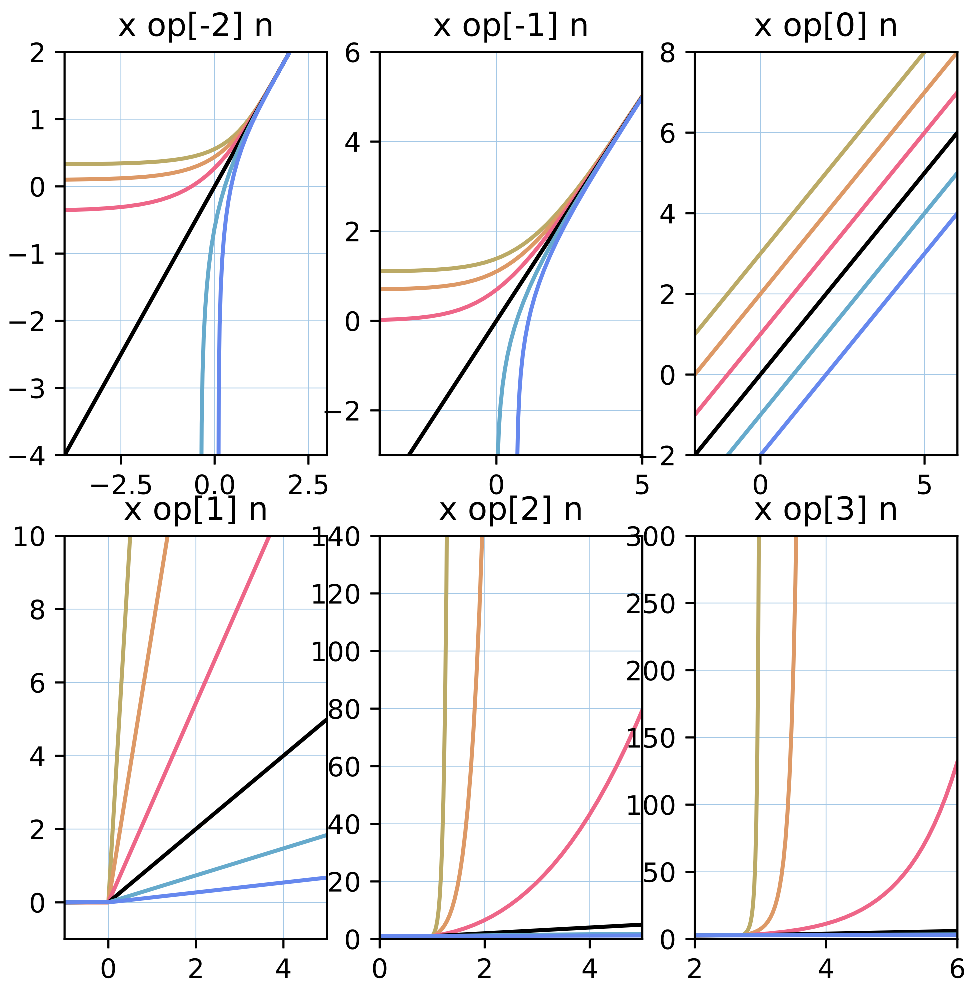
The λ value for some given x may have a binary bit pattern including, with no unusual rarity, runs of six, seven, eight or more of the same bit. We saw an eleven in the numerical example. A count on the stack of merely five can make one of these flat plateaus too flat to discern on a plot. Higher counts imply the need for larger numbers of bits for computing useful results. In practice, we may truncate the stack as in the numerical example.

After some consideration of the matter, the reader can see that these unboundedly extreme slopes correspond to runs of several 0's or several 1's in the binary expansion of λ. There is, of course, no limit to how long such a run can be, and an infinite number of such runs exist in the binary representations of all possible real values in any given range of λ however small.

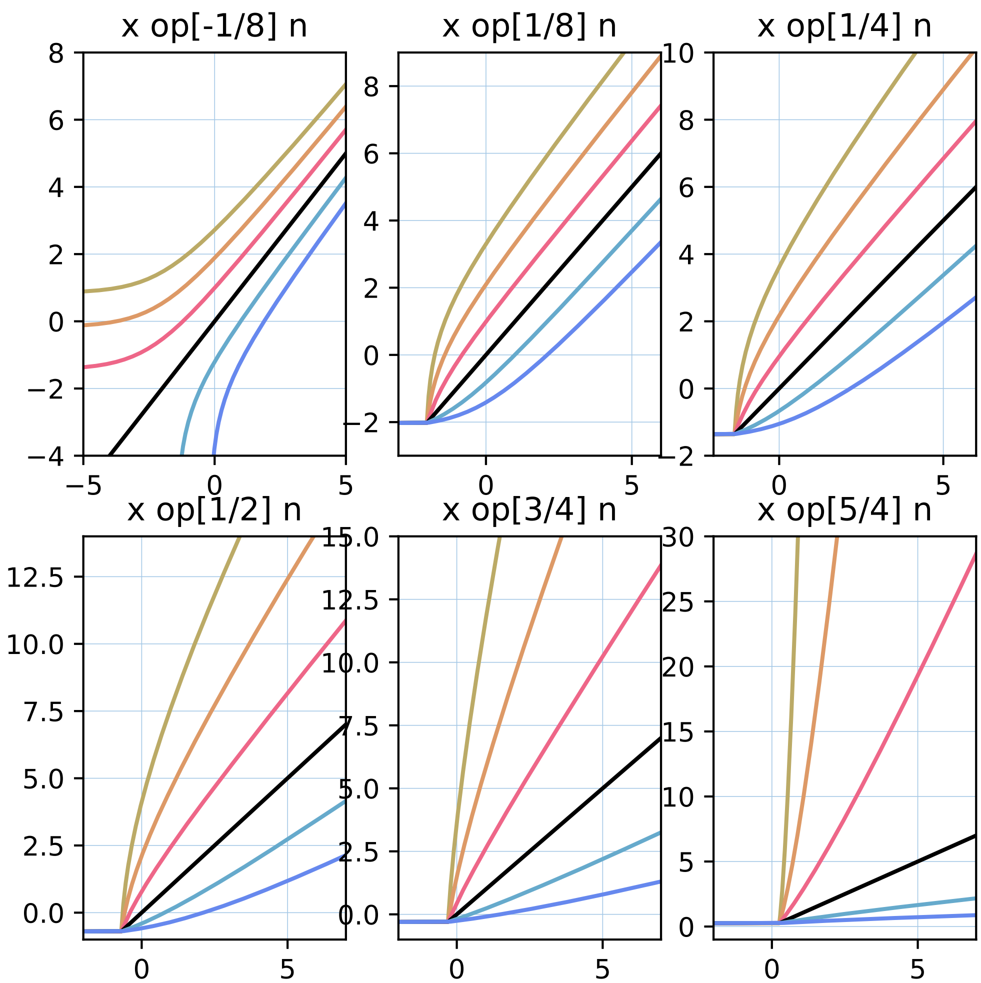
7. Plots: Generalized Arithmetic Operator

Let us return to the original purpose of this paper - to generalize the arithmetic operations of addition and multiplication. We now have a definition, depending on fractionally iterated exponentials, and we have two specific ways to dealing with those.

For the integer-order generalized arithmetic operators such as ¤ , ¤ , $\times$, $*$, it does not matter which Abel or Schröder's function we choose. We plot the values of some varying x combined with a constant, for several choices of operator order. The black line is the case of combining x with 0_n of that order, making $y = x$.



For fractional order operators, we show only the Szekeres case. Compared to the plain addition operator, order zero, we see that operators of slightly lower order cause the set of curves to splay apart for negative x. Slight positive order operators cause a gathering of curves to a point, at $x = -1$ for the order $^1/_8$ operator. Near that point the curves separate approximately as lines with different slopes. In all near-zero order cases, the curves become parallel for large positive x.



8. Directions for Further Research

This paper presents only motivation, definitions and initial explorations of an idea. Many deeper questions and ideas could be pursued.

1 The first thing that needs more work is simple: we can't deal effectively with values below the forcer for any given order of operator. For example, following strictly the methods of this paper, we can't multiply negative numbers. Using complex numbers allows us to carry around an extra πi when logarithming[1] that keep track of signs. But this doesn't generalize well to real-valued orders of operators.

2 When the common $+$ and $\times$ operators are restricted to working on only integers, or only

[1]Is this a word? It is now.

the natural numbers, interesting phenomena appear: Modular math, primes, the Goldbach Conjecture, and so on. What weird sidestreets and byways of abstract math might we create using $+$ and $*$, or $+$ and *ophalf* on special subsets of real numbers? How about defining higher-order "primes" as integers that can't be composited using $x * y$ for integer x and y? Could $*$ and the higher operators be of service in the study of the regular prime numbers?

3 Consider ❋$_\varepsilon$ where ε is close to zero. What approximation formulas can one write? What is the relation of ❋$_\varepsilon$ to $+$? Does it make sense to write differential equations where the order of operation r is the variable we differentiate with respect to?

4 Naturally, every curious mathematician will want to explore the behavior of $\exp_r(z)$ over the entire complex plane for z. It is not clear how the symmetry-based method could be generalized. Szekeres' method appears to have some hope, though, since the key part is defined by polynomials, and these are easy to generalize. But with an imaginary part to z, we no longer can simply go "up" or "down" a ladder of exponentiations or applications of Szekeres' function g to make use of the asymptotic approximation 5.2. Kneser pulls ahead of its competitors in this regard, being defined by methods of complex analysis.

5 Can the order of iteration r be extended to complex numbers? What does ❋$_i$ do? What are the properties of the function $\exp_z(0)$ for complex z?

6 $\exp(z)$ is a very well-behaved function over the entire complex plane, being analytic everywhere, single valued, the kind of function often termed "entire." Same is true for $\exp_0(z) = z$. In contrast $\ln(z)$ gives us trouble. In complex analysis we may treat it as multiple-valued, $\ln(z) = \ln(abs(z)) + i\arg(z) + 2\pi i n$ for all integers n. We speak of Reimann surfaces, or declare arbitrary (but practical) branch cuts. What happens for $\exp_r(z)$ when r is between 0 and -1? How does the topology of the Riemann surface change with order of iteration? Even with positive non-integer r it is not yet established how things look in the complex plane.

7 Asymptotic growth of analytic functions is often characterized by powers, exponentials, and logarithms. Between polynomial and simple exponential growth, we may find sub-exponential beasts such as $\exp(\sqrt{x})$. We have now added new beasts to the zoo. How does $\exp_{1/2}(x)$ grow? $\exp_{0.0001}(x)$? Would we have uses for stranger beasts such as $\exp_{1/2}(\sqrt{x})$? The iterated exponentials $\exp_r(x)$ with $0 < r < 1$ provide a new tool of exploration in this super-poly, sub-exponential realm.

8 Scientists often try to fit exponential or Gaussian curves to experimental data. Sometimes the fit isn't so good. A Gaussian function $A\exp(-cx^2)$ might be fudged a bit to something like $A\exp(-cx^{1.85})$ and A replaced with a polynomial. Additional terms may be added to provide for better fitting. Alternative formulas may be tried. The danger, of course, is the risk of overfitting and long messy expressions that are hard to optimize. Could iterated exponentials, or expressions joining terms using, say, $\spadesuit$, be of use in fitting measured data? Without a better theoretical foundation, computationally efficient approximations, and practical formulas for everyday use, this is unlikely anytime soon to help with data analysis or computational efforts.

9 We have looked at only symmetric operations over real numbers. $\circledast_r$ is an Abelian group operator. There is the huge body of knowledge in rings and fields to consider, besides the usual algebras based on real and complex numbers. Where else is allowed a concept of fractionally iterated exponentials? Can we use it to define operators like $\circledast_r$? How abstract can we go?

9. Summary

It is possible to extend the common arithmetic operators $+$ and x to an operator parameterized by a real number, but not by simply taking the idea of "repeated addition" to the next level. The distributive law and use of identity elements lead us to a simple formula, with arbitrariness settled by requiring a specific solution to Abel's or Shroeder's equation. We described two such solutions in sufficient detail for computations.

This paper was exploratory in nature; rigorous logic was scarcely seen. The author hopes that the ideas presented here inspire further research into the nature and applications of iterated exponentials and the operator $\circledast_r$.

Acknowledgements The author wishes to thank Dr. Stephen Wright of Oakland University for his support and critique during the early part of this work; J. R. Boyd and Kathy Alexander of Guilford College for discussions and suggestions; my fellow grad students at Indiana University and Colorado State University; and colleagues throughout my career for interest and questions.

This work was presented in earlier form [9] [10] at the Sixth Annual Conference on Undergraduate Mathematics held at Hendrix College in Conway Arkansas in 1981, and earlier at a gathering at Miami University in Oxford Ohio for which detailed information is regretably missing.

References

[1] Keith Briggs. The work of George Szekeres on functional equations, 2006. URL *http://keithbriggs.info, direct link to PDF http://keithbriggs.info/documents/Szekeres_seminar_QMUL_2006jan10.pdf*.

[2] Astrid E. Golomb. Super-Arithmetic. *J. Undergraduate Mathematics* **9** (1), page 11–16, page 11–16 (1977).

[3] Lars Kindermann. Iterative Roots and Fractional Iteration. URL *https://reglos.de/lars/ffx.html*. A list of references to papers, articles, etc.

[4] Hellmuth Kneser. Reel analytische Lösungen der Gleichung $\phi(\phi(x)) = e^x$ und verwandter Funktionalgleichungen. *J. f. Reine Angew. Math.* **187**, 56–67 (1950).

[5] Marek Kuczma. *Functional Equations in a Single Variable.* Polish Scientific Publishers, 1968.

[6] E. Schröder. Ueber iterirte Functionen. *Mathematische Annalen* **3**, 296–322 (1870).

[7] G. Szekeres. Fractional Iteration of Exponentially Growing Functions. *J. Australian Math. Soc.* **2** (3), page 301–320, page 301–320 (1962).

[8] Tetration, Wikipedia, 2018. URL *http://en.wikipedia.org/wiki/Tetration.*

[9] Daren Wilson. Fractional Order Arithmetic, part I. *J. Undergraduate Mathematics* **12** (2), 51–54 (1980).

[10] Daren Wilson. Fractional Order Arithmetic, part II. *J. Undergraduate Mathematics* **13** (?) (1981).