Introduction

I will start by ‘deriving’ the limit formula defining the constant e as $\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$ with the motivation being finding a base, b, such that the function b^x is its own derivative. Then I will attempt to generalize this for positive and negative powers of e.

Throughout this article, I will try to describe my thought process behind every step. As such, obvious disclaimer: I aim to be very intutive and beginner-friendly, and I do not pretend to be rigorous.

Finally, before diving into the article, readers may check out the Appendix at the end that covers elementary properties of exponentiation for a quick refresher.

Shrinking Step Sizes of Difference Quotient

So, for some $f(x)$, if I increment x by Δx, $\Delta y = f(x+\Delta x) - f(x)$. And the difference quotient is $\frac{\Delta y}{\Delta x} = \frac{f(x+\Delta x)-f(x)}{\Delta x}$. As $\Delta x \to 0$, the difference quotient converges to the derivative of $f(x)$.

My goal, then, is to find a function, $f(x)$ where this quotient is itself. So I want $\frac{\Delta y}{\Delta x} = f(x)$ which, solving for Δy necessitates that by incrementing x by Δx, $\Delta y = f(x) * (\Delta x)$. That is, Δy scales Δx by a scaling factor that is $f(x)$ itself.

2^x is a Step Size of 1 Approximation

I will straight up present the function $f(x) = 2^x$ as a function that when I take a $\Delta x = 1$ step of size 1, $\Delta y = f(x+1) - f(x) = 2^{x+1} - 2^x = 2 * 2^x - 2^x = 2^x = f(x)$ Thus, the difference quotient for this step size of 1 is $\frac{\Delta y}{\Delta x} = \frac{f(x)}{1} = f(x)$. And so the function $f(x) = 2^x$ satisfies the desired property of $\frac{\Delta y}{\Delta x} = f(x)$ for $\Delta x = 1$.

From Algorithms

Let me end this section by taking a brief, optional, detour to explain why 2^x was a very plausible function. Having a computer “science” background, I am familiar with properties of powers of 2 that arise when dealing with binary objects. There are several recursive algorithms including binary search, quick select, binary tree traversals, that take a problem of size N and in a single step, reduce it to a problem of size $\frac{N}{2}$. Thus the entire problem takes roughly $\log_2(N)$ steps and is very efficient. If I treat steps taken as x values and remaining problem size as $f(x)$ or y values, this results in exponential decay of $f(x) = N * \left(\frac{1}{2}\right)^x$. When $x = 0$, the problem size, or work left, is N units of work, when $x = 1$, the problem size, or work left, is $\frac{N}{2}$ units of work, and so forth. The first step when there is N work left saves $\frac{N}{2}$ work, the second step when there is $\frac{N}{2}$ work left saves $\frac{N}{4}$ work, and each successive step saves half of the work left. “Saving” per each step size where $\Delta x = 1$ is the negative of Δy. So saving from step x to step $x+1$ is $\frac{f(x)}{2}$ meaning $\Delta y = -\left(\frac{1}{2}\right) * f(x)$. But again, as $\Delta x = 1$, $\frac{\Delta y}{\Delta x} = \Delta y = -\left(\frac{1}{2}\right) * f(x)$ for $f(x) = N * \left(\frac{1}{2}\right)^x$. Note that N does not appear in the difference quotient at all. So I have found a function, $f(x) = k * \left(\frac{1}{2}\right)^x$ whose difference quotient scales itself by a factor of $-\left(\frac{1}{2}\right)$. Scaling itself is a crucial property and this suggests an exponential function is desired, though maybe not decay.

From Difference Quotient

A straight forward way to arrive at 2^x is to first assume I am searching for an exponential function. And then using algebra to set the difference quotient (with step size $\Delta x = 1$) of the function b^x equal to itself and solve for b to get $b = 2$:

$$\frac{\Delta y}{\Delta x} = f(x)$$

$$\frac{b^{x+1} - b^x}{1} = b^x$$

$$(b^x) * b - b^x = b^x$$

$$(b^x) * (b - 1) = b^x$$

$$b - 1 = 1$$

$$b = 2$$

Smaller steps

While $f(x) = 2^x$ satisfies $\frac{\Delta y}{\Delta x} = f(x)$ for $\Delta x = 1$, I want to find a different function that satisfies this constraint for a smaller step size.

Given the function 2^x is exponential and I'm working towards finding e where e^x, an exponential function, is it's own derivative, this new function for a smaller step size is presumably exponential as well in the form of $f(x) = b^x$.

Step Size of $\frac{1}{3}$

Given a function $f(x) = b^x$, what should b be, such that $\frac{\Delta y}{\Delta x} = f(x)$ for $\Delta x = \frac{1}{3}$? Clearly, $\Delta y = \left(\frac{1}{3}\right) * (\Delta x)$.

This can also be seen from similar triangles, like these 3 below: TODO draw 3 side by side triangles: 1st one base 1, height f(x), 2nd one base 1/3, height (1/3)*f(x), 3rd one base dx, height dx*f(x)

So when x increments by $\frac{1}{3}$, the resulting Δy needs to be $\left(\frac{1}{3}\right) * b^x$. Also, $\Delta y = b^{x+\left(\frac{1}{3}\right)} - b^x$ So solve for b:

$$b^{x+\left(\frac{1}{3}\right)} - b^x = \left(\frac{1}{3}\right) * b^x$$

$$b^x * b^{\frac{1}{3}} - b^x = \left(\frac{1}{3}\right) * b^x$$

$$b^{\frac{1}{3}} - 1 = \left(\frac{1}{3}\right)$$

$$b^{\frac{1}{3}} = 1 + \left(\frac{1}{3}\right)$$

$$b = \left(1 + \left(\frac{1}{3}\right)\right)^3$$

So when $b = \left(1 + \left(\frac{1}{3}\right)\right)^3$, the function $f(x) = b^x$ satisfies the property that $\frac{\Delta y}{\Delta x} = f(x)$ for $\Delta x = \frac{1}{3}$. This expression for b looks suspiciously like the limit definition for e. And more generally when $b = \left(1 + \left(\frac{1}{k}\right)\right)^k$, the function $f(x) = b^x$ satisfies the property that $\frac{\Delta y}{\Delta x} = f(x)$ for $\Delta x = \frac{1}{k}$. So as $k \to \infty$, $b = e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$.

My Analysis and Terminology

Ok, so I rarely like symbolic manipulation without explanation. Let me try and explain what happened, and to do so, I will introduce some concepts and terms. And in the next section, I'll use these concepts and terms to derive expressions for exponents of e.

Multiplicative Factor

The first term I will define is the “multiplicative factor” associated with a particular Δx.

I view slope and difference quotient as means to understand the behaviour of a function locally at some x, how $y = f(x)$ responds to some change Δx. The numerator of the difference quotient is $\Delta y = f(x + \Delta x) - f(x)$. Start at $(x, f(x))$ and end up at $(x + \Delta x, f(x) + \Delta y)$. Moving Δx from x induces the addition of Δy to y. $f(x + \Delta x) = f(x) + \Delta y$.

I feel it is very natural when analyzing exponential functions to consider how, for some Δx, y gets multiplied by some fixed multiple, the multiplicative factor, associated with moving this Δx at all x’s. For example, take $f(x) = 8^x$. As shown in the appendix, taking a step size of $\Delta x = 1$ corresponds to a multiplicative factor of 8. And taking a step size of $\Delta x = \frac{1}{3}$ corresponds to a multiplicative factor of 2. More generally, the multiplicative factor associated with Δx for $f(x) = b^x$ is $b^{\Delta x}$. Because $f(x + \Delta x) = b^{x + \Delta x} = b^x * b^{\Delta x}$ So moving Δx from x induces the multiplication of $b^{\Delta x}$ to y, and what y gets multiplied by is the multiplicative factor associated with that particular Δx. $f(x + \Delta x) = f(x) * \text{multiplicativeFactor}$.

Growth Factor

The second term I will define is the “growth factor” associated with a particular Δx as $\text{growthFactor} = \text{multiplicativeFactor} - 1$, so 1 less than the multiplicative factor for that same Δx.

Why subtract 1? Recall taking a small step from any $(x, f(x))$ places me at $(x + \Delta x, f(x) * \text{multiplicative Factor})$. So substituting growth factor in, this places me at $(x + \Delta x, f(x) * (1 + \text{growthFactor}))$. Or $(x + \Delta x, f(x) + f(x) * \text{growthFactor})$.

The idea is that the the multiplicative factor has the action of scaling $f(x)$ so the resulting y coordinate, $f(x + \Delta x) = \text{multplicativeFactor} * f(x)$, is in terms of $f(x)$. And $f(x)$, the starting y coordinate before taking the x step of Δx, is clearly in terms itself, like $f(x) = 1 * f(x)$. So since the starting coordinate is in terms of $f(x)$ and the ending coordinate is in terms of $f(x)$, their difference as well naturally can be viewed in terms of $f(x)$ as well.

And that’s the role of the growth factor. Taking a step Δx induces a change $\Delta y = f(x) * \text{growthFactor}$. This demonstrates a key property of exponents, that the Δy starting at a y value of $f(x)$ is written in terms of $f(x)$. Just like how, for e, I want $\frac{\Delta y}{\Delta x}$ to be $f(x)$.

Setting the “growthFactor” to be Δx makes sense as I’ve already shown. $\frac{\Delta y}{\Delta x} = \frac{f(x) * \cancel{\Delta x}}{\cancel{\Delta x}}$ TODO link the 3 side-by-size triangles picture.

In this case of deriving e, I set the growth factor to be Δx, but I will soon play around with different growth factors and introduce a new term (growth rate = growth factor / delta x) in the process. And for e with a growth factor of $(1) * \Delta x$, the growth rate is simply 1. As a preview, growth rate is finally an actual, well established term in math that’s not something I made up: typically in math and science, growth rate is referred to with upper or lower case r or λ.

Putting It All Together

So, approximating the base e, the growth factor taking an extremely small step Δx should be Δx itself. While means the multiplicative factor associated with this extremely small Δx should be $1 + \Delta x$. And, again, for any base, the multiplicative factor induced by Δx on the function b^x is $b^{\Delta x}$. So, as $\Delta x \to 0$, $\text{multiplicativeFactor} = 1 + \Delta x = e^{\Delta x}$.

To go from $e^{\Delta x}$ to e^1, raise $e^{\Delta x}$ by $\frac{1}{\Delta x}$. Δx is very small, so $\frac{1}{\Delta x}$ is very large and represents how many times the multiplicative factor for step size Δx must be compounded. For simplicity, if I want to deal with clean integers, let the small $\Delta x = \frac{1}{k}$ for a large k. Then $\frac{1}{k}$ cleanly divides 1, so simply raise the multiplicative factor by k to restore 'e'. That is, $e^{\frac{1}{k}} * e^{\frac{1}{k}} = e^{\frac{2}{k}}$, $e^{\frac{1}{k}} * e^{\frac{1}{k}} * e^{\frac{1}{k}} = e^{\frac{3}{k}}$, and so, k factors of $e^{\frac{1}{k}}$ will restore e^1. See the Appendix if this seems confusing. So I'm done, the multiplicaitive factor when substituting Δx with $\frac{1}{k}$ is $1 + \frac{1}{k}$ and it needs to be raised, again, substituting $\frac{1}{k}$ for Δx to the kth power. And as $k \to \infty$, this is the limit definition of e.

Connection with "Continuous Compounding"

This section is a quick detour, but consistent with all material so far. Viewing the growth factor in terms of ΔX made sense from an algebraic perspective, so as to cancel out this ΔX with the ΔX denominator when computing the difference quotient derivative approximation. But I want to briefly also explain, justify even, this by linking it to the "continuous compounding" interpretation and expression for e, which as how I and probably most others were first introduced to this mysterious constant.

Start by approximating continuous compounding as compounding once every Δt units of time. (t and x are different variable names for the same input, independent variable). So this Δt is extremely tiny. Say I have amount a at $t = 0$. And after a single compounding event, so now at $t = \Delta t$, the increase I get, this growth factor, is $ar(\Delta t)$. r is the growth rate. And as Δt is tiny, so is this resulting product. Of course, the question is, why multiply by Δt? This comes from approximating continous compounding. The key is the relation of the number 1 and Δt, namely $k = \frac{1}{\Delta t}$ (Mentally, to me, this is the same as period Δt and frequency k (but replace 2π default trig period length with 1 time unit length)). Continous compouding means per 1 unit of time, there are k compounding events. As Δt is miniscule, k is insanely large, and with the remainder being upper bounded by Δt, I view k as an integer. And I view multiplying by Δt as a scaling operation, namely, scaling down by a factor of $k = \frac{1}{\Delta t}$. But that's precisely how continuous compounding works, rather than applying the interest rate or growth factor r once at $t = 1$ after a unit of time, apply $\frac{r}{k}$, a k-scaled down rate interest, but instead compensate by compounding it k times.

Exponents of e

My motivation for this article was to understand the derivation of the exponential distribution as per my CS70 (Discrete Math Probability Theory) notes. In particular, the introduction of the lambda variable felt slightly arbitrary to me. And I also felt rusty in my understanding of the number e and exponentiation itself. So I compiled my thoughts in this article.

Positive integer powers of e

Ok, so for e, the associated growth factor was Δx, the step size itself. Note that this is a clean multiple of Δx. But what if I tweak the growth factor and double it to $2\Delta x$? Or set the growth rate, sometimes referred to as lambda, to 2? Then the new multiplicative factor becomes $1 + 2\Delta x$. So a small step of Δx now induces $\Delta y = y2\Delta x$. Whereas when lambda was 1 (the case for e), the associated $\Delta y = y\Delta x$. So with a lambda of 2, doubling lambda, the change in the output, $\Delta(f(x)) = \Delta y = 2y\Delta x$ doubles as well. So, given small deltas scale linearly (local linearity approximations), a half-step of $\frac{\Delta x}{2}$ would induce the same Δy that a full step would for e. So $\frac{k}{2}$ lambda = 2 steps would result in the same effect as taking k lambda = 1 steps. And as I showed, k lambda = 1 steps induces a multiplicative factor of e. So $\frac{k}{2}$ lambda = 2 steps induces a multiplicative factor of e so k steps would induce a multiplicative factor of e^2. So, k steps with the step size halved : $\frac{1}{k}$ to $\frac{1}{2k}$ (again, always allowed to do this as $k \to \infty$ anyways) with lambda 2 results in e^2. Thus $\lim_{n\to\infty} \left(1 + \frac{2}{n}\right)^n = e^2$ and more generally for positive integer powers lambda (λ), $\lim_{n\to\infty} \left(1 + \frac{\lambda}{n}\right)^n = e^\lambda$ And all of this is the chain rule at play.

Now to show this exact idea symbolically:

$$\lim_{n \to \infty} \left(1 + \frac{2}{n}\right)^n$$

$$\approx \left(1 + \frac{2}{k}\right)^k$$

(for large k)

$$\approx \left(1 + \frac{2}{2k}\right)^{2k}$$

(replace k with $2k$, $k \to \infty \Rightarrow 2k \to \infty$)

$$= \left(1 + \frac{1}{k}\right)^{2k}$$

$$= \left(\left(1 + \frac{1}{k}\right)^k\right)^2$$

$$\approx e^2$$

(approximate definition of e when k gets large)

Negative integer powers of e

TODO this section, feel tired and feel this warrants a bit more consideration and thought on my part, like not immediately obvious to me.. Want to show what did for last section except for lambda = −1.

Bonus: Logarithmic Differentiation to Prove Power Rule

I want to compute the derivative of x^n. The idea, to me, behind logarithmic differentiation is to view any base as a power of e. That is, $x = e^k \mid k = \ln(x)$. Now, given the base is e and as per the entire note thus far, a very small Δk induces a multiplicative factor of $e^{\Delta k} = 1 + \Delta k$ and a change in x, $\Delta x = x \Delta k$. Again because $\Delta x = \Delta\left(e^k\right) = e^{k + \Delta k} - e^k = e^k e^{\Delta k} - e^k = e^k\left(e^{\Delta k} - 1\right) = e^k(1 + \Delta k - 1) = e^k \Delta k$. This is consistent with the derivative, $\frac{\mathrm{d}x}{\mathrm{d}k} = x \mid x = e^k$, as Δk applied to $x = e^k$ scales that Δk by x itself. So I have the relation of Δx as Δk scaled by x itself, and again, this applies for any $x = e^k$ (even (e^n)^k which I will make use of soon), as regardless of x, the multiplicative factor induced by the Δk is the same, $1 + \Delta k$. Note that this goes the other direction as well, $\Delta k = \frac{\Delta x}{x}$ (thus the derivative of $\ln(x)$ is $\frac{1}{x}$). In fact, for the purposes of the derivative of powers of x, I eventually want to say that the Δx induces the Δk. but point is these deltas are related by the scaling factor of x, the starting value itself.

Ok, great. I have some connection between x and k and Δx and Δk. But what about x^n? How can I apply deltas to that? $x^n = \left(e^k\right)^n = e^{nk}$. So the Δx induces a Δk resulting in e^{nk} becoming $e^{n(k + \Delta k)} = e^{nk + n\Delta k} = e^{nk} e^{n \Delta k} = e^{nk}\left(e^{\Delta k}\right)^n$. So the Δk induces a multiplicative factor of $\left(e^{\Delta k}\right)^n$. Recall $e^{\Delta k}$ is simply the same multiplicative factor $1 + \Delta k$. I don't think the exact multiplicative factor's value is important insomuch as the fact that the new multiplicative factor is the one I showed earlier, raised to the n, so n applications of it. The change in exponent of $x^n = \left(e^k\right)^n$ is $n \Delta k$, where Δk was the change in exponent of x induced by Δx. Chain rule vibes where the exponent, rather than simply being incremented by Δk is now being incremented by that $n \Delta k$, a scalar multiple of n, so linearity as I know how $x = e^k$ reponds to Δk (increments by Δx such that $\Delta x = x \Delta k$) and note that e^{nk}, being an exponent of e, would respond to that same, lone, Δk added to its exponent the same way, the same multiplicative factor of $e^k = 1 + \Delta k$, so $\Delta(x^n) = \Delta(e^{nk}) = e^{nk+\Delta k} - e^{nk} = e^{nk}e^{\Delta k} - e^{nk} = e^{nk}(1 + \Delta k) - e^{nk} = e^{nk}\Delta k = x^n\Delta k$ (again, in computing $\Delta(x^n)$, Δk is scaled by the value of x^n, the value whose delta I am trying to compute). Now that was just for a single Δk, but there are n of them. And each Δk induces a multiplicative factor whose multiplication to x^n induces a Δx of $x^n\Delta k$. And by local linearity of Δx responding to a very small Δk, $n\Delta k$ will thus result in the true $\Delta(x^n) = x^n n\Delta k$. This is the chain rule at play. Another way I see this, is again small deltas idea: as $\Delta k \to 0$, $n\Delta k \to 0$, and for any very small change in exponent, be it Δk or $n\Delta k$, the change to e^{nk} induced by said very small change in exponent is that change itself multiplied by e^{nk} itself. Again, this reults in the true $\Delta(x^n) = x^n n\Delta k$ for a change in exponent of $n\Delta k$. This is a bit counter intuitive to me, n applications of a multiplicative factor behaving linearly, but that's the idea of local linearity is my rationale. Almost done, I have $\Delta(x^n) = nx^n\Delta k$, but to get the derivative with respect to x which is the entire problem, I need to write Δk in terms of Δx, and I've already established their relation, $\Delta k = \frac{\Delta x}{x}$, so $\Delta(x^n) = nx^n\frac{\Delta x}{x} = nx^{n-1}\Delta x \Rightarrow \frac{\Delta(x^n)}{\Delta x} = nx^{n-1}$. Done.

Appendix: Exponentiation Basics

I'll explore basics of exponentiation here using integers and motivate some properties of exponents, especially the property: $(b^k)^x = b^{k*x}$. Firstly, what does b^x mean? b^x evaluates to $\overbrace{b * b * \ldots * b}^{\text{x times}}$.

Symbolically, this is a product of x factors of b. Visually, I like to use trees that with branching factor b. For example, below shows a complete binary tree to represent 2^h, the case where $b = 2$.

TODO draw complete binary tree, a "2-Tree" here

The levels of these trees are 0-indexed, meaning at the 0th level, there is 1 node (the root), at the 1st level, there are b nodes, at the second level, there are $b * b$ nodes. Each successive level introduces another factor of b, since every node at the previous level splits into b more nodes. 1 node introduces b child nodes, 2 nodes introduce $2 * b$ nodes, all k nodes introduce $k * b$ children. Thus, at some level, l, there are b^l nodes, and the relation between successive levels is: $b^{l+1} = b^l * b$. And this relation naturally extends to $b^{l+k} = b^l * b^k$, that is, adding k to the exponent introduces k more factors of b that act on b^l.

Different bases

Let me add another base for consideration: 8^x. Below are 2 trees side-by-side that terminate with 64 leaves.

TODO draw these trees and make them line up, so distance between levels of the 8^x would be 3x that of 2^x tree

Observe that these two trees are quite closely related. Let me state the relation exactly as follows: every 3 levels of doubling for the 2-Tree produces the same effect of a single level of the 8-Tree.

So the 8-Tree is a 'compressed' version of the 2-Tree, by a factor of 3, based on the following equivalancy.

TODO draw another side by side picture of 3 levels of the 2-Tree and 1 level of the 8-tree, again lined up

Because $8 = 2^3 = 2 * 2 * 2$, 3 levels of doubling results in 1 level of multiplying by 8.

Let h be the height of the tree where if the lowest, leaf, level is indexed at l, $h = l - 1$. So the 8^h tree has 8^h leaves. When $h = 1$, there are 8 leaves. And when $h = 2$, there are 64 leaves. Now for the 2^h tree, when $h = 3$ there at 8 leaves. And when $h = 6$, there are 64 leaves. So, more generally, this shows that $8^h = 2^{3*h}$. But $8 = 2^3$, so $8^h = \left(2^3\right)^h$, and this proves $\left(2^3\right)^h = 2^{3*h}$ More generally, if X is some number as a power of b, say, $X = b^h$, then X^k multiplies the height of the b-tree representation of X by k. Note that this is only for integer values of k. I will, very soon, motivate this for rational powers as well (namely, $k = \frac{1}{3}$).

Finally, and this is, I suspect how most people including myself learned exponents, I can readily see all this when writing out factors: $8^2 = (8) * (8) = (2 * 2 * 2) * (2 * 2 * 2) = 2^6$. The number of factors is h, the argument of $f(h) = b^h$ and it is evident that the number of factors in the 8-expansion gets multiplied by 3 to get the number of factors in the 2-expansion. Like it takes 2 8's to write out 64 but it takes $6 = 2 * 3$ 2's to write out 64 using factors of all 2's. (If you are familiar with hexademical and binary numberings a similar compression by a factor of 4 happens where every hexadecimal digit valued from 0-15 can be converted into 4 binary digits)

OK, but what about instead of multiplying by 3, dividing by 3. Consider $8^{\frac{1}{3}}$. For the function $f(h) = 8^h$, the input h is the height. But a fractional height doesn't make sense? But if use the relation I just derived, where every 1 level of the 8-Tree is equivalent to 3 levels of 2-Tree, every 2 levels of the 8-Tree is equivalent to 6 levels of the 2-Tree, it follows that 1/3 level of the 8-Tree is equivalent to 1 level of the 2-Tree. That is, I'm assuming the ratio of 1 level 8-Tree : 3 levels 2-Tree,

$$X \text{ 8-level} = X\cancel{\text{8-level}} * \left(\frac{3 \text{ 2-level}}{1 * \cancel{\text{8-level}}}\right) = 3X \text{ 2-level}$$

or, equivalently,

$$X \text{ 2-level} = X\cancel{\text{2-level}} * \left(\frac{1 \text{ 8-level}}{3 * \cancel{\text{2-level}}}\right) = \left(\frac{1}{3}\right) X \text{ 8-level}$$

And so $8^{\frac{1}{3}} = 2^1 = 2$ and more generally, $b^{\frac{1}{k}} = x$ where $x^k = b$. And symbollically, this is readily displayed by $\left(b^{\frac{1}{k}}\right)^k = b$.