

DISTRIBUTIVITY, EXTENDED

(Tommy Burch, on and off from 08.2024 to 05.2025) (wxMaxima does not export the cleanest of latex files, so the resulting pdf contains a number of quirks and errors)

1 At or Down

This all started back in the late '90s at the UofA in a physics lab prep session (with R. McCroskey and R. Haar, among others). The discussion hinged on finding an operation (i termed it 'at' with the symbol @) over which addition was distributive. The exp / log solution below i came up with followed later the same day... Note that, for multiplication and addition, the distributive property can be "revealed" through exponentials and their inverse, the natural logarithms:

(% i1) `add(a,b) := log(exp(a)*exp(b));`

$$\text{add}(a, b) := \log(\exp(a) \exp(b)) \quad (\% \text{ o1})$$

(% i3) `add(a,b);`
`add(c*a,c*b);`

$$b + a \quad (\% \text{ o2})$$

$$bc + ac \quad (\% \text{ o3})$$

(% i4) `ratsimp(%);`

$$(b + a) c \quad (\% \text{ o4})$$

Placing the multiplication between the exponentials in the logarithm, we arrive at the relation (addition) over which it is distributive. We can try the same with + between the exp's...

1.1 At _ +

Define the @ (or "down") relation such that addition is distributive over it: $a + (b @ c) = (a+b) @ (a+c)$...Or: $a + \text{down}(b,c) = \text{down}(a+b,a+c)$. Trying the following ansatz:

(% i5) at_eq : At(a,b) = log(exp(a) + exp(b));

$$\text{At}(a, b) = \log \left(e^b + e^a \right) \quad (\text{at_eq})$$

(% i6) down(a,b) := radcan(log(exp(a) + exp(b)));

$$\text{down}(a, b) := \text{radcan}(\log(\exp(a) + \exp(b))) \quad (\% \text{ o6})$$

Check:

(% i7) down(a+b,a+c);

$$\log \left(e^c + e^b \right) + a \quad (\% \text{ o7})$$

yes! (It is also an obviously commutative operation.) This should work with any base for the power / log's: e.g.,

(% i8) log10(x) := log(x) / log(10);

$$\log_{10}(x) := \frac{\log(x)}{\log(10)} \quad (\% \text{ o8})$$

(% i9) radcan(log10(10^ (a+b) + 10^ (a+c)));

$$\frac{\log \left(2^c 5^c + 2^b 5^b \right) + (\log(5) + \log(2)) a}{\log(5) + \log(2)} \quad (\% \text{ o9})$$

(% i10) logcontract(%);

$$\frac{\log \left(2^c 5^c + 2^b 5^b \right) + \log(10) a}{\log(10)} \quad (\% \text{ o10})$$

yep! The ambiguity probably speak volumes (see, e.g., M. Burgin, arXiv:1010.3287 : there, the focus appears not to be in maintaining distributivity, unlike here, where it's the main focus). It may be worth exploring whether functions other than powers/logarithms can retain distributivity... at first glance, seemingly not. Some values with base e:

(% i11) down(0,0);

$$\log(2) \quad (\% \text{ o11})$$

(% i12) down(1,0);

$$\log(e + 1) \quad (\% \text{ o12})$$

(% i13) down(x,0);

$$\log (\%e^x + 1) \quad (\% \text{ o13})$$

(% i15) down(x,x);
down(x-log(2),x-log(2));

$$x + \log (2) \quad (\% \text{ o14})$$

$$x \quad (\% \text{ o15})$$

Associative:

(% i17) down(a,down(b,c));
down(down(a,b),c);

$$\log \left(\%e^c + \%e^b + \%e^a \right) \quad (\% \text{ o16})$$

$$\log \left(\%e^c + \%e^b + \%e^a \right) \quad (\% \text{ o17})$$

(% i20) diff(down(a,b),a);
diff(down(a,b),b);
diff(diff(down(a,b),b),a);

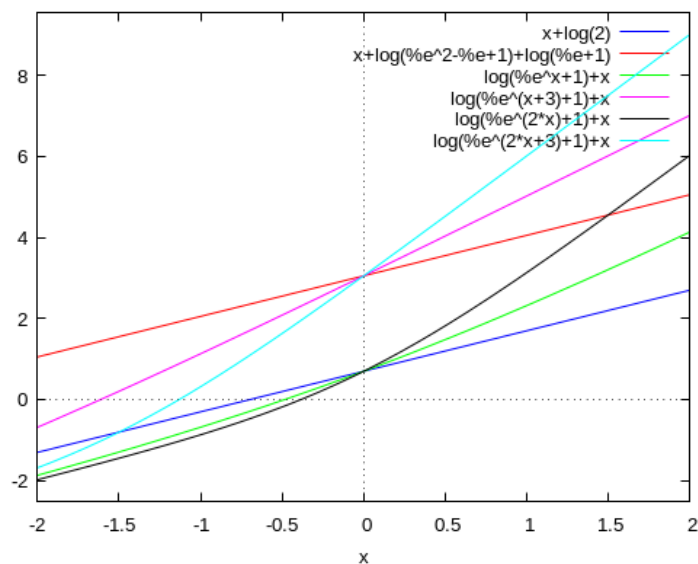
$$\frac{\%e^a}{\%e^b + \%e^a} \quad (\% \text{ o18})$$

$$\frac{\%e^b}{\%e^b + \%e^a} \quad (\% \text{ o19})$$

$$-\frac{\%e^{b+a}}{\left(\%e^b + \%e^a \right)^2} \quad (\% \text{ o20})$$

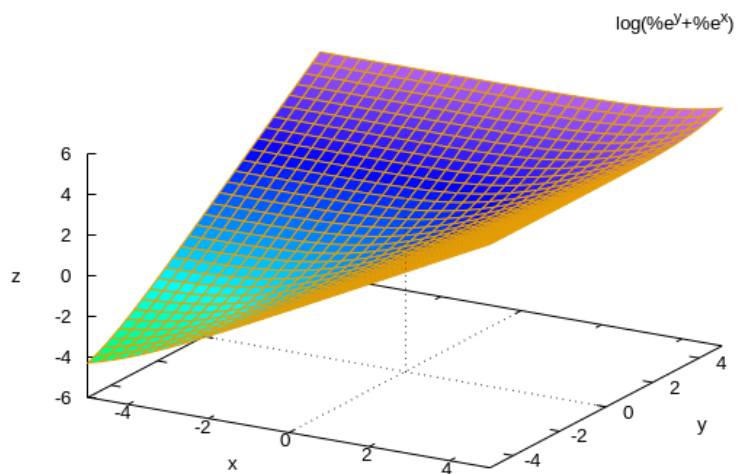
Let's have a look:

```
(% i21) wxplot2d ([down(x,x),down(x,x+3),down(x,2*x),down(x,2*x+3),down(x,3*x),down(x,3*x+3)],
[x, -2, 2])$
```



(% t21)

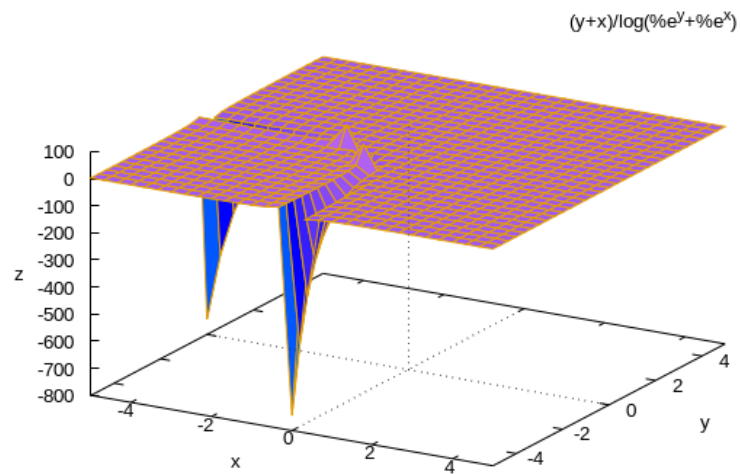
```
(% i22) wxplot3d (down(x,y), [x, -5, 5], [y, -5, 5])$
```



(% t22)

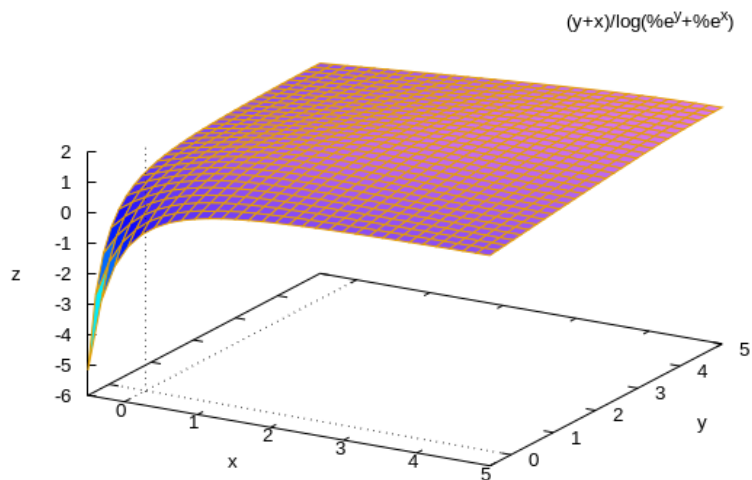
Comparing $x+y$ with $\text{down}(x,y)$:

(% i23) wxplot3d ((x+y)/down(x,y), [x, -5, 5], [y, -5, 5])\$



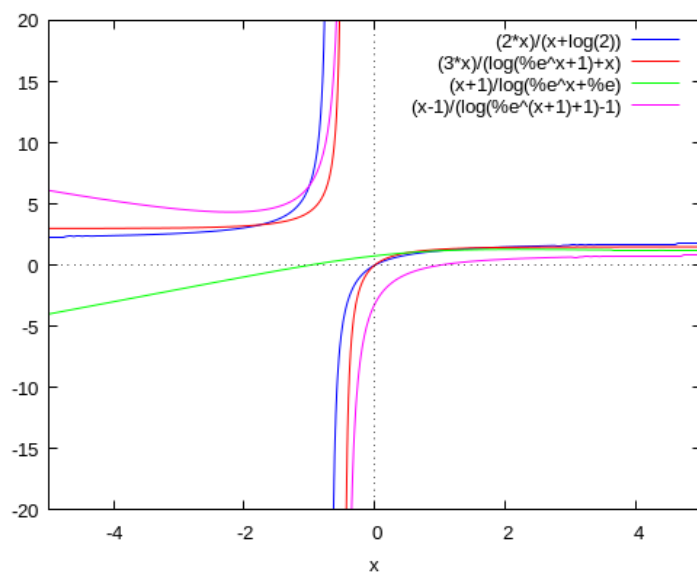
(% t23)

(% i24) wxplot3d ((x+y)/down(x,y), [x, -0.5, 5], [y, -0.5, 5])\$



(% t24)

(% i25) wxplot2d ([2*x/down(x,x), 3*x/down(x,2*x),(x+1)/down(x,1),(x-1)/down(x,-1)], [x, -5, 5], [y, -20, 20])\$



(% t25)

1.2 At -,*,/

We can also consider a "down-like" operation with a minus instead a plus:

(% i26) `down_min(a,b) := radcan(log(exp(a) - exp(b)));`

$$\text{down_min}(a, b) := \text{radcan}(\log(\exp(a) - \exp(b))) \quad (\% \text{ o26})$$

Associativity:

(% i30) `down(down_min(a,b),c);`
`down_min(down(a,c),b);`
`down(down_min(a,b),down_min(c,d));`
`down(down_min(a,d),down_min(c,b));`

$$\log\left(\%e^c - \%e^b + \%e^a\right) \quad (\% \text{ o27})$$

$$\log\left(\%e^c - \%e^b + \%e^a\right) \quad (\% \text{ o28})$$

$$\log\left(-\%e^d + \%e^c - \%e^b + \%e^a\right) \quad (\% \text{ o29})$$

$$\log\left(-\%e^d + \%e^c - \%e^b + \%e^a\right) \quad (\% \text{ o30})$$

Same distributive property:

(% i31) `down_min(a+b,a+c);`

$$\log\left(\%e^b - \%e^c\right) + a \quad (\% \text{ o31})$$

As we've already seen, with times, we achieve addition. With divide, one achieves subtraction:

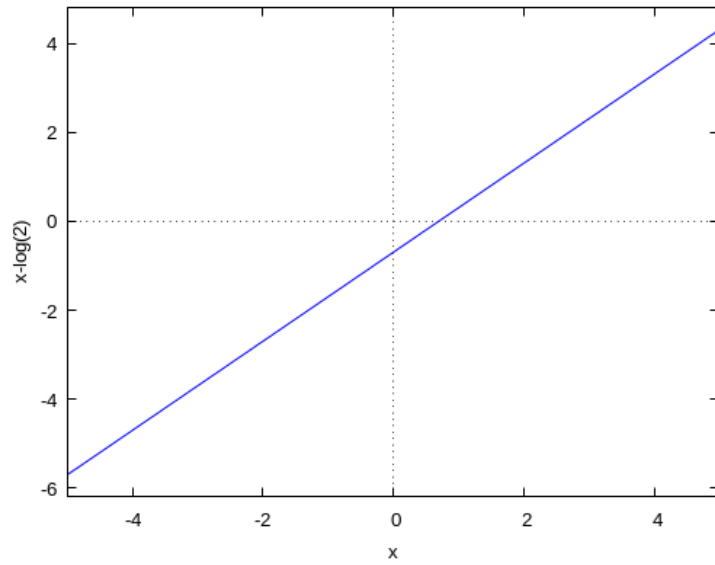
(% i33) `down_div(a,b) := radcan(log(exp(a) / exp(b)));`
`down_div(a,b);`

$$\text{down_div}(a, b) := \text{radcan}\left(\log\left(\frac{\exp(a)}{\exp(b)}\right)\right) \quad (\% \text{ o32})$$

$$a - b \quad (\% \text{ o33})$$

Using @_/ or down_div for the "ratio" of + and @...

```
(% i35) down_div(x+x,down(x,x));
wxplot2d (2*x - down(x,x), [x, -5, 5])$
x - log(2)                                     (% o34)
```



(% t35)

1.3 Down differentiation

Given down, down_min, and down_div, one is free to define a "down derivative". However, we must take care with the notion of infinitesimals: normally $h \rightarrow 0$ in the definition of the derivative, but here we need $h \rightarrow -\infty$ to approach the @_+ identity ($x @ -\infty = x$)...

```
(% i36) Df_Dx_eq : Df_Dx = limit( down_min( f(down(x,h)) , f(x) ) - h , h, -infinity);
```

$$Df_Dx = \lim_{h \searrow -\infty} \log \left(\frac{e^{f(\log(e^x + e^h))}}{e^{f(x)}} - h \right) \quad (Df_Dx_eq)$$

```
(% i40) Df(x,h) := down_min( f(down(x,h)) , f(x) );
Df(x,h);
Df_Dx(x) := ratsimp( limit( Df(x,h) - h , h, minf) );
Df_Dx(x);
```

```
Df(x,h) := down_min( f(down(x,h)) , f(x))                                     (% o37)
```


$$\log \left(e^{f(\log(e^x + e^h))} - e^{f(x)} \right) \quad (38)$$

$$\text{Df_Dx}(x) := \text{ratsimp}(\text{limit}(\text{Df}(x, h) - h, h, -\infty)) \quad (\% \text{ o39})$$

$$\lim_{h \searrow -\infty} \log \left(e^{f(\log(e^x + e^h))} - e^{f(x)} \right) = h \quad (40)$$

```
(% i43) f(x) := A*x^ b + c;
      Df(x,h);
      Df_Dx(x) ;

      f(x) := Ax^b + c                                     (% o41)
```

$$\log \left(\% e^{A \log (\% e^x + \% e^h)^b} - \% e^{A x^b} \right) + c \quad (\% 042)$$

$$(b-1)\log(x) + Ax^b - x + c + \log(b) + \log(A) \quad (\% \text{ o43})$$

```
(% i46) f(x) := log(A*x^ b + c);
      Df(x,h);
      Df_Dx(x) ;

      f(x) := log (A x^ b + c)                                (% o44)
```

$$\log \left(\log \left(\varphi e^x + \varphi e^h \right)^b - x^b \right) + \log(A) \quad (\varphi \text{ o45})$$

$$(b-1)\log(x) - x + \log(b) + \log(A) \tag{046}$$

```
(% i49) f(x) := exp(A*x^ b + c);
        Df(x,h);
        Df_Dx(x) ;

f(x) := exp (Ax^b + c)                                     (% o47)
```

$$\log \left(e^{e^{A \log(e^x + e^h)^{b+c}}} - e^{e^{Ax^b+c}} \right) \quad (\% \text{ o48})$$

$$(b-1) \log(x) + e^{Ax^b+c} + Ax^b - x + c + \log(b) + \log(A) \quad (\% \text{ o49})$$

(% i52) f(x) := down(g(x) , p(x));
 Df(x,h);
 Df_ Dx(x) ;
 f(x) := down(g(x) , p(x)) \quad (\% \text{ o50})

$$\log \left(e^{p(\log(e^x + e^h))} + e^{g(\log(e^x + e^h))} - e^{p(x)} - e^{g(x)} \right) \quad (\% \text{ o51})$$

$$\lim_{h \searrow -\infty} \log \left(e^{p(\log(e^x + e^h))} + e^{g(\log(e^x + e^h))} - e^{p(x)} - e^{g(x)} \right) - h \quad (\% \text{ o52})$$

So the down-derivative maintains not only the extended distributivity, but also the property that the exponential function keeps its special role: $D_Dx(e^{\hat{x}}) = e^{\hat{x}}$.

(% i55) f(x) := b^ (A*x+c);
 Df(x,h);
 Df_ Dx(x) ;
 f(x) := b^{Ax+c} \quad (\% \text{ o53})

$$\log \left(e^{b^{A \log(e^x + e^h)^{b+c}}} - e^{b^{Ax+c}} \right) \quad (\% \text{ o54})$$

$$(A \log(b) - 1) x + \log(b)c + b^{Ax+c} + \log(\log(b)) + \log(A) \quad (\% \text{ o55})$$

(% i58) f(x) := log(x);
 Df(x,h);
 Df_ Dx(x) ;
 f(x) := log(x) \quad (\% \text{ o56})

$$\log \left(\log \left(e^x + e^h \right) - x \right) \quad (\% \text{ o57})$$

$$-x \quad (\% \text{ o58})$$

(% i59) kill(f);

done (% o59)

(% i60) Eig_Df_eq : Df_Dx(x) = f(x);

$$\lim_{h \rightarrow -\infty} \log \left(e^{f(\log(e^x + e^h))} - e^{f(x)} \right) - h = f(x) \quad (\text{Eig_Df_eq})$$

As we can see from the previous solutions of Df_Dx, this DE is uniquely solved by f(x)=x.

(% i61) kill(f);

done (% o61)

Alternative expressions for Df_Dx(x) (with x=log(z)):

(% i65) x = log(z);
log(diff(exp(f(log(z))) ,z,1));
log(diff(exp(f(log(z))) ,z,1));
f(x) - x + log(diff(f(x),x,1));

$$x = \log(z) \quad (\% \text{ o62})$$

$$\log \left(\frac{d}{dz} e^{f(\log(z))} \right) \quad (\% \text{ o63})$$

$$\log \left(e^{f(\log(z))} \left(\frac{d}{dz} f(\log(z)) \right) \right) \quad (\% \text{ o64})$$

$$\log \left(\frac{d}{dx} f(x) \right) + f(x) - x \quad (\% \text{ o65})$$

2 Down2

Heading further "down", define "three dots" (therefore), or down2, such that: a
 @ down2(b,c) = down2(a@b,a@c) ...Or: down(a,down2(b,c)) = down2(down(a,b),down(a,c))
 ...

(% i66) td_eq : Td(a,b) = log(down(exp(a),exp(b)));

$$\text{Td}(a, b) = \log \left(\log \left({}_0e^{{}_0e^b} + {}_0e^{{}_0e^a} \right) \right) \quad (\text{td_eq})$$

(% i67) down2(a,b) := radcan(log(down(exp(a),exp(b))));

$$\text{down2}(a, b) := \text{radcan} \left(\log \left(\text{down} \left(\exp(a), \exp(b) \right) \right) \right) \quad (\% \text{ o67})$$

(% i68) down2(down(a,b),down(a,c));

$$\log \left(\log \left({}_0e^{{}_0e^c} + {}_0e^{{}_0e^b} \right) + {}_0e^a \right) \quad (\% \text{ o68})$$

yes!down2-identity would have to be at $\infty + i\pi$.

(% i69) down2(x,x);

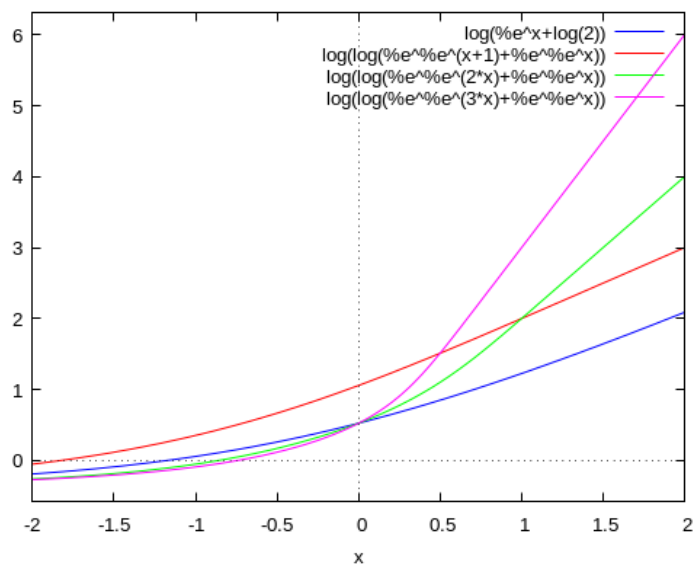
$$\log({}_0e^x + \log(2)) \quad (\% \text{ o69})$$

(% i70) diff(down2(a,b),a);

$$\frac{{}_0e^{{}_0e^a + a}}{\left({}_0e^{{}_0e^b} + {}_0e^{{}_0e^a} \right) \log \left({}_0e^{{}_0e^b} + {}_0e^{{}_0e^a} \right)} \quad (\% \text{ o70})$$

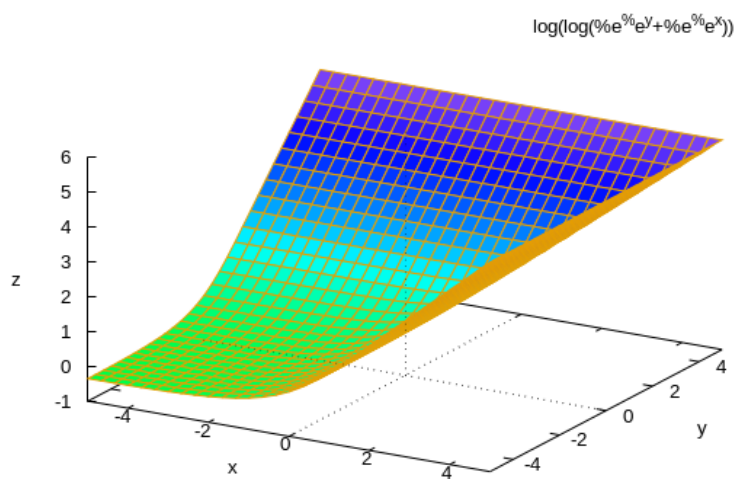
Let's take a look...

(% i71) wxplot2d ([down2(x,x),down2(x,x+1),down2(x,2*x),down2(x,3*x)], [x, -2, 2])\$



(% t71)

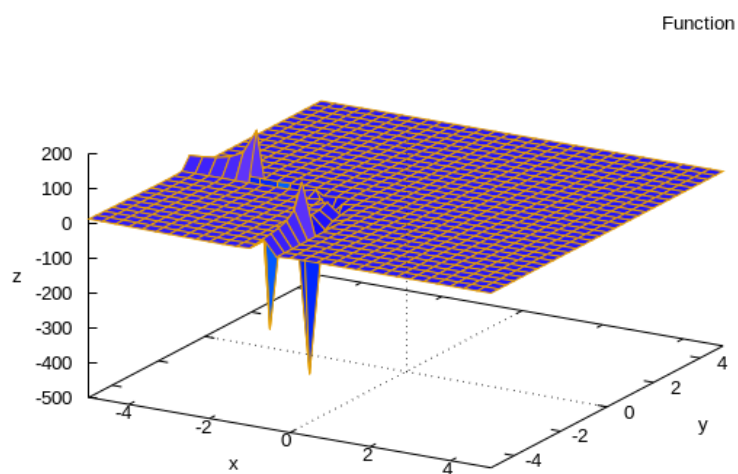
(% i72) wxplot3d (down2(x,y), [x, -5, 5], [y, -5, 5])\$



(% t72)

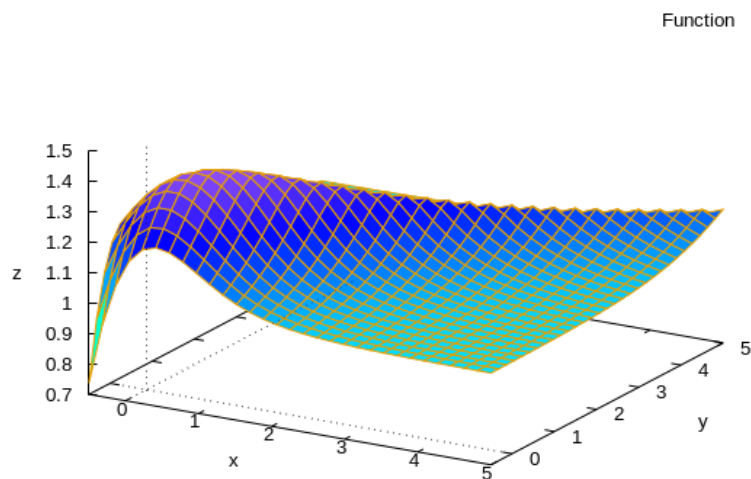
Comparing $\text{down}(x,y)$ with $\text{down2}(x,y)$:

(% i73) wxplot3d (down(x,y)/down2(x,y), [x, -5, 5], [y, -5, 5])\$



(% t73)

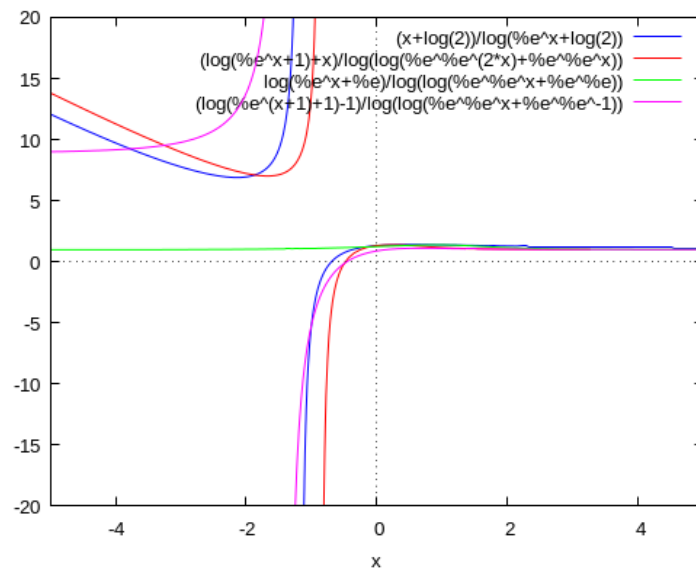
(% i74) wxplot3d (down(x,y)/down2(x,y), [x, -0.5, 5], [y, -0.5, 5])\$



(% t74)

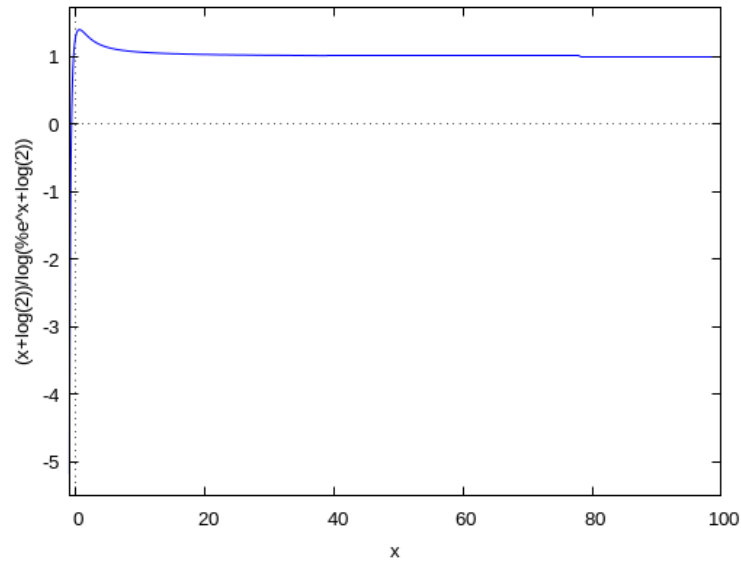
(% i75) wxplot2d ([down(x,x)/down2(x,x), down(x,2*x)/down2(x,2*x),down(x,1)/down2(x,1),down(x,-1)/down2(x,-1)], [x, -5, 5], [y, -20, 20])\$

plot2d : expression evaluate on non-numeric values somewhere in plotting range.



(% t75)

```
(% i76) wxplot2d (down(x,x)/down2(x,x), [x, -1, 100])$
```



(% t76)

3 Up

Let's take a look in the "up" direction for an operation which is distributive over multiplication. Defining "star" (or "up"): $a * (b \cdot c) = (a * b) \cdot (a * c)$... Or: $\text{up}(a, b \cdot c) = \text{up}(a, b) \cdot \text{up}(a, c)$...

```
(% i77) str_eq : Str(a,b) = exp(log(a)*log(b));
```

$$\text{Str}(a, b) = e^{\log(a) \log(b)} \quad (\text{str_eq})$$

```
(% i78) up(a,b) := radcan( exp(log(a)*log(b)) );
```

$$\text{up}(a, b) := \text{radcan}(\exp(\log(a) \log(b))) \quad (\% \text{ o78})$$

```
(% i79) up(a,b);
```

$$a^{\log(b)} \quad (\% \text{ o79})$$

```
(% i80) up(a,b*c);
```

$$a^{\log(c) + \log(b)} \quad (\% \text{ o80})$$


```
(% i81) up(a,b) * up(a,c);
```

$$a^{\log(c) + \log(b)} \quad (\% \text{ o81})$$

which is the same as the former. So, yes! Check with "down"-like operation:

```
(% i82) log( up(exp(a),exp(b)) );
```

$$ab \quad (\% \text{ o82})$$

yes! up(a,b) could also be expressed with b as the base:

```
(% i86) is( up(a,b) = b ^ log(a) );
is( up(a,b) = a ^ log(b) );
is( up(b,a) = a ^ log(b) );
is( up(a,b) = up(b,a) );
```

$$\text{false} \quad (\% \text{ o83})$$

$$\text{true} \quad (\% \text{ o84})$$

$$\text{true} \quad (\% \text{ o85})$$

$$\text{true} \quad (\% \text{ o86})$$

... although it seems Maxima has trouble (it enforces some type of alphabetization?) with that realization in the present form...? So one advantage of up(a,b), over "naive" exponentiation (a^b or b^a), can be seen immediately: namely commutativity... up(a,b)=up(b,a). Identity for up (or up_times):

```
(% i87) up(e,a);
```

$$a \quad (\% \text{ o87})$$

... the base e. Some special values:

```
(% i89) up(x,x);
x ^ log(x);
```

$$e^{\log(x)^2} \quad (\% \text{ o88})$$

$$x^{\log(x)} \quad (\% \text{ o89})$$

```
(% i92) diff(up(a,b),a);
        diff(up(a,b),b);
        diff(diff(up(a,b),a),b);
```

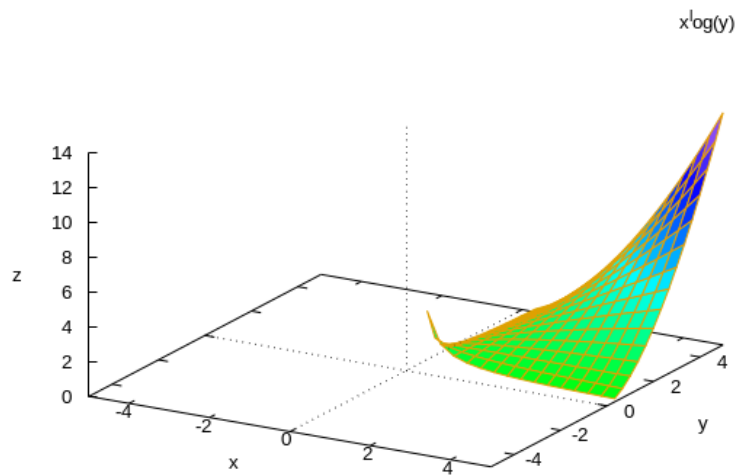
$$a^{\log(b)-1} \log(b) \quad (\% \text{ o90})$$

$$\frac{a^{\log(b)} \log(a)}{b} \quad (\% \text{ o91})$$

$$\frac{a^{\log(b)-1} \log(a) \log(b)}{b} + \frac{a^{\log(b)-1}}{b} \quad (\% \text{ o92})$$

Let's take a look:

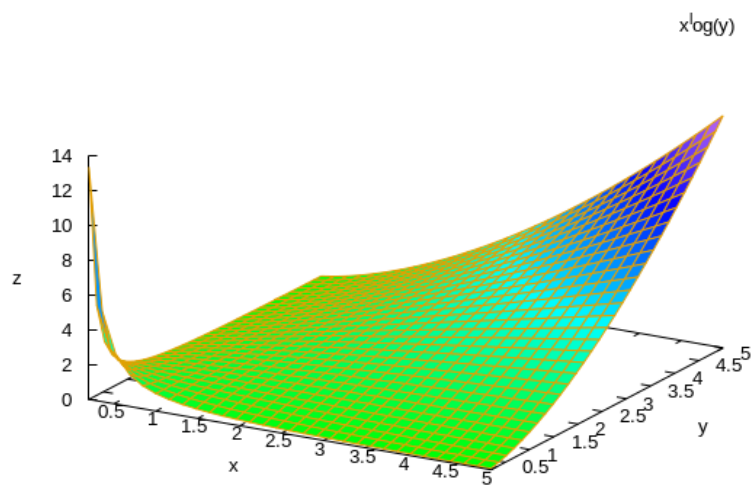
```
(% i93) wxplot3d (up(x,y), [x, -5, 5], [y, -5, 5])$
```



(% t93)

The singularity at the origin comes as no surprise. And for finite, real values, x and y should remain > 0 .

```
(% i94) wxplot3d (up(x,y), [x, 0.2, 5], [y, 0.2, 5])$
```



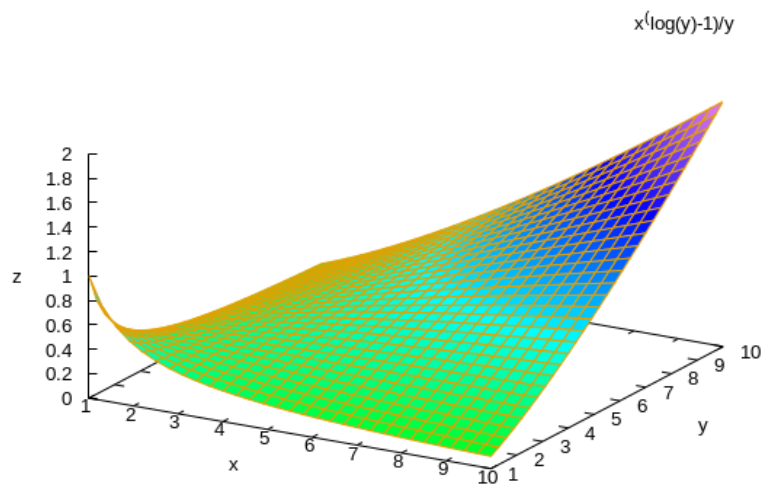
(% t94)

Compare with $x.y$:

```
(% i96) up(x,y)/(x*y);  
wxplot3d (up(x,y)/(x*y), [x, 1, 10], [y, 1, 10])$
```

$$\frac{x^{\log(y)-1}}{y}$$

(% o95)



(% t96)

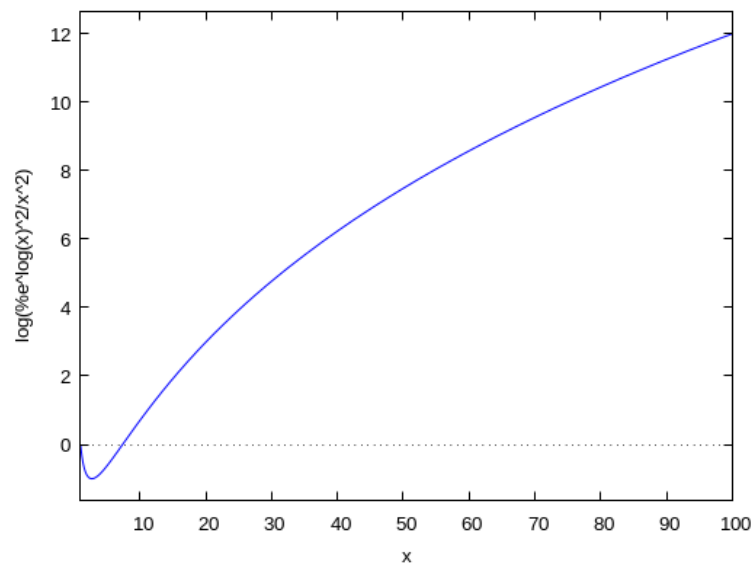
```
(% i99) up(x,x)/(x*x);
        x^(log(x)-2);
        wxplot2d (log(up(x,x)/(x*x)), [x, 1, 100])$
```

$$\frac{e^{\log(x)^2}}{x^2}$$

(% o97)

$$x^{\log(x)-2}$$

(% o98)



(% t99)

So $\text{up}(x,x)$ does eventually win over $x^2 \dots$ at $x=e^2$, where...

```
(%      up(exp(2),exp(2));
i101)  float(%), numer;
```

e^4

(% o100)

54.59815003314423

(% o101)

Using up_div for the "ratio"...

```
(%      exp(log(up(x,x))/log(x*x));
i103)  radcan(%);
```

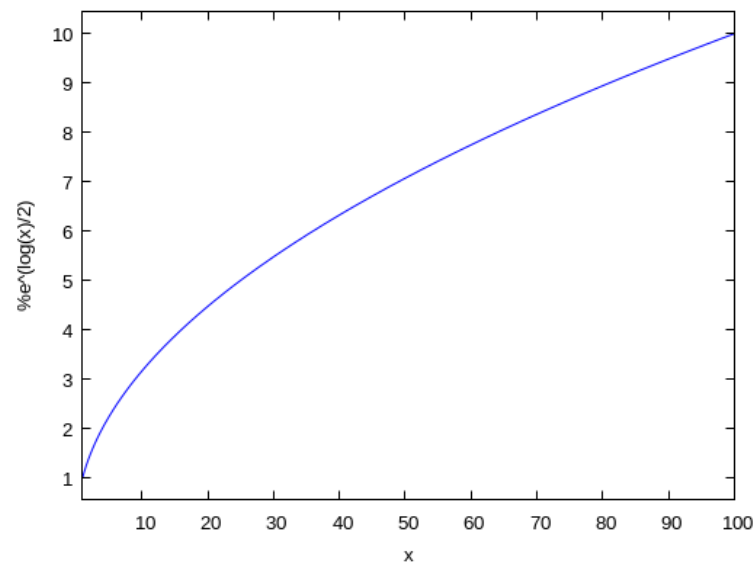
$e^{\frac{\log(x)}{2}}$

(% o102)

\sqrt{x}

(% o103)

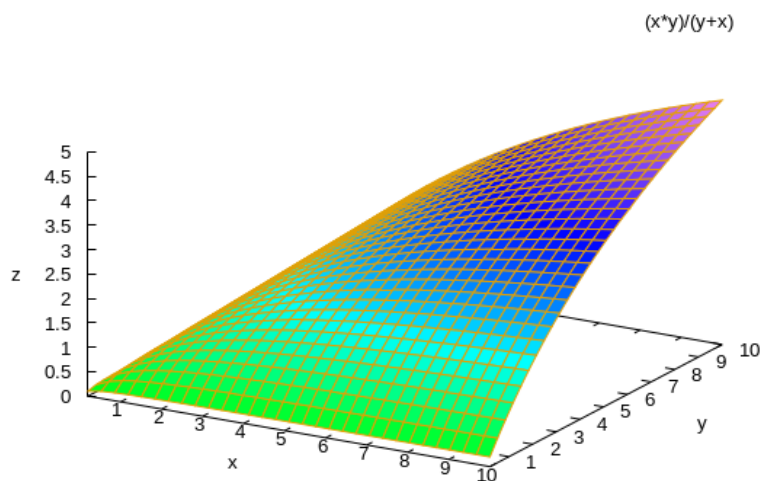
```
(% wxplot2d (exp(log(up(x,x))/log(x*x)), [x, 1, 100])$
i104)
```



(% t104)

And x,y with x+y :

```
(% wxplot3d (x*y/(x+y), [x, 0.2, 10], [y, 0.2, 10])$
i105)
```



(% t105)

3.1 Up differentiation

To avoid problems with alphabetical preferences...

```
(% up(x,h);
i107) up(a,h);
```

$h^{\log(x)}$ (% o106)

$a^{\log(h)}$ (% o107)

we'll use 'a' as the independent variable here. Working with up_/, up_+ as times, up_- as divide, and the new "up_+"-identity, h->1...

```
(% Uf(a,h) := f(a*h) / f(a) ;
i111) Uf(a,h);
Uf_Ua(a) := radcan( limit( exp( log(Uf(a,h)) / log(h) ) , h, 1, plus) );
Uf_Ua(a);
```

$Uf(a, h) := \frac{f(ah)}{f(a)}$ (% o108)

$$\frac{f(ah)}{f(a)} \quad (\% \text{ o109})$$

$$\text{Uf_Ua}(a) := \text{radcan} \left(\text{limit} \left(\exp \left(\frac{\log(\text{Uf}(a, h))}{\log(h)} \right), h, 1, \text{plus} \right) \right) \quad (\% \text{ o110})$$

$$\lim_{\rightarrow h \setminus \setminus > 1+} \frac{f(ah)^{\frac{1}{\log(h)}}}{f(a)^{\frac{1}{\log(h)}}} \quad (\% \text{ o111})$$

```
(%      Uf_Ua(a) := ratsimp( limit( Uf(a,h)^ (1 / log(h)) , h, 1, plus));
i113)   Uf_Ua(a);
```

$$\text{Uf_Ua}(a) := \text{ratsimp} \left(\text{limit} \left(\text{Uf}(a, h)^{\frac{1}{\log(h)}}, h, 1, \text{plus} \right) \right) \quad (\% \text{ o112})$$

$$\lim_{\rightarrow h \setminus \setminus > 1+} \left(\frac{f(ah)}{f(a)} \right)^{\frac{1}{\log(h)}} \quad (\% \text{ o113})$$

```
(%      f(a) := n*a + c;
i116)   Uf(a,h);
        Uf_Ua(a);
```

$$f(a) := na + c \quad (\% \text{ o114})$$

$$\frac{ahn + c}{an + c} \quad (\% \text{ o115})$$

$${}_0e^{\frac{an}{an+c}} \quad (\% \text{ o116})$$

```
(%      assume(y>1);
i120)   f(a) := a^ y + c;
        Uf(a,h);
        Uf_Ua(a);
```

$$[y > 1] \quad (\% \text{ o117})$$

$$f(a) := a^y + c \tag*{(\% o118)}$$

$$\frac{(ah)^y + c}{c + a^y} \tag*{(\% o119)}$$

$$\%e^{\frac{a^y y}{c+a^y}} \tag*{(\% o120)}$$

```
(%      f(a) := exp(b*a^ y + c);
i123)   Uf(a,h);
        Uf_ Ua(a);

f(a) := exp(ba^y + c) \tag*{(\% o121)}
```

$$\%e^{b(ah)^y - a^y b} \tag*{(\% o122)}$$

"Is "

y

" an "

integer

"?"

no ;

$$\%e^{a^y by} \tag*{(\% o123)}$$

```
(%      f(a) := b^ (a^ y + c);
i126)   Uf(a,h);
        Uf_ Ua(a);

f(a) := b^{a^y+c} \tag*{(\% o124)}
```

$$b^{(ah)^y - a^y} \tag*{(\% o125)}$$

"Is "

$$y$$

" an "

integer

"?"

no;

"Is "

$$y - 1$$

" positive, negative or zero?"

positive;

$$\%e^{a^y \log(b)y} \tag*{(\% o126)}$$

(% f(a) := exp(b^ (a^ y + c));
i129) Uf(a,h);
Uf_ Ua(a);

$$f(a) := \exp\left(b^{a^y + c}\right) \tag*{(\% o127)}$$

$$\%e^{b^{(ah)^y + c} - b^{c + a^y}} \tag*{(\% o128)}$$

"Is "

$$c$$

" an "

integer

"?"

no;

$$\%e^{a^yb^{c+a^y}\log(b)y} \quad (\% \text{ o129})$$

(% f(a) := g(a) * p(a);
i132) Uf(a,h);
 Uf_ Ua(a);

$$f(a) := g(a) p(a) \quad (\% \text{ o130})$$

$$\frac{g(ah) p(ah)}{g(a) p(a)} \quad (\% \text{ o131})$$

$$\lim_{\rightarrow h \setminus \setminus > 1+} \left(\frac{g(ah) p(ah)}{g(a) p(a)} \right)^{\frac{1}{\log(h)}} \quad (\% \text{ o132})$$

So the up-derivative maintains not only the extended distributivity, but also the property that the exponential function keeps its special role: $U_Ua(\mathfrak{e}\hat{a}) = \mathfrak{e}\hat{a}$.

(% f(a) := b^ exp(a^ y + c);
i135) Uf(a,h);
 Uf_ Ua(a);

$$f(a) := b^{\exp(a^y+c)} \quad (\% \text{ o133})$$

$$b^{\%e^{(ah)^y+c} - \%e^{c+a^y}} \quad (\% \text{ o134})$$

"Is "

c

" an "

integer

"?"

no;

$$\%e^{a^y \log(b)y} \%e^{c+a^y} \quad (\% \text{ o135})$$

```
(%      f(a) := log(a);
i138)   Uf(a,h);
        Uf_ Ua(a);

f(a) := log(a) \quad (\% \text{ o136})
```

$$\frac{\log(ah)}{\log(a)} \quad (\% \text{ o137})$$

$$\%e^{\frac{1}{\log(a)}} \quad (\% \text{ o138})$$

```
(%      kill(f);
i139)

done \quad (\% \text{ o139})
```

Alternative expressions for Uf_ Ua(a) (with a=exp(z)):

```
(%      a = exp(z);
i144)   exp('diff( log(f(exp(z))) ,z,1));
        exp(diff( log(f(exp(z))) ,z,1));
        exp( a * diff(f(a),a,1) / f(a) );
        exp('diff( log(f(a)) ,log(a),1));

a = \%e^z \quad (\% \text{ o140})
```

$$\%e^{\frac{d}{dz} \log(f(\%e^z))} \quad (\% \text{ o141})$$

$$\%e^{\frac{\frac{d}{da} f(\%e^z)}{f(\%e^z)}} \quad (\% \text{ o142})$$

$$\%e^{\frac{a \left(\frac{d}{da} f(a) \right)}{f(a)}} \quad (\% \text{ o143})$$

$$\%e^{\frac{d}{d \log(a)} \log(f(a))} \quad (\% \text{ o144})$$

This leads to an alternative expression for the "normal" derivative, d/dx , in terms of the down derivative:

```
(%      exp(x + 'Df_Dx(x) - f(x));
i148)   z = log(x); g(x) = log(f(x));
        exp('Dg_Dz(x));
```

$$\%e^{Df_Dx(x)-f(x)+x} \quad (\% \text{ o145})$$

$$z = \log(x) \quad (\% \text{ o146})$$

$$g(x) = \log(f(x)) \quad (\% \text{ o147})$$

$$\%e^{Dg_Dz(x)} \quad (\% \text{ o148})$$

4 Up2

Now to be distributive over "up": $\text{up2}(a, \text{up}(b,c)) = \text{up}(\text{up2}(a,b), \text{up2}(a,c)) \dots$

```
(%      up2(a,b) := radcan( exp( up(log(a),log(b)) ) );
i149)
```

$$\text{up2}(a, b) := \text{radcan}(\exp(\text{up}(\log(a), \log(b)))) \quad (\% \text{ o149})$$

```
(%      up(up2(a,b),up2(a,c));
i151)   up2(a,up(b,c));
```

$$\%e^{\log(a)^{\log(\log(c))+\log(\log(b))}} \quad (\% \text{ o150})$$

$$\%e^{\log(a)^{\log(\log(c))+\log(\log(b))}} \quad (\% \text{ o151})$$

Bingo! And obviously also commutative, unlike say, tetration ($\hat{y}(x)$ or $\hat{x}(y)$), which is neither. Identity for up2 (or up2_times):

```
(%      up2(e^ e,a);
i152)

a                                     (% o152)
```

... or base to the power of base. So one can imagine that for up3 it would be $e^{(e^e)}$

```
(%      diff(up2(a,b),a);
i153)

log(a)^(log(log(b))-1)e^(log(a)^(log(log(b))))log(log(b))
a                                     (% o153)
```

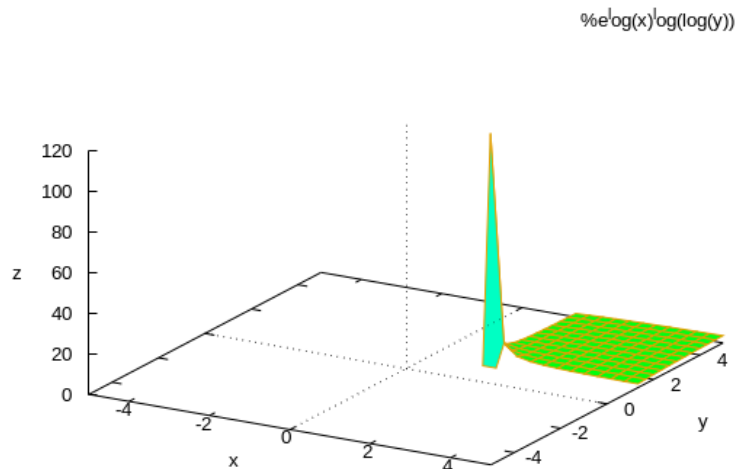
```
(%      up2(x,x);
i155)      exp(log(x)^ log(x));

e^(e^(log(log(x))^2))                (% o154)
```

```
e^(log(x)^(log(x)))                  (% o155)
```

Taking a look...

```
(%      wxplot3d (up2(x,y), [x, -5, 5], [y, -5, 5])$
i156)
```

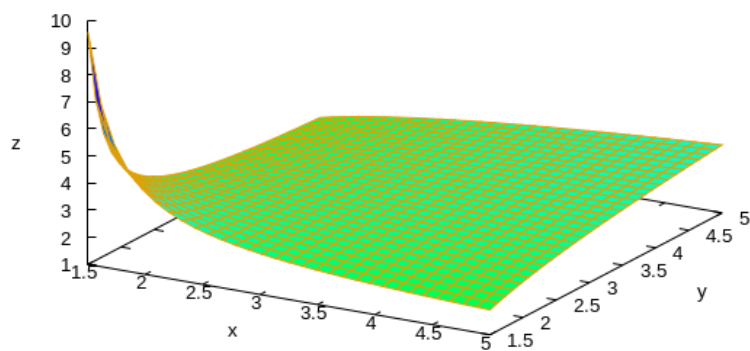


(% t156)

Problems now at x or $y = 1$.

(% wxplot3d (up2(x,y), [x, 1.5, 5], [y, 1.5, 5])\$
i157)

%e^{log(x)}log(log(y))

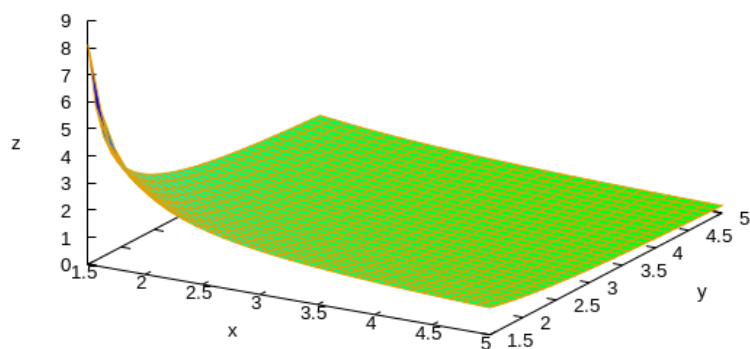


(% t157)

Comparing $up2(x,y)$ with $up(x,y)$:

```
(% wxplot3d (up2(x,y)/up(x,y), [x, 1.5, 5], [y, 1.5, 5])$
i158)
```

```
%elog(x)log(log(y))/xlog(y)
```



```
(% t158)
```

```
(% up2(x,x)/up(x,x);
i162) log(up2(x,x)/up(x,x));
log(x)^log(log(x))-log(x)^2;
wxplot2d (log(up2(x,x)/up(x,x)), [x, 1.5, 5000])$
```

```
%e%elog(log(x))2-log(x)2
```

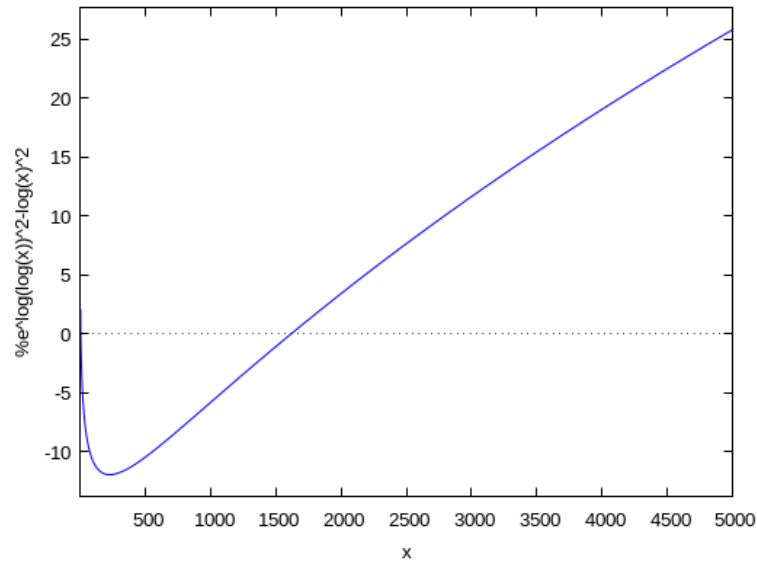
(% o159)

```
%elog(log(x))2-log(x)2
```

(% o160)

```
log(x)log(log(x))-log(x)2
```

(% o161)



(% t162)

So up2(x,x) does eventually win over up(x,x)... at $x=e(\hat{e^2})$, where...

```
(%      up2(exp(exp(2)),exp(exp(2)));
i164)  float(%), numer;
```

$\%e^{\%e^4}$ (% o163)

5.14843556263455710²³ (% o164)

Holy Avogadro, batman! Using up2_div for the "ratio"...

```
(%      up_div(a,b) := radcan( exp(log(a) / log(b)) );
i166)  up2_div(a,b) := radcan( exp( up_div(log(a),log(b)) ) );
```

$\text{up_div}(a,b) := \text{radcan}\left(\exp\left(\frac{\log(a)}{\log(b)}\right)\right)$ (% o165)

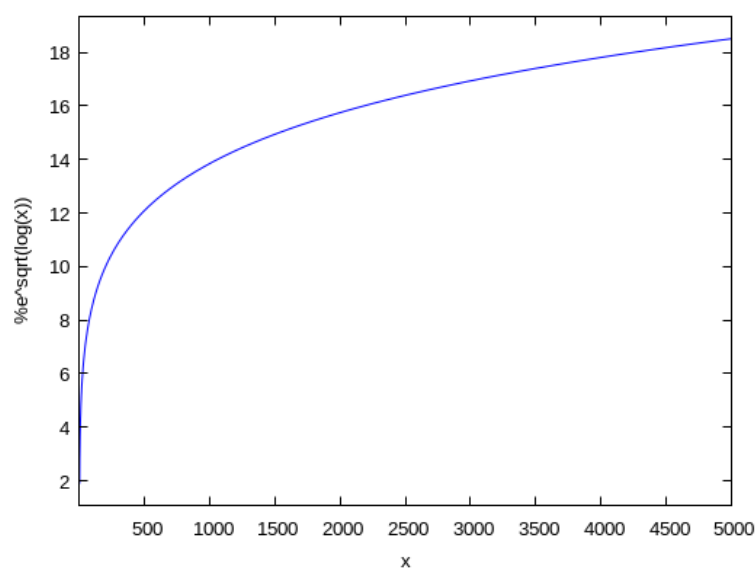
$\text{up2_div}(a,b) := \text{radcan}\left(\exp(\text{up_div}(\log(a),\log(b)))\right)$ (% o166)

```
(%      up2_div(up2(x,x),up(x,x));
i167)
```

$$e^{\sqrt{\log(x)}}$$

```
(% o167)
```

```
(%      wxplot2d (up2_div(up2(x,x),up(x,x)), [x, 1.5, 5000])$
i168)
```

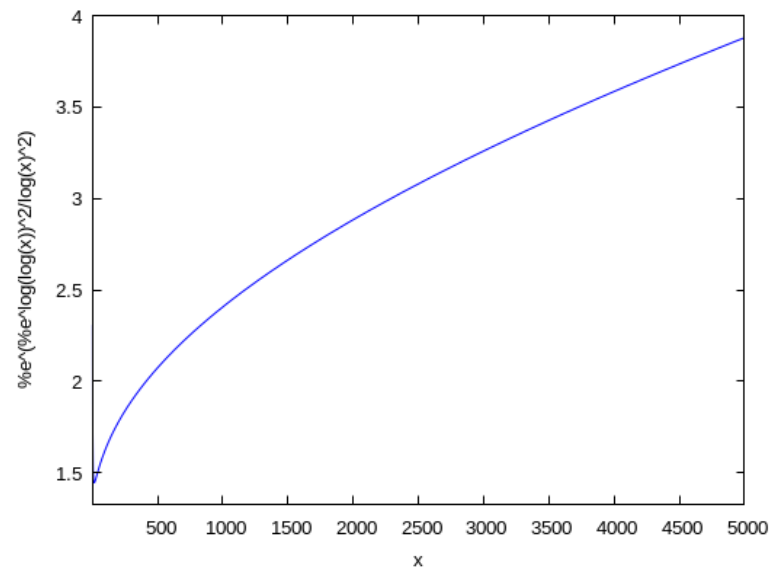


```
(% t168)
```

```
(%      up_div(up2(x,x),up(x,x));
i170)      wxplot2d (up_div(up2(x,x),up(x,x)), [x, 3, 5000])$
```

$$e^{\frac{\log(\log(x))^2}{\log(x)^2}}$$

```
(% o169)
```



(% t170)