Part 1

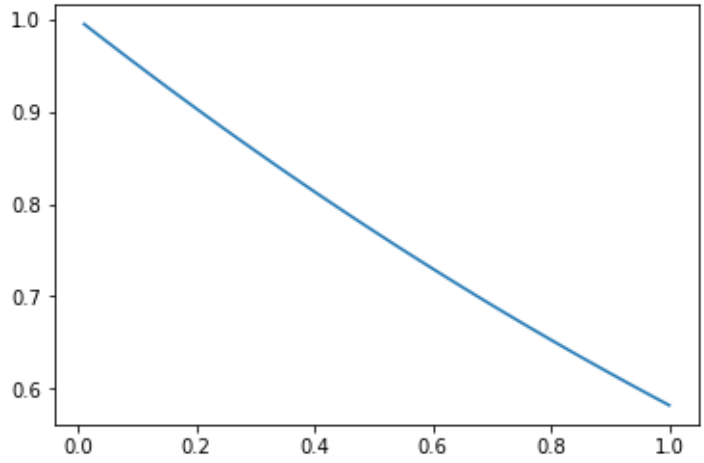
Problem 2

(b) from left to right, the sum is 244.71. If k>=17, then the sum no longer changes. The relative error is 7.3839*10^(-5)
(c) from right to left, the sum is 244.70. If k>=17, then the sum no longer changes. The relative error is 3.2971*10^(-5)
(d) from (i) to (iv), the values are 0.0037353, 0.004, 0, 0.01, k = 24, 19, 19, 19 respectively. The relative errors are 0.086002, 0.021232, 1, 1.4469
Thus, we can see adding from right to left is always the most accurate, because add small numbers first leads to higher accuracy. And we should avoid add positive and negative part respectively. Adding from right to left also converges more quickly.
(e) compute e^(-5.5) by first compute e^5.5 the get its inverse 1/e^(5.5). The value is 0.0040866, the relative errors is 4.19496*10^(-5).

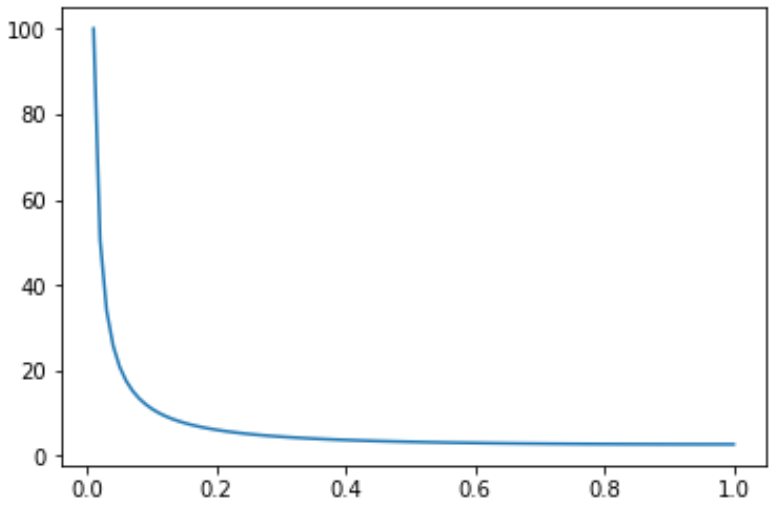
Problem 4

(c)

(condf)(x) = x/(e^x-1)



(condA)(x) = e^x/x



Problem 5

n_stop = 10*13, and the values are

2
2.5937424601
2.704813829422
2.716923932236
2.718145926825
2.718268237192
2.718280469096
2.718281694132
2.718281798347
2.718282052012
2.718282053235
2.718282053357
2.718523496037
2.716110034087 (n_stop)
2.716110034087

The final value it converges is 1. This is because when n is greater than 2^53, that is about 10**16, 1/n becomes 0 in the machine. Thus (1+1/n) is 1 and (1+1/n)^n = 1.

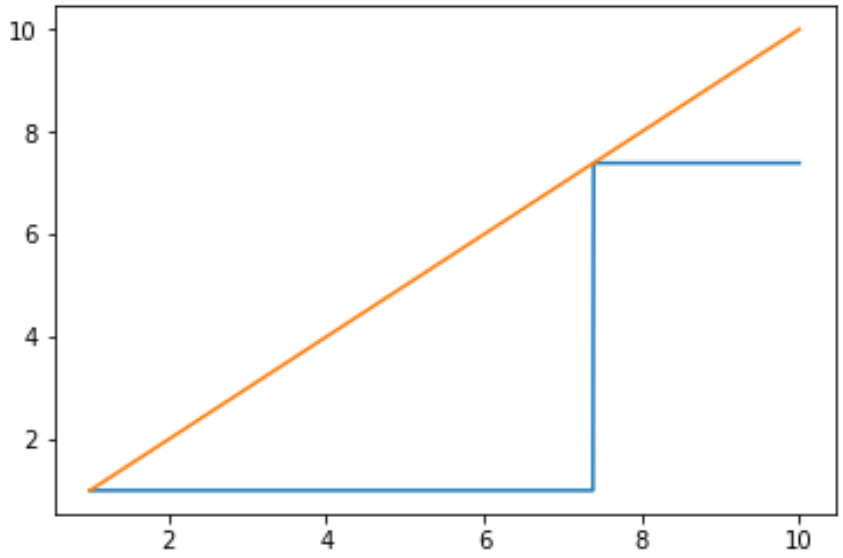
The reason it seemingly converges to 2.716110034087 is

1. The relative error of 1/n is about 10*13/10*16 = 1/1000 (that is e_x), after adding and exponentiating, the relative error is about $(\frac{n}{n+1}e_x + ne_1)$, which is in the order of 10^(-3), thus only the first 2 or 3 digit is accurate

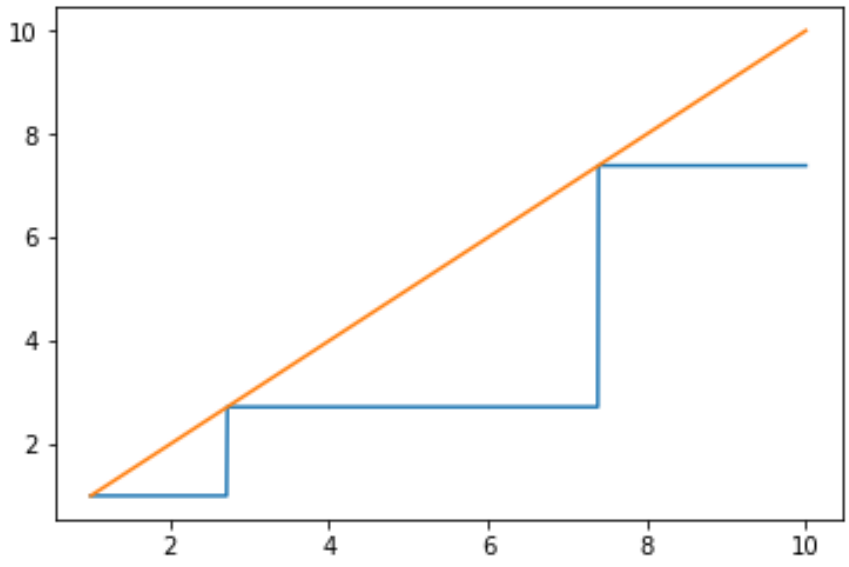
2. Coincidently, 1+1/10^13 and (1+1/10^14)^10 has almost the same binary form, while this is not true for (1+1/10^15)^100.

Problem 6

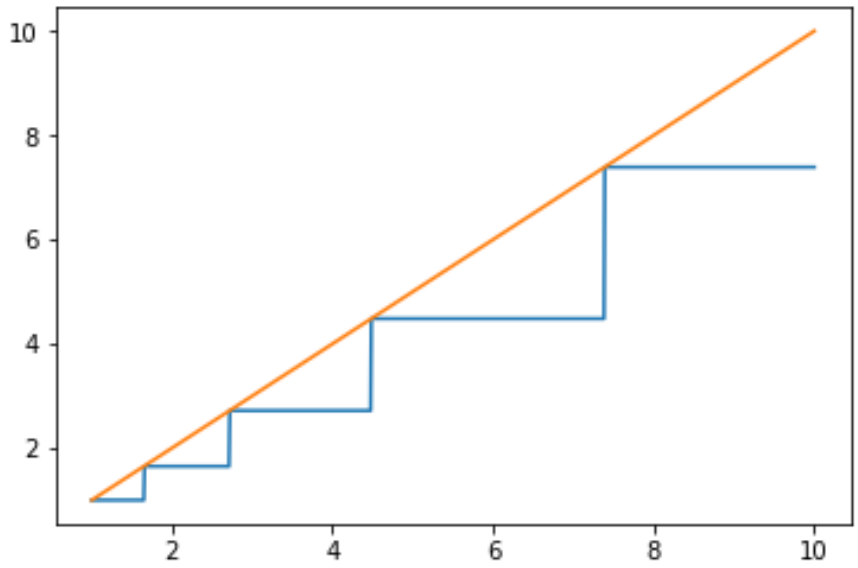
N = 53, the figure is as follow:



N = 52, the figure is as follow:



N = 51, the figure is as follow:



For N=53, all numbers are represented as either (1/2+1/2^53)*2 or (1/2)*2 after square root in computer, thus exponentiate them will yield either [(1/2+1/2^53)*2]^(2^53) = e^2 or 1.
For N=52, all numbers represented as (1/2)*2, (1/2+1/2^53)*2 or (1/2+1/2^52)*2, thus exponentiate them will yield 1, e or e^2.
The same thing holds for N=51 and other N

Problem 7

(a) 2432902008176640000,

-8752948036761600000,

1380375975364070400,

-1287093124515098880,

803781182264505177 6,

-359979517947607200,

120664780378037336 0,

-311333643161390640,

63030812099294896,

-10142299865511450,

1307535010540395,

-135585182899530,

11310276995381,

-756111184500,

40171771630,

-1672280820,

53327946,

-1256850,

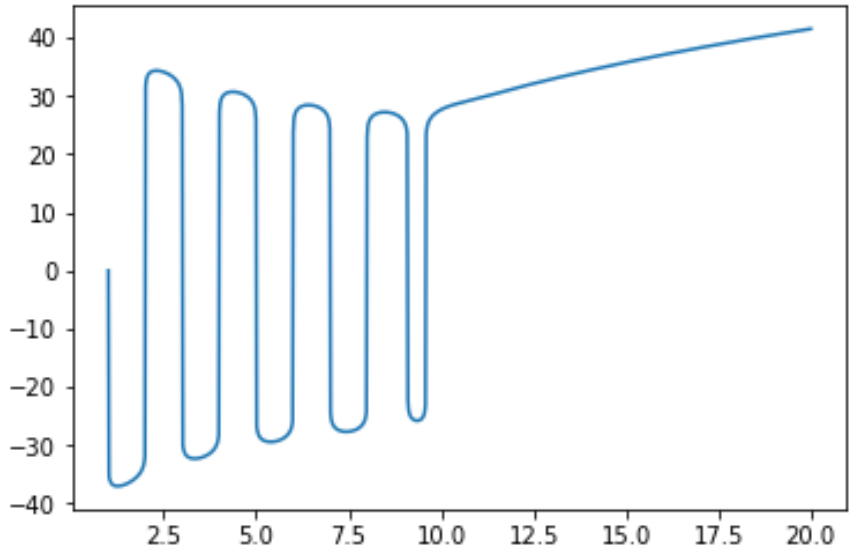
20615,

-210.0,

1

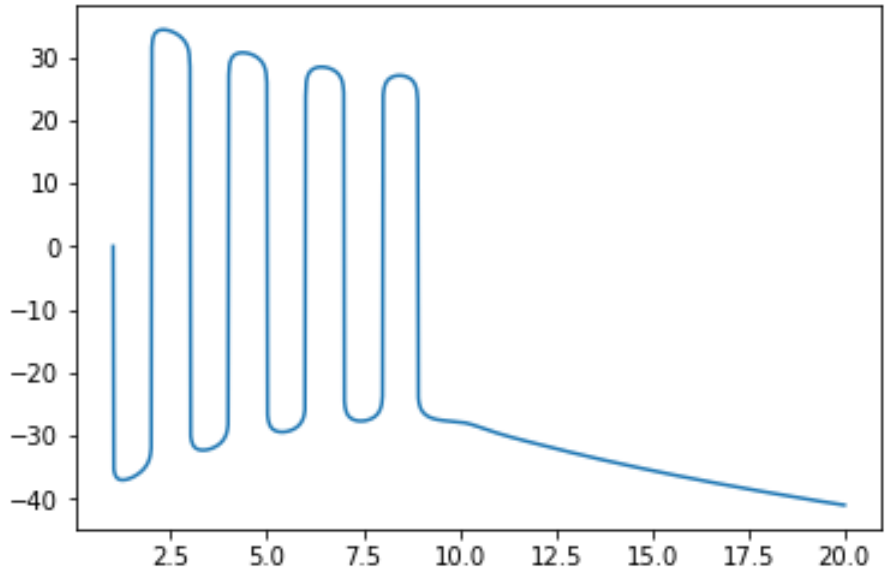
(b) use scipy.optimize.newton, the root is 19.9999949571418

(c) δ = 10−8, the root is 9.585389646516598
10−6, the root is 7.752713003402644
10−4, the root is 5.969334849605957
10−2, the root is 5.469592915093453



plot of sgn(w(x))*log(1+|w(x|) for δ = 10−8

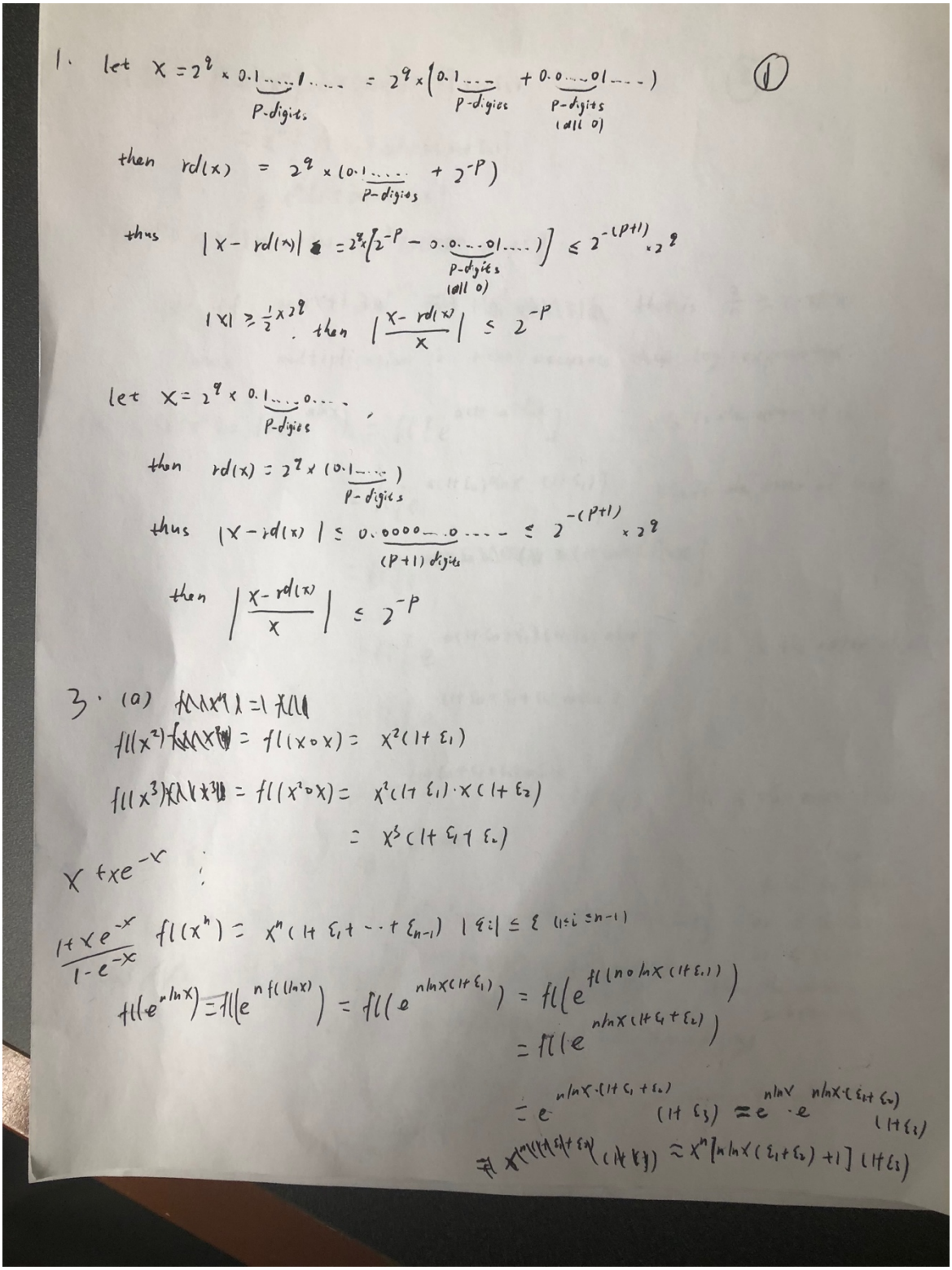
(d) for root 16, the value becomes 8.007305890325446
for root 17, the value becomes 8.917114218525562



plot of sgn(w(x))*log(1+|w(x|)

(e) see other part

Part 2



1. let $x = 2^q \times 0.1\underbrace{\ldots\ldots}_{p\text{-digits}}1\ldots\ldots = 2^q \times (0.1\underbrace{\ldots\ldots}_{p\text{-digits}} + 0.0\underbrace{\ldots\ldots}_{p\text{-digits (all 0)}}01\ldots\ldots)$ ①

then $rd(x) = 2^q \times (0.1\underbrace{\ldots\ldots}_{p\text{-digits}} + 2^{-p})$

thus $|x - rd(x)| = 2^q \times (2^{-p} - 0.0\underbrace{\ldots\ldots}_{p\text{-digits (all 0)}}01\ldots\ldots) \le 2^{-(p+1)} \times 2^q$

$|x| \ge \frac{1}{2} \times 2^q$, then $\left|\frac{x - rd(x)}{x}\right| \le 2^{-p}$

let $x = 2^q \times 0.1\underbrace{\ldots\ldots}_{p\text{-digits}}0\ldots\ldots$

then $rd(x) = 2^q \times (0.1\underbrace{\ldots\ldots}_{p\text{-digits}})$

thus $|x - rd(x)| \le 0.\underbrace{0000\ldots0}_{(p+1)\text{ digits}}\ldots\ldots \le 2^{-(p+1)} \times 2^q$

then $\left|\frac{x - rd(x)}{x}\right| \le 2^{-p}$

3. (a) $fl(x^2) = fl(x \circ x) = x^2(1+\varepsilon_1)$

$fl(x^3) = fl(x^2 \circ x) = x^2(1+\varepsilon_1) \cdot x(1+\varepsilon_2)$

$= x^3(1+\varepsilon_1+\varepsilon_2)$

$x + xe^{-x}$;

$\frac{1+xe^{-x}}{1-e^{-x}}$ $fl(x^n) = x^n(1+\varepsilon_1+\cdots+\varepsilon_{n-1})$ $|\varepsilon_i| \le \varepsilon$ $(i=1,\ldots,n-1)$

$fl(e^{n \ln x}) = fl(e^{n\, fl(\ln x)}) = fl(e^{n \ln x(1+\varepsilon_1)}) = fl(e^{fl(n \circ \ln x(1+\varepsilon_1))})$

$= fl(e^{n \ln x(1+\varepsilon_1+\varepsilon_2)})$

$= e^{n \ln x \cdot (1+\varepsilon_1+\varepsilon_2)}(1+\varepsilon_3) \approx e^{n \ln x} \cdot e^{n \ln x \cdot (\varepsilon_1+\varepsilon_2)}(1+\varepsilon_3)$

$\approx x^n[n \ln x(\varepsilon_1+\varepsilon_2)+1](1+\varepsilon_3)$

$fl(e^{n\ln x}) \approx x^n[n\ln x(\varepsilon_1+\varepsilon_2)+1](1+\varepsilon_3)$ ②

$\approx x^n[1+\varepsilon_3+(n\ln x)(\varepsilon_1+\varepsilon_2)]$

$\leq x^n[1+(2n\ln x+1)\varepsilon]$

$fl(x^n) = x^n(1+\varepsilon_1+\varepsilon_2) \leq x^n[1+(n-1)\varepsilon]$

so if $2n\ln x+1 \geq n-1$, that is $\frac{2}{n} \geq 1-2\ln x$

then multiplication is more accurate than log-exponential

(b) (i) $fl(x^a) = fl(e^{a\ln x}) = fl[e^{a(1+\varepsilon_a)\ln x}]$ (ε_a is the error of a)

$= fl[e^{a(1+\varepsilon_a)\ln x \cdot (1+\varepsilon_1)}]$ (ε_1 is the error of $\ln x$)

$= fl[e^{a(1+\varepsilon_a+\varepsilon_1)\ln x}]$

$= fl[e^{a(1+\varepsilon_a+\varepsilon_1)(1+\varepsilon_2)\ln x}]$ (ε_2 is the error of multiply)

$\int_x^{x+\varepsilon_x} f(u)du$ $= fl[e^{(1+\varepsilon_a+\varepsilon_1+\varepsilon_2)a\ln x}]$

$= e^{(1+\varepsilon_a+\varepsilon_1+\varepsilon_2)a\ln x}(1+\varepsilon_3)$ (ε_3 is the error of exp)

$x(1+\varepsilon_x)$

$x+x\varepsilon_x$ $\approx x^a[1+\varepsilon_3+(a\ln x)(\varepsilon_a+\varepsilon_1+\varepsilon_2)]$

if a is large or x is very small or x is very large. then the error will be substantial

$fl(x^a) = fl(e^{a\ln x}) = fl[e^{a\ln(x+\varepsilon_x)}]$ ε_x is the error of x

$= fl[e^{a\ln(x+\varepsilon_x)(1+\varepsilon_1)}]$ ε_1 is the error of $\ln x$

$= fl[e^{a\ln(x+\varepsilon_x)(1+\varepsilon_1+\varepsilon_2)}]$ ε_2 is the error of $a\ln x$

$= fl[e^{a\ln x+\ln(1+\frac{\varepsilon_x}{x})}]$

(ii) $fl(X^a) = fl(e^{a\ln X}) = fl\{e^{a\ln[X(1+\varepsilon_x)]}\} = fl\{e^{a[\ln X(1+\frac{\ln(1+\varepsilon_x)}{\ln X})]}\}$

$= fl\{e^{a\ln X(1+\varepsilon_1+\varepsilon_2)[1+\frac{\ln(1+\varepsilon_x)}{\ln X}]}\}$ ③

$= fl\{e^{a\ln X(1+\varepsilon_1+\varepsilon_2)(1+\frac{\varepsilon_x}{\ln X})}\}$

$= e^{a\ln X(1+\varepsilon_1+\varepsilon_2+\frac{\varepsilon_x}{\ln X})(1+\varepsilon_3)}$

$= e^{a\ln X}[1+(\varepsilon_1+\varepsilon_2)\cdot a\ln X + a\varepsilon_x](1+\varepsilon_3)$

$= e^{a\ln X}[1+(\varepsilon_1+\varepsilon_2)a\ln X + a\varepsilon_x + \varepsilon_3]$

thus, the propagated error of a is $X^a\cdot(a\ln X)\varepsilon_a$

the propagated error of X is $X^a\cdot a\varepsilon_x$

when a is very large or X is very small or X is very large
ε_a could be very substantial

when a is very large. ε_x could be substantial

4. (a) $(\text{cond}f)(x) = \dfrac{x f'(x)}{f(x)} = \dfrac{xe^{-x}}{1-e^{-x}} = \dfrac{x}{e^x - 1} \le 1$

(4)

(b) $f_A(x) = f(x_A)$

$f_A(x) = fl\left[1 - e^{-x(1+\varepsilon_x)}\right] = fl\left[1 - e^{-x}(1+x\varepsilon_x)\right]$

$= fl\left[1 - e^{-x}(1 - x\varepsilon_x + \varepsilon_1)\right]$ (ε_1 is the error of exp)

$= fl\left[1 - e^{-x}(1 - x\varepsilon_x + \varepsilon_1)\right](1+\varepsilon_2)$ (ε_2 is the error of multiplication)

$= (1 - e^{-x})\left[1 + \dfrac{e^{-x}}{1-e^{-x}}(-x\varepsilon_x + \varepsilon_1) + \varepsilon_2\right]$

□

$(1 - e^{-x})\left[f_A(x) - f(x)\right] = f(x_A) - f(x)$, this is

$(1 - e^{-x})\left[\dfrac{e^{-x}}{1-e^{-x}}(-x\varepsilon_x + \varepsilon_1) + \varepsilon_2\right] = f'(x)(x - x_A)$, thus

$x - x_A = \left[\dfrac{1}{1-e^{-x}}(-x\varepsilon_x + \varepsilon_1) + e^x\varepsilon_2\right](1 - e^{-x})$

$\therefore (\text{cond}A)(x) = \left|\dfrac{x - x_A}{x}\right| / \varepsilon = \left[\dfrac{1}{1-e^{-x}}(-x\varepsilon_x + \varepsilon_1) + e^x\varepsilon_2\right](1 - e^{-x}) / x\varepsilon$

$\le (\text{cond}f)(x) / \left[1 - e^{-x}\right]$

$= \left(\dfrac{1+x}{1-e^{-x}} + e^x\right) / x = \dfrac{x + e^x}{x} > 1$

if $\varepsilon_x = 0$, $(\text{cond}A)(x) = \dfrac{e^x}{x}$, is still larger than 1

(c) $(\text{cond}A)(x) = \left[(-x\varepsilon_x + \varepsilon_1)/x + \frac{e^x(1-e^{-x})}{x}\varepsilon_2\right]/\varepsilon$

$= \left[-\varepsilon_x + \frac{1}{x}\varepsilon_1 + \frac{e^x-1}{x}\varepsilon_2\right]/\varepsilon$.

if $\varepsilon_x = 0$, then $(\text{cond}A)(x) = \left[\frac{1}{x}\varepsilon_1 + \frac{e^x-1}{x}\varepsilon_2\right]/\varepsilon$

we can see that the error in exponentiating is expanded by $\frac{1}{x}$

thus it is the cause of poor conditioning

(d) $\left|\frac{f_A(x)-f(x)}{f(x)}\right| = \left|\frac{e^{-x}}{1-e^{-x}}(-x\varepsilon_x + \varepsilon_1) + \varepsilon_2\right|$ $(\varepsilon_x = 0)$

$= \left|\frac{e^{-x}\cdot\varepsilon_1}{1-e^{-x}} + \varepsilon_2\right| \le \varepsilon\left|\frac{e^{-x}}{1-e^{-x}} + 1\right| = \left|\frac{e^x}{e^x-1}\right|\cdot\varepsilon$

if we are willing to lose at most 1 bit of significance,

then $\left|\frac{e^x}{e^x-1}\right| \le 10$, we have $x \ge 0.1$

if we are willing to lose at most 2 bits

then $\left|\frac{e^x}{e^x-1}\right| \le 100$, $\frac{e^x}{e^x-1} \approx \frac{1}{x}$, thus $x \ge 0.01$

for 3 bits, $x \ge 0.001$, for 4 bits, $x \ge 0.0001$

(e) $\left|\frac{f_A(x)-f(x)}{f(x)}\right| \le \varepsilon\left|\frac{e^x}{e^x-1}\right|$

(f) we can compute $f(x) = 1-e^{-x}$ by computer $\frac{e^x-1}{e^x} \approx \frac{x+\frac{x^2}{2}+\cdots}{1+x+\frac{x^2}{2}+\cdots}$

for very small x

7. (a) (i) let $p(\overset{\Omega_k}{*}, a) = 0$

then we have $\frac{\partial p}{\partial \Omega_k} \cdot d\Omega_k + \sum_{l=0}^{n-1} \frac{\partial p}{\partial a_{*l}} da_{*l} = 0$

 (6)

thus $\frac{\partial \Omega_k}{\partial a_l} = -\frac{\frac{\partial p}{\partial a_l}}{\frac{\partial p}{\partial \Omega_k}} = -\frac{\Omega_k^l}{p'(\Omega_k)}$.

then $T_{kl} = \left| \frac{a_l \frac{\partial \Omega_k}{\partial a_l}}{\Omega_k} \right| = \left| \frac{a_l \Omega_k^{l-1}}{p'(\Omega_k)} \right|$

$(\text{cond } \Omega_k)(\vec{a}) = \sum_{l=0}^{n-1} \left| \frac{a_l \Omega_k^{l-1}}{p'(\Omega_k)} \right|$

(ii) for $r=14$, the condition number is 5.4×10^{13}

for $r=16$, the condition number is 3.5×10^{13}

for $r=17$, the condition number is 1.8×10^{13}

for $n=20$, the condition number is 1.4×10^{11}

the condition number for this problem is so large that even a small change in the coefficient, the root will change greatly.

(iii) no algorithm can help here, the problem is inherently ill-conditioned, small pertubation of coefficient can cause large changes in the root.

8(a) $\delta y_{n+1} = -(n+1)\,\delta y_n$, thus

$$\delta y_N = \frac{N!}{k!}(-1)^{N-k}\,\delta y_k \quad (k<N)$$

and since $y_{n+1} = \int_0^1 x^{n+1}e^x dx \le \int_0^1 x^n e^x dx = y_n$

we have $y_N \le y_k$, $(k<N)$

that is $\left|\frac{\delta y_N}{y_N}\right| \ge \frac{N!}{k!}\left|\frac{\delta y_k}{y_k}\right|$

so $\varepsilon_{y_k} \le \frac{k!}{N!}\varepsilon_{y_N}$, $(\text{cond } y_k)(y_N) \le \frac{k!}{N!}$

(b) if we want $\varepsilon_{y_k} \le \varepsilon$, we just need $\frac{k!}{N!}\varepsilon_{y_N} \le \varepsilon$,

that is $N! \ge \frac{k!}{\varepsilon}$

(c) $\varepsilon = 2^{-53}$, $k=20$, we have $N! \ge \frac{20!}{2^{-53}} \simeq 2.2\times10^{34}$

since $31! \simeq 8.2\times10^{33}$, $32! \simeq 2.6\times10^{35}$

we have $N=32$

(d) the value of backward recurrance relation is 0.12380383076256993,

the value from direct integration is 0.12380383076256998.

the difference is quite small