1. a. f’(x) = n(x+1)^(n-1) by chain rule, f’(0) = n
   f’’(x) = n(n-1)(x+1)^(n-2) by chain rule, f’’(0) = n(n-1)
   f’’’(x) = n(n-1)(n-2)(x+1)^(n-3) by chain rule, f’’’(0) = n(n-1)(n-2)
   (1+x)^n = 1 + nx + n(n-1)x^2/2! + n(n-1)(n-2)x^3/3! + …
   n(n-1)x^2/2! + n(n-1)(n-2)x^3/3! + … = o(x) at x=0 because lim(x to 0) (n(n-1)x^2/2! + n(n-1)(n-2)x^3/3! + …) / x = lim(x to 0) (n(n-1)x/2! + n(n-1)(n-2)x^2/3! + …) = 0
   Thus (1+x)^n = 1 + nx + o(x)

   b. lim(x to 0) xsin(sqrt(x))/x^(3/2) = lim(x to 0) sin(sqrt(x))/x^(1/2)
   By L’Hospital’s rule, lim(x to 0) sin(sqrt(x))/x^(1/2) = lim(x to 0) (cos(sqrt(x))/(2x^(½)))/(x^(-½)/2) = lim(x to 0) cos(sqrt(x)) = 1 thus xsin(sqrt(x)) = O(x^(3/2))

   c. lim(t to 0) e^(-t)/(1/t^2) = lim(t to 0) (t^2)/(e^t) = 0/1 = 0
   Thus e^(-t) = O(1/t^2)

   d. 0 <= e^(-x^2) <= 1 for all x
   Thus integral(0,ε) 0 dx <= integral(0,ε) e^(-x^2) dx <= integral(0,ε) 1 dx
   Thus 0 <= integral(0,ε) e^(-x^2) dx <= ε
   Thus integral(0,ε) e^(-x^2) = O(ε)

2. a. x_before = A^(-1)b
   x_after = A^(-1)b + A^(-1)*Delta(b)
   Delta(x) = A^(-1)*Delta(b)
   det(a) = ad-bc = -2*10^-10
   A^(-1) = 1/(2*10^-10) * [[1 - 10^(-10),-1],[-1 - 10^(-10),1]]
   Delta(x) = 1/(2*10^-10) * [[1 - 10^(-10),-1],[-1 - 10^(-10),1]]*[Delta(b1),Delta(b2)]^T

   b. K = ||A^-1||*||A|| = 3,636,363,700

   c. Delta(x) = 1/(2*10^-10) * [[1 - 10^(-10),-1],[-1 - 10^(-10),1]]*[z * 10^-5,y * 10^-5]^T = [z/2*10^-5,y/2*10^-5]
   Relative error = ||Delta(x)|| / ||x|| = 7.071 * 10^-6 / 1.414 = ½ * 10^-5
   The relative error is less than or equal to the condition number times the 2 norm of the perturbation over the 2 norm of b. If the perturbations are different, the relative error will differ for each component of x. Different perturbations are more realistic than the same ones because different sources of uncertainty may be affecting the problem.

3. a. f’(x) = e^x
   K = xe^x/(e^x - 1)

By L'Hospital's rule, lim(x to 0) xe^x/(e^x - 1) = lim(x to 0) (e^x + xe^x)/(e^x) = 1. Therefore, the function is not ill conditioned when x is close to 0 so there are no points at which x is ill conditioned.

b. This algorithm is stable because it computed e^x, which is stable, and y -1 where y is the result of a stable function, which is also stable. Despite the entire function f(x) = e^x -1, being unstable for values of x close to 0, this algorithm is stable because it separates it into two different functions that are stable.

c. e^(9.999999999500000 * 10^-10) evaluates to 1.000000000aaaaaaa where a represents another correct digit not shown on my calculator. That value minus 1 evaluates to 0.000000000aaaaaaa, so there are 7 correct significant digits.

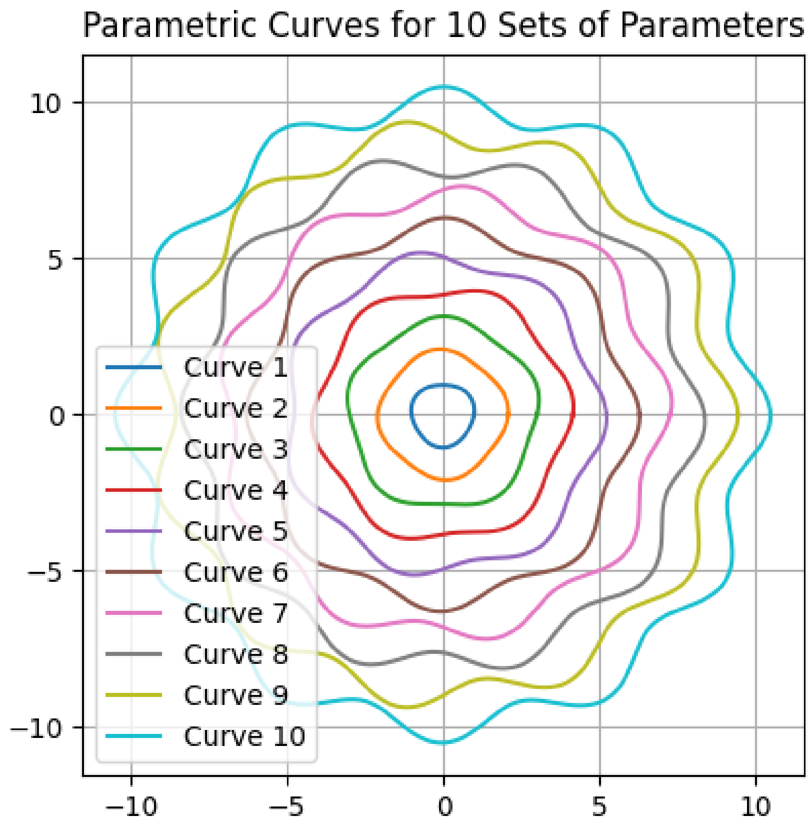
d. f(x) is approximately x + x^2/2! + x^3/3! + … by taylor series approximation.

e. f(9.999999999500000 * 10^-10) = 10^-9 + 5*10^-19 + (5*10^-28)/3 + …

4. Python code pushed to Git

   Sum from part a: -23.915381134014112

   Circles from part b:


Parametric Curves for 10 Sets of Parameters
10
5
0
−5
−10
−10
−5
0
5
10
Curve 1
Curve 2
Curve 3
Curve 4
Curve 5
Curve 6
Curve 7
Curve 8
Curve 9
Curve 10