HW 2

Derek Walker

1.

(a):

binomial theorem:

$$(1+x)^n = 1 + nx + \frac{n(n-1)}{2}x^2 + \text{higher order terms}$$

Let $g(x) = \frac{n(n-1)}{2}x^2$

$$\lim_{x \to 0} \left| \frac{g(x)}{x} \right| = \lim_{x \to 0} \left| \frac{n(n-1)}{2} x \right| = 0$$

$\Rightarrow \quad g(x) = o(x)$

$\Rightarrow (1+x^n) = 1 + nx + o(x)$

(b):

$\sin\sqrt{x} \approx \sqrt{x}$ as $x \to 0$

$\Rightarrow x\sin\sqrt{x} \approx x\sqrt{x} = x^{3/2} = O(x^{3/2})$

$\Rightarrow x\sin\sqrt{x} = O(x^{3/2})$

(c):

$$\lim_{t \to 0} \left| \frac{e^{-t}}{1/t^2} \right| = \lim_{t \to 0} \left| \frac{t^2}{e^t} \right| = \lim_{t \to 0} \left| \frac{2t}{e^t} \right| = 0$$

$\Rightarrow e^{-t} = o(1/t^2)$

(d):

$$\int_0^\varepsilon e^{-x^2} dx = \int_0^\varepsilon \sum_{k=0}^{\infty} \frac{x^{2k}}{k!} dx$$

$$\approx \int_0^\varepsilon \left(1 + x^2 + \frac{x^4}{2}\right) dx$$

$$= \varepsilon + O(\varepsilon^3)$$

$$\lim_{\varepsilon \to 0} \frac{\varepsilon + O(\varepsilon^3)}{\varepsilon} \approx 1$$

$\Rightarrow \int_0^\varepsilon e^{-x^2} dx = O(\varepsilon)$

2.

(a):

$$A(\vec{x}+\Delta\vec{x}) = \vec{b}+\Delta\vec{b}$$
$$A\vec{x} + A\Delta\vec{x} = \vec{b}+\Delta\vec{b}$$
$$A\vec{x}=\vec{b} \implies A\Delta\vec{x} = \Delta\vec{b}$$
$$\implies \Delta\vec{x} = A^{-1}\Delta\vec{b}$$

(b):

$$\text{relerr}_x = \frac{\|\Delta x\|}{\|x\|} = \frac{\|A^{-1}\Delta b\|}{\|x\|}$$

$$\kappa_A(b) = \lim_{\|\Delta b\|\to 0} \frac{\|A^{-1}\Delta b\|/\|x\|}{\|\Delta b\|/\|b\|}$$
$$= \lim_{\|\Delta b\|\to 0} \frac{\|A^{-1}\Delta b\|\,\|b\|}{\|x\|\,\|\Delta b\|}$$
$$= \lim_{\|\Delta b\|} \frac{\|A^{-1}\Delta b\|\,\|Ax\|}{\|x\|\,\|\Delta b\|}$$
$$\leq \|A\|_2\|A^{-1}\|_2 = \kappa_A$$

$$\implies \kappa_A = \|A\|\,\|A^{-1}\|$$

using different norm ($\|\cdot\|_\infty$) b/c easier

$$\kappa_A = \|A\|_\infty \|A^{-1}\|_\infty$$
$$= 1\cdot 1$$
$$= 1$$

(c):

$$\text{relerr}_x = \frac{\|\Delta x\|}{\|x\|}$$

$$\|\Delta b\| = 10^{-5} = \|A\Delta x\| \leq \|A\|\,\|\Delta x\|$$

$$\implies \text{relerr}_x \leq \frac{\|\Delta b\|}{\|A\|\,\|x\|} = \frac{\|\Delta b\|}{\|x\|} = \frac{10^{-5}}{\sqrt{2}}$$

The behavior will be different for different Δb b/c $A\Delta x = \Delta b$ w/ A exact. The more realistic situation is w/ different Δb.

larger perturbation $\implies$ larger relerr

κ_A stays the same

3.

3]

(a):

$$K_f(x) = |f'(x)| \frac{|x|}{|f(x)|}$$

$$= e^x \frac{|x|}{|e^x - 1|}$$

$$= \frac{|x|e^x}{|e^x - 1|}$$

$X=0$ results in an asymptote

$\Rightarrow K_f \to \infty \Rightarrow$ ill-conditioned

(b):

This algorithm is stable for $x>0$ but unstable for $x \to 0$ due to the loss of precision w/ subtraction. for $x>0$, K goes down I think.

(C):

I get 1.0×10^{-9} up to 7 decimal digits. This is less than 16 which makes sense b/c subtraction is involved & x is close to 0. $\Rightarrow$ loss of precision

(d):

$$f(x) = -1 + e^x$$
$$= -1 + \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Want $\text{relerr}_f(x) \le 10^{-16}$

$$f(x) = -1 + 1 + x + \frac{x^2}{2} + \frac{x^3}{6}$$
$$= x + \frac{x^2}{2} + \frac{x^3}{6}$$

(e):

It does.

(f):

It seems to work w/ $f(x) = x + \frac{x^2}{2}$ as well to 16 digits.

(g):

Cool.

4.

(a):

See code in APPM-4600 github repository or attached at the end of this document. I got -17.545259710757044 as the sum.

(b):

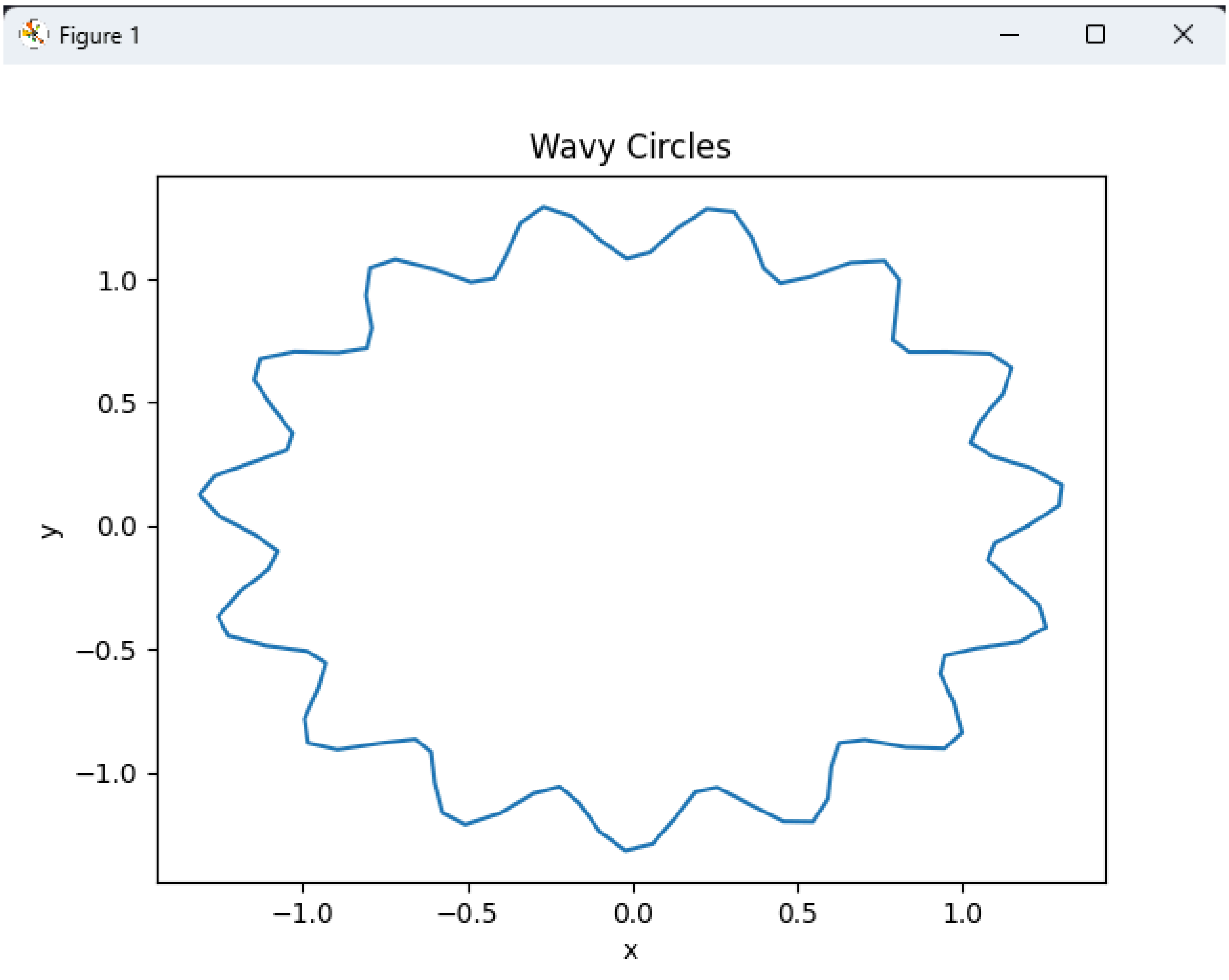

Figure 1: Parametric wavy circle curve

Figure 1 used one iteration of the x and y functions. See code in APPM-4600 github repository or attached at the end of this document.

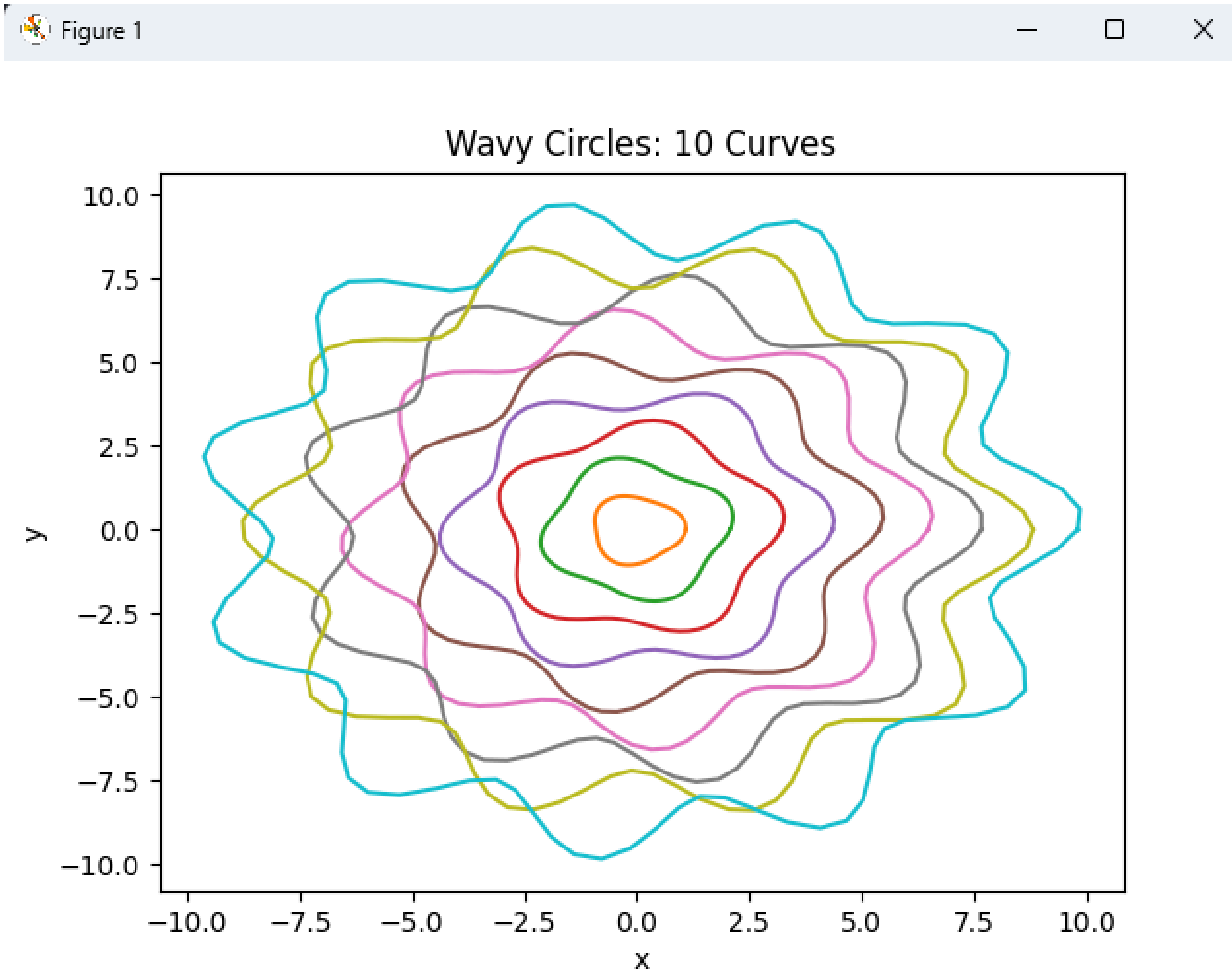

Figure 2: 10 wavy circles graphed using varied iterations of the parametric curve in Figure 1.

See code in APPM-4600 github repository or attached at the end of this document.

```
import math
import random
import numpy as np
import matplotlib.pyplot as plt

### Q3
def f3(x):
    y = math.exp(x)
    return y-1

def poly_f3(x):
    return x + (1/2)*x**2 + (1/6)*x**3

print(f3(9.999999995000000*10**-10))
print(poly_f3(9.999999995000000*10**-10))


### Q4.a
t = np.arange(0, np.pi, np.pi / 30, dtype=float)
t = np.append(t, t[len(t)-1] + np.pi/30) # one more iteration
y = np.cos(t)

def sum_fn(y, t):
    sum = 0
    for i in range(0, len(t)-1):
        sum += t[i] * y[i]
    return sum

print(f"the sum is: {sum_fn(y, t)}")


### Q4.b1
R = 1.2
dr = 0.1
f = 15
p = 0

def x_fn(t):
    return R * (1 + dr * math.sin(f * t + p)) * math.cos(t)

def y_fn(t):
    return R * (1 + dr * math.sin(f * t + p)) * math.sin(t)

theta = np.linspace(0, 2*np.pi, 100)
xs = [x_fn(t) for t in theta]
ys = [y_fn(t) for t in theta]
plt.plot(xs, ys)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Wavy Circles')
plt.show()

### Q4.b2
plt.xlabel('x')
plt.ylabel('y')
plt.title('Wavy Circles: 10 Curves')

for i in range(0, 10):
    R = i
```

```
    dR = 0.05
    f = 2+i
    p = random.uniform(0, 2)

    xs = [x_fn(t) for t in theta]
    ys = [y_fn(t) for t in theta]

    plt.plot(xs, ys)

plt.show()
```