APPM 4600 - HW #2 Gustav Cedergrund

1)

a) Show $(1+x)^n = 1 + nx + o(x)$ as $x \to 0$

$$\lim_{x\to0} \frac{(1+x)^n - 1 - nx}{x} = \lim_{x\to0} \frac{\left(1+nx+\frac{n(n-1)}{2!}x^2+\ldots\right) - 1 - nx}{x} \leftarrow \text{binomial formula}$$

$$= \lim_{x\to0} \frac{\frac{n(n-1)}{2!}x^2 + \frac{n(n-1)(n-2)}{3!}x^3 + \ldots}{x} \cdot \frac{\frac{1}{x}}{\frac{1}{x}}$$

$$= \lim_{x\to0} \frac{\frac{n(n-1)}{2!}x + \frac{n(n-1)(n-2)}{3!}x^2 + \ldots}{1} = \underline{0}$$ so $(1+x)^n - 1 - nx$ is $o(x)$

$\therefore (1+x)^n = 1 + nx + o(x)$ as $x \to 0$ ■

b) Show $x\sin(\sqrt{x}) = O(x^{3/2})$ as $x \to 0$

$$\lim_{x\to0^+} \frac{x\sin(\sqrt{x})}{x^{3/2}} = \lim_{x\to0^+} \frac{\sin(\sqrt{x})}{\sqrt{x}}$$

$$\overset{LH}{=} \lim_{x\to0^+} \frac{\cos(\sqrt{x}) \cdot \frac{1}{\cancel{2}x^{1/2}}}{\frac{1}{\cancel{2}x^{1/2}}} = \lim_{x\to0^+} \cos(\sqrt{x}) = \boxed{1} \neq 0$$

$\therefore x\sin(\sqrt{x}) = O(x^{3/2})$ as $x \to 0$ ■

c) Show $e^{-t} = o\left(\frac{1}{t^2}\right)$ as $t \to \infty$

$$\lim_{t\to\infty} \frac{e^{-t}}{1/t^2} = \lim_{t\to\infty} \frac{t^2}{e^t}$$

$$\overset{LH}{=} \lim_{t\to\infty} \frac{2t}{e^t}$$

$$\overset{LH}{=} \lim_{t\to\infty} \frac{2}{e^t} = \boxed{0}$$

$\therefore e^{-t} = o\left(\frac{1}{t^2}\right)$ as $t \to \infty$ ■

d) Show $\int_0^\varepsilon e^{-x^2}dx = O(\varepsilon)$ as $\varepsilon \to 0$

$$\lim_{\varepsilon\to0} \frac{\int_0^\varepsilon e^{-x^2}dx}{\varepsilon} \overset{LH}{=} \lim_{\varepsilon\to0} \frac{e^{-\varepsilon^2}}{1}$$

$$= \lim_{\varepsilon\to0} e^{-\varepsilon^2} = e^{-0^2} = e^0 = \boxed{1} \neq 0$$

$\therefore \int_0^\varepsilon e^{-x^2}dx = O(\varepsilon)$ as $\varepsilon \to 0$ ■

2) $A = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1+10^{-10} & 1-10^{-10} \end{bmatrix}$ $\vec{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$A^{-1} = \begin{bmatrix} 1-10^{10} & 10^{10} \\ 1+10^{10} & -10^{10} \end{bmatrix}$ $\vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

Here, we are solving $A\vec{x} = \vec{b}$ w/ some perturbation in b of $\begin{bmatrix} \Delta b_1 \\ \Delta b_2 \end{bmatrix}$

a) An exact formula for change in solution b/w exact problem and perturbed problem Δx is

$$\|\Delta x - x\| = \left\| A^{-1}\begin{bmatrix} 1+\Delta b_1 \\ 1+\Delta b_2 \end{bmatrix} - x \right\|$$

$$= \left\| \begin{bmatrix} 1-10^{10} & 10^{10} \\ 1+10^{10} & -10^{10} \end{bmatrix}\begin{bmatrix} 1+\Delta b_1 \\ 1+\Delta b_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|$$

$$= \left\| \begin{bmatrix} \Delta b_1(1-10^{10}) + 10^{10}\Delta b_2 + 1 - 1 \\ \Delta b_1(1+10^{10}) - 10^{10}\Delta b_2 + 1 - 1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} \Delta b_1(1-10^{10}) + 10^{10}\Delta b_2 \\ \Delta b_1(1+10^{10}) - 10^{10}\Delta b_2 \end{bmatrix} \right\|$$

$$\boxed{= \sqrt{2\Delta b_1^2(10^{20}+1) - 4\Delta b_1 \Delta b_2 \cdot 10^{20} + 10^{20}\cdot 2\Delta b_2^2}}$$

b) Condition # of A, by definition, is given (bounded) by formula

$$K(A) = \|A^{-1}\| \cdot \|A\| = (-2\cdot 10^{10})(-10^{-10}) = \frac{-2}{-1}\cdot\frac{10^{10}}{10^{10}}$$
$$= \boxed{2}$$

c) Let $\Delta b_1, \Delta b_2$ be of magnitude 10^{-5}

relative error is $\frac{\|\Delta x - x\|}{\|x\|} = \frac{\sqrt{2\Delta b_1^2(10^{20}+1) - 4\Delta b_1\Delta b_2\cdot 10^{20} + 10^{20}\cdot 2\Delta b_2^2}}{\sqrt{2}}$

w/ $\Delta b_1, \Delta b_2$ different magnitude $\rightarrow \leq 2\cdot 10^5$

Relationship b/w rel. error, condition #, and the perturbation is direct.

If $\Delta b_1, \Delta b_2$ are same

$$\|\Delta x - x\| = \sqrt{\cancel{2\Delta b^2\cdot 10^{20}} + 2\Delta b^2 - \cancel{4\Delta b^2\cdot 10^{20}} + \cancel{2\Delta b^2\cdot 10^{20}}} = \sqrt{2\Delta b^2} = \sqrt{2}\,\Delta b$$

s.t relative error is simply equal to the magnitude of perturbation.

I feel perturbations are more likely to be the same as the perturbations themselves are usually patterned/systematic as they come from the same place. Thus, they would cause perturbations that are similar, or the same. If the perturbation came from a random source, however, the perturbations would be different.

3) Let $f(x) = e^x - 1$

a) Find relative condition # $\kappa(f(x))$. By formula,

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

so, $f(x) = e^x - 1$
$f'(x) = e^x$ s.t.

$$\boxed{\kappa(f(x)) = \frac{|xe^x|}{|e^x - 1|} = \frac{xe^x}{e^x - 1} \quad \forall x \in \mathbb{R}}$$

No specific values for x on which $f(x)$ is ill-conditioned, but condition # increases linearly as $x \to \infty$

b) Computing $f(x)$ using the algorithm:

1: y = math.e^x ; 2: return $y-1$

Is the algorithm stable?

Let ε be a perturbation s.t. $\tilde{x} = x + \varepsilon$,

then $\tilde{y} = e^{(x+\varepsilon)} \Rightarrow \tilde{y} = e^x e^\varepsilon$ s.t. $\tilde{y} - 1$ is $e^x e^\varepsilon - 1$

As algorithm produces error of around e^ε for error ε, I would say algorithm is not stable, especially when input $x \approx 0$ b/c then there would be lots of cancellation of significant digits.

c) Let $x = 9.999999995000000 \cdot 10^{-10}$ s.t. $f(x) = 10^{-9}$ to 16 dec. places.

Algorithm produces $1.000000082740371 \cdot 10^{-9}$, which is correct to 8 digits (7 decimals). It is expected as when x is close to 0, we subtract two floats that are close to each other s.t. cancellation occurs.

d) Find poly. representation of $f(x)$ accurate to 16 digits for 'x' from (c)

Using Taylor,

$$f(x) = e^x - 1 = \left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots\right) - 1 = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

Need to be accurate to 16 digits (relative error), $\therefore$ we need Taylor expansion to $\frac{\text{abs. error}}{x} \le 10^{-16} \Rightarrow \text{abs. error} \le 9.999999995 \cdot 10^{-25}$.

$\therefore$ we need $R_n \le 9.999999995 \cdot 10^{-25}$

continued

3) continued

d) continued

$$R_n \leq 9.99999995 \cdot 10^{-25}$$

when $n=3$ as

$$e^x \cdot \frac{x^3}{3!} = 1.6666666658 \cdot 10^{-28} < 9.99999995 \cdot 10^{-25}$$

Thus, polynomial representation of $f(x)$ accurate to 16 digits for x is $\boxed{f(x) = x + \frac{x^2}{2!}}$

e) $f(x) = x + \frac{x^2}{2!}$ evaluated as $x = 9.999999995000000 \cdot 10^{-16}$ is 10^{-9}.

$\therefore$ (d) is correct.

---

4) Codes and outputs attached.

Problem 4

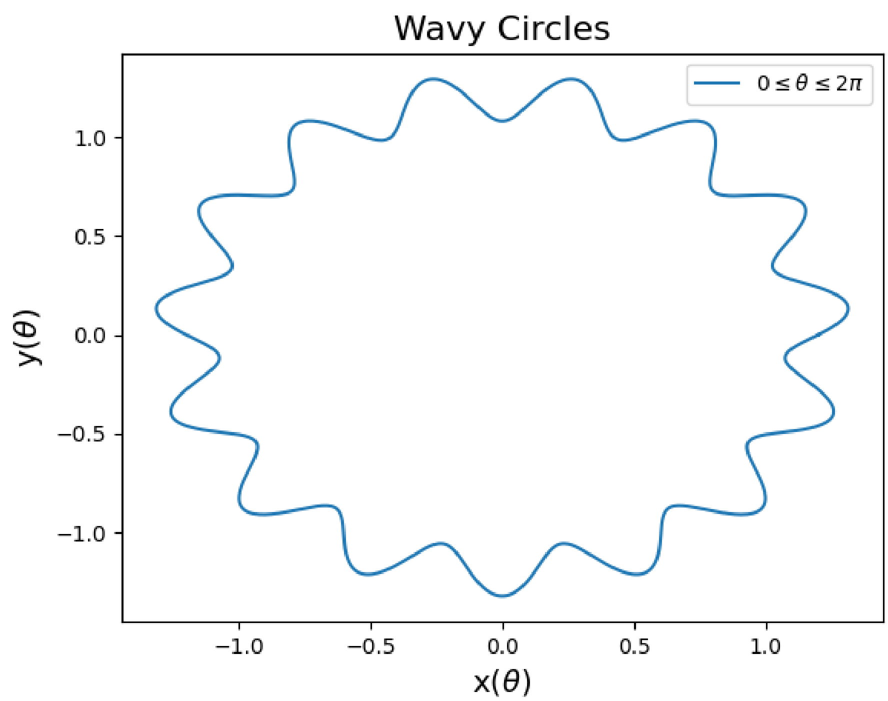
a)
Code:

```python
import numpy as np


def driver():
    # define vectors
    t = np.arange(0, np.pi, np.pi / 30)
    y = np.cos(t)

    # compute sum
    S = 0
    for k in range(len(t)):
        S += t[k] * y[k]

    # print sum and stop
    print("the sum is:", S)
    return


driver()
```

Output:

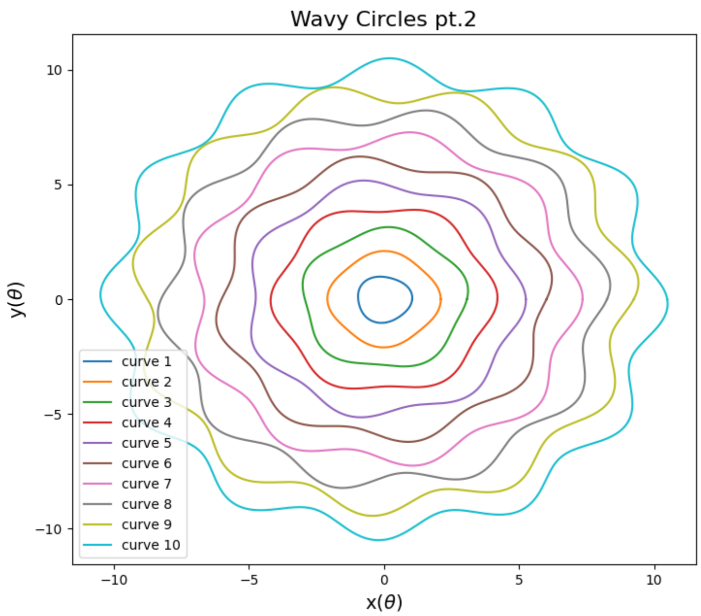
```
cedergrund@Gustavs-MacBook-Pro:~/Documents/fall-2023/APPM4600/homework/hw2/py_files|⇒  python3 prob4a.py
the sum is: -17.545259710757044
```

b)
i)




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


def driver():
    # constants
    R = 1.2
    delta_r = 0.1
    f = 15
    p = 0

    # create parametric curves
    x = lambda x: R * (1 + delta_r * np.sin(f * x + p)) * np.cos(x)
    y = lambda x: R * (1 + delta_r * np.sin(f * x + p)) * np.sin(x)

    # map parametric functions over theta domain
    theta = np.linspace(0, 2 * np.pi, 1000)
    x_vals = x(theta)
    y_vals = y(theta)

    # plot figure and stop
    ax = plt.figure().add_subplot()
    ax.plot(x_vals, y_vals, label="$0\leq \\theta \leq 2\pi$")
    ax.set_title("Wavy Circles", fontsize=16)
    ax.set_xlabel("x($\\theta$)", fontsize=14)
    ax.set_ylabel("y($\\theta$)", fontsize=14)
    ax.legend()
    plt.show()
    return


driver()
```

ii)




```python
import numpy as np
import matplotlib.pyplot as plt


def driver():
    # create parametric curves
    x = (
        lambda theta, R, delta_r, f, p: R
        * (1 + delta_r * np.sin(f * theta + p))
        * np.cos(theta)
    )
    y = (
        lambda theta, R, delta_r, f, p: R
        * (1 + delta_r * np.sin(f * theta + p))
        * np.sin(theta)
    )

    # plot 10 parametric functions using for loop
    theta = np.linspace(0, 2 * np.pi, 1000)
    x_vals = np.zeros((10, 1000))
    y_vals = np.zeros((10, 1000))
    ax = plt.figure().add_subplot()

    for i in range(10):
        curve_num = i + 1
        R = curve_num
        delta_r = 0.05
        f = 2 + curve_num
        p = np.random.uniform(0, 2)

        x_vals[i] = x(theta, R, delta_r, f, p)
        y_vals[i] = y(theta, R, delta_r, f, p)

        ax.plot(x_vals[i], y_vals[i], label="curve " + str(curve_num))

    # finish plotting and stop
    ax.set_title("Wavy Circles pt.2", fontsize=16)
    ax.set_xlabel("x($\\theta$)", fontsize=14)
    ax.set_ylabel("y($\\theta$)", fontsize=14)
    ax.legend()
    plt.show()
    return


driver()
```