APPM 4600 — HOMEWORK #2

Jaden Snell

February 7, 2024

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

$$\begin{aligned}(1+x)^n &= \sum_{k=0}^{n} \binom{n}{k} x^k \\ &= \binom{n}{0} x^0 + \binom{n}{1} x^1 + \sum_{k=2}^{n} \binom{n}{k} x^k \\ &= 1 + nx + \sum_{k=2}^{n} \binom{n}{k} x^k\end{aligned}$$

We can classify $\sum_{k=2}^{n} \binom{n}{k} x^k \in o(x)$ because every x^k term after $k \geq 2$ becomes negligible as $x \to 0$.We can use the limit $\frac{f(x)}{g(x)}$ as x approaches to prove this with $f(x) = \binom{n}{2} x^2$ and $g(x) = x$

$$\frac{f(0)}{g(0)} = \frac{0}{0}$$

Using L'Hopital's rule, the limit simplifies to

$$\frac{f'(x)}{g'(x)} = \frac{2x}{1}$$

which approaches 0 as x approaches 0 thus proving that $(1+x)^n = 1 + nx + o(x)$ as $x \to 0$.

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

$$\sin(x) \approx x \text{ for small x}$$

With the identity above, we can reasonably say that as $x \to 0$, $\sin(\sqrt{x}) \approx \sqrt{x}$. Meaning that

$$x \sin(\sqrt{x}) \approx x^{\frac{3}{2}} \in O(x^{\frac{3}{2}}) \text{ as } x \to 0.$$

because the limit $\frac{f(x)}{g(x)} = \frac{x^{\frac{3}{2}}}{x^{\frac{3}{2}}} = 1$.

(c) Show that $e^{-t} = o(\frac{1}{t^2})$ as $t \to \infty$.
We can show $e^{-t} = o(\frac{1}{t^2})$ as $t \to \infty$ through the limits below.

$$\begin{aligned}
\lim_{t\to\infty} \frac{e^{-t}}{\frac{1}{t^2}} &= \lim_{t\to\infty} \frac{t^2}{e^t} \\
&= \lim_{t\to\infty} \frac{2t}{e^t} \\
&= \lim_{t\to\infty} \frac{2}{e^t} \\
&= 0
\end{aligned}$$

Thus showing through L'Hopital's rule that e^{-t} converges to zero much faster than $\frac{1}{t^2}$, therefore $e^{-t} \in o(\frac{1}{t^2})$ as $t \to \infty$.

(d) Show that $\int_0^\epsilon e^{-x^2}\, dx = O(\epsilon)$ as $\epsilon \to 0$.
Given the function, $f(x) = e^{-x^2}$, and that we know $f(x)$ is decreasing for $x > 0$, $0 \leq x \geq \epsilon$. This means $e^{-x^2} \leq 1$ as $e^0 = 1$ giving

$$\begin{aligned}
\int_0^\epsilon e^{-x^2} dx &\leq \int_0^\epsilon e^{-0^2} dx \\
&\leq \int_0^\epsilon 1\ dx \\
&\leq \epsilon
\end{aligned}$$

. Therefore, $\int_0^\epsilon e^{-x^2} \in O(\epsilon)$ as $\epsilon \to 0$.

2. Consider solving $Ax = b$ where $A = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1+10^{-10} & 1-10^{-10} \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. The exact solution is $x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and the inverse of A is $\begin{bmatrix} 1-10^{10} & 10^{-10} \\ 1+10^{10} & -10^{-10} \end{bmatrix}$. In this problem we will investigate a perturbation in b of $\begin{bmatrix} \Delta b_1 \\ \Delta b_2 \end{bmatrix}$ and the numerical effects of the condition number.

(a) Find an exact formula for the change in the solution between the exact problem and the perturbed problem Δx. The perturbation of the problem yields

$$A(x + \Delta x) = b + \Delta b$$

Subtracting the original system from the perturbed system yields

$$A\Delta x = \Delta b$$

Assuming A is inveritble yields the solution for δx

$$\Delta x = A^{-1}\Delta b$$

which is the exact solution for the change in the solution between the original system and the perturbed system.

(b) What is the condition number of A? The condition number of a matrix A is $||A||_\infty \; ||A^{-1}||_\infty$ **?**. Plugging in A from this problem into this formula yields

$$\begin{aligned}
\kappa(A) &= ||A||_\infty \; ||A^{-1}||_\infty \\
&= ||\begin{bmatrix} 1 & 1 \\ 1 + 10^{-10} & 1 - 10^{-10} \end{bmatrix}||_\infty \; ||\begin{bmatrix} 1 - 10^{10} & 10^{10} \\ 1 + 10^{10} & -10^{10} \end{bmatrix}||_\infty \\
&= 2 \cdot 2 \times 10^{11} \\
&= 4 \times 10^{11}
\end{aligned}$$

which reveals that A must be ill conditions due to the large condition number.

(c) Let Δb_1 and Δb_2 be of magnitude 10^{-5}, not necessarily the same value. What is the relative error in the solution? What is the relationship between the relative error, the condition number, and the perturbation. Is the behavior different if the perturbations are the same? Which is more realistic: same value of perturbation or different value of perturbation?

The relative error in the solution due to perturbation Δb can be represented as

$$\frac{\|\Delta x\|}{\|x\|} \geq \kappa(A) \frac{\|\Delta b\|}{\|b\|}$$

and different perturbations will commonly yield different relative errors despite the same magnitude because of the ill-conditioned nature of the matrix A. It's also more realistic to have different perturbations because data is rarely uniform which necessitates a well-conditioned problem.

3. Recall the concept of a relative condition number $\kappa_{\text{f}}(x)$ for a function $f(x)$. For $\widetilde{x} = x + \delta x$, and $\delta x \to 0$, it gives us an upper bound on the relative error on the output $\widetilde{y} = f(\widetilde{x})$. That is:

$$\frac{|f(\widetilde{x}) - f(x)|}{|f(\widetilde{x})|} \leq \kappa_{\text{rel}}(x) \frac{|\widetilde{x} - x|}{|x|}$$

For a differentiable function $f(x)$, there is a formula for the relative condition number:

$$\kappa_{\text{f}}(x) = \frac{|xf'(x)|}{|f(x)|}$$

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

(a) What is the relative condition number $\kappa_{\text{f}}(x)$? Are there any values of x for which this is ill-conditioned (for which $\kappa_{\text{f}}(x)$ is very large)?

The relative condition number for the function $f(x) = e^x - 1$, is

$$\kappa_{\text{f}}(x) = \frac{|xe^x|}{|e^x - 1|}$$

yielding an ill-conditioned problem for any large value of x.

(b) Consider computing $f(x)$ via the following algorithm:

```
1: y = math.e^{x}
2: return y - 1
```

Is this algorithm stable? Justify your answer.

No, this algorithm is not stable because as x approaches 0, e^x approaches 1 which leads to cancellation errors the closer to 1 that e^x gets through the subtraction occurring in the return statement.

(c) Let x have the value $9.999999995000000 \times 10^{-10}$, in which case the true value for $f(x)$ is equal to 10^{-9} up to 16 decimal places. How many correct digits does the algorithm listed above give you? Is this expected?

The algorithm listed gave 7 digits of accuracy, 0.000000001 which is about what is expected for a as the machine uses a simpler Taylor expansion to derive the answer making it prune to the large cancellation errors.

(d) Find a polynomial approximation of $f(x)$ that is accurate to 16 digits for $x = 9.999999995000000 \times 10^{-10}$. Hint: use Taylor series, and remember that 16 digits of accuracy is a relative error, not an absolute one.
To approximate $e^x - 1$ to 16 digits of accuracy at $x = 9.999999995 \times 10^{-10}$ using its Taylor series, we expand $e^x - 1$ around 0:

$$e^x - 1 = x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

The error in the Taylor series approximation after n terms is given by the next term in the series. For e^x, the error term is approximately:

$$E_n \approx \frac{x^{n+1}}{(n+1)!}$$

We want this error to be less than 10^{-16}. Thus, we find the smallest n such that:

$$\frac{x^{n+1}}{(n+1)!} < 10^{-16}$$

Given $x = 9.999999995 \times 10^{-10}$, we find that the error term is less than 10^{-16} when $n = 1$. Therefore, the polynomial approximation of e^x to 16 digits of accuracy at the specified x is:

$$e^x - 1 \approx x$$

This approximation is sufficient due to the extremely small value of x, which renders higher-order terms negligible for this level of precision.

(e) Verify that your answer from part (d) is correct.

I pushed a file to the Homework 2 folder in my github repository containing the code that proves this in the file `polynomial.py`.

(f) [Optional] How many digits of precision do you have if you do a simpler Taylor series?

(g) [Fact; no work required] Matlab provides `expm1` and Python provides `numpy.expm1` which are special-purpose algorithms to compute $e^x - 1$ for $x \approx 0$. You could compare your Taylor series approximation with `expm1`.

4. **Practicing Python**

(a) Create a vector t with entries starting at 0 incrementing by $\frac{\pi}{30}$ to π. Then create the vector $y = \cos(t)$. Write a code that evaluates the following sum:

$$S = \sum_{k=1}^{n} t(k)y(k)$$

Print the statement "The sum is S" with the numerical value of S.

(b) **Wavy circles.** In one figure, plot the parametric curve

$$x(\theta) = R(1 + \delta r \sin(f\theta + p)) \cos(\theta)$$
$$y(\theta) = R(1 + \delta r \sin(f\theta + p)) \sin(\theta)$$

for $0 \leq \theta \leq 2\pi$ and for $R = 1.2$, $\delta r = 0.1$, $f = 15$ and $p = 0$. Make sure to adjust the scale so that the axis have the same scale.

In a second figure, use a for loop to plot 10 curves and let $R = i$, $\delta r = 0.05$, $f = 2 + i$ for the i-th curve. Let the value of p be a uniformly distributed random number (look up `random.uniform`) between 0 and 2.

The code with the sum of the vector outputted and for the figures is in the file `pypractice.py`. The figures are in `parametric-curve.png` and `wavybands.png` respectively. All files can be found in the **Homework 2** file of the github repository shared with the instructor.