AERO-222: Introduction to Aerospace Computation - Fall 2021
Homework #2 - Due Date: Wednesday, October 6, 2021

Show all work and justify your answers!

Instructions

- *This homework contains both handwritten and coding problems and shall be submitted according to the following guidelines.*
- *Hardcopy:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall include screenshots of any hand-written work.*
  - *For coding problems, the hardcopy shall include any relevant derivations and emphasize the final results (i.e. boxed, highlighted, etc.). INCLUDE ALL CODING RESULTS (including plots, final values) IN THE HARDCOPY.*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.pdf"
- *Coding Submission:*
  - *Due on CANVAS at 11:59 PM on the day of the deadline.*
  - *Shall be submitted as a single file according to the provided template with the following naming scheme:* "LastnameHW#.py"
  - *The script shall print out all outputs asked for in the problem.*
- *Late submissions will be accepted with a 10 point deduction per day late.*

---

1. **Newton's Method (15 pts) Code.** Apply the Newton method to find the root of the equation, $\sqrt{x+1} = e^x - 1$, using $x_0 = 0$ as an initial guess.
   1. Report the final estimated error (ε_x), **not the residual**, of your solution using the prescribed residual tolerance: $|f(x_k)| < \varepsilon_y = 10^{-12}$. Report also the number of iterations.
   2. Repeat the same problem, but stop the iterations when $|x_{k+1} - x_k| \geq |x_k - x_{k-1}|$ is satisfied. Report the number of iterations.
   3. Plot the convergence criteria as a function of iteration number from parts 1) and 2)

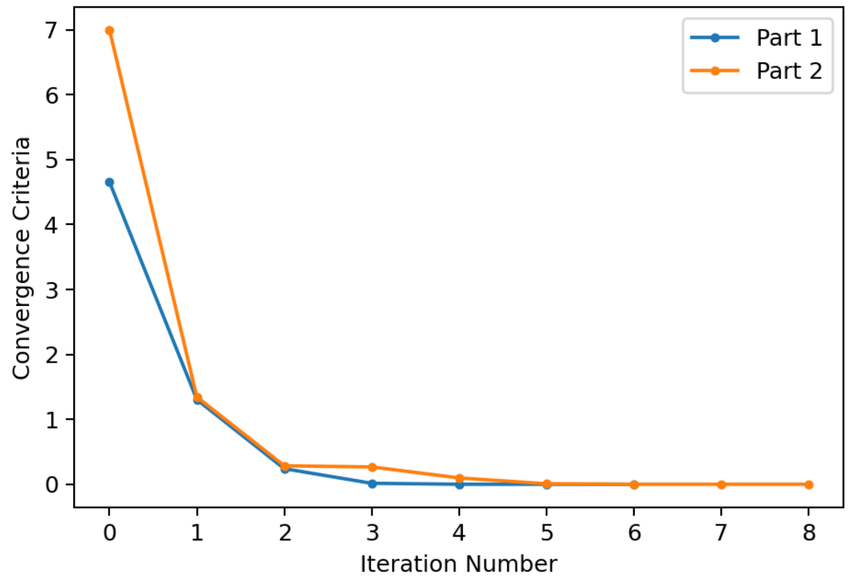

Figure 1: The general trend should look like this, all solutions converge to zero.

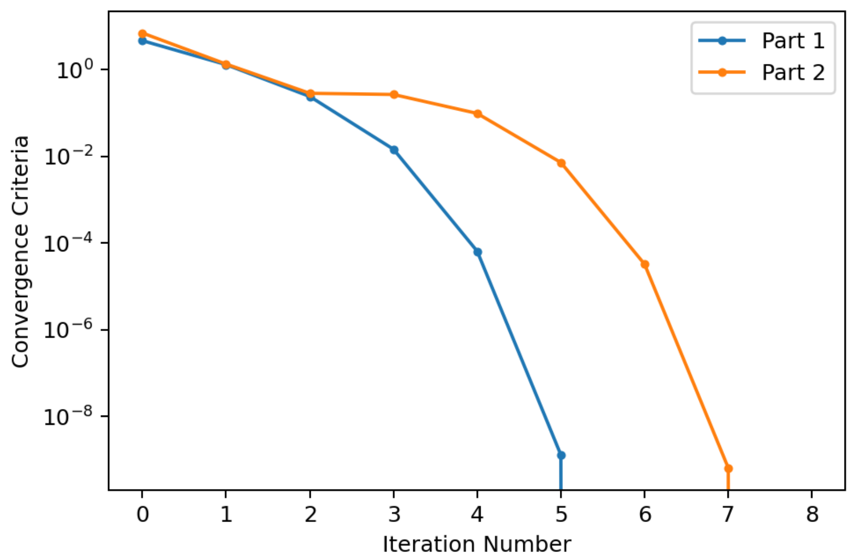

Figure 2: If they plot on a log scale it should follow a trend similar to this

Solutions:

$$\begin{cases} N_{\varepsilon_y} = 7 (\text{this may vary depending on the h used in finite differences}) \\ \varepsilon_x = |x_{k+1} - x_k| \approx 7.2141 \times 10^{-8} \\ N_{\varepsilon_x} = 9 (\text{may also vary, but will always be larger than part 1's solution}) \end{cases}$$

2. Gaussian Elimination (15 pts) By-hand. Use Gaussian Elimination with scaled partial pivoting to solve the following system of equations:

$$\begin{cases} x_1 + 3x_2 + x_3 = -1 \\ 2x_1 + 2x_2 - 6x_3 = 2 \\ 3x_1 - x_2 + 2x_3 = 3 \end{cases}$$

Solution:
The system is first expressed as an augmented matrix:

$$\begin{bmatrix} 2 & 1 & -1 & -2 \\ 4 & 1 & 2 & 4 \\ 6 & 1 & 1 & 6 \end{bmatrix}$$

Forward elimination: First, we swap rows 1 and 3:

$$\begin{bmatrix} 6 & 1 & 1 & 6 \\ 4 & 1 & 2 & 4 \\ 2 & 1 & -1 & -2 \end{bmatrix}$$

Multiply row 1 by 1/2 and subtract from row 2. Multiply row 1 by 1/3 and subtract from row 3.

$$\begin{bmatrix} 6 & 1 & 1 & 6 \\ 0 & 1/3 & 4/3 & 0 \\ 0 & 2/3 & -4/3 & -4 \end{bmatrix}$$

Swap rows 2 and 3:

$$\begin{bmatrix} 6 & 1 & 1 & 6 \\ 0 & 2/3 & -4/3 & -4 \\ 0 & 1/3 & 4/3 & 0 \end{bmatrix}$$

Multiply row 2 by 1/2 and subtract from row 3.

$$\begin{bmatrix} 6 & 1 & 1 & 6 \\ 0 & 2/3 & -4/3 & -4 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

Back substitution:

$$\begin{aligned} x_3 &= \frac{2}{2} = 1 \\ x_2 &= \frac{3}{2}(-4 + 4/3) = -4 \\ x_1 &= \frac{1}{6}(6 - 1 + 4) = 1.5 \end{aligned}$$

3. An Aerospace Application (15 pts) Code. The following question brings together elements from multiple methods we've covered so far. An Airbus A320 is flying at Mach 0.7. It's Pitot tubes, shown in Figure **??**, measure the total, or stagnation, pressure

(pressure when the air hits and is stopped by the tube) as well as the static free-stream pressure.

The ratio of total to static pressure is determined to be: $p_0/p = 1.289$. The following equation relates this change in pressure to the inlet Mach number, M, and γ, the heat capacity ratio:

$$\frac{p_0}{p} = \left[1 + \frac{\gamma - 1}{2} M^2\right]^{\gamma/(\gamma-1)}$$

Based on the given pressure ratio and Mach number, determine the value of γ that satisfies this equation to *four significant figures* of accuracy. State what method you used, how many iterations it took and give your final solution estimate. **Hint:** Don't try to solve analytically for γ. It might be helpful to plot the function $f(\gamma)$ over a range of values to help you pick a good initial first guess. Choose from any of the iterative methods learned in class to solve.

Solutions:

(a) First, rearrange the equation so that it equates to zero:

$$f(k) = 0 = \left[1 + \frac{\gamma - 1}{2} M^2\right]^{\gamma/(\gamma-1)} - \frac{p_0}{p}$$

A few different methods are possible to find the root. Taking the derivative of this function with respect to γ is complicated and so Newton's Method is not recommended, so one can try the Secant Method. I ended up using the bisection method to estimate my solution.

Error Tolerance: $\varepsilon = 1 \times 10^{-5}$
Initial Steps: $x_0 = 0,\ x_1 = 0.1$
Given Parameters: $M = 0.7,\ p_0/p = 1.289$

Results from Secant Method:

Number of Iterations = 3

$x_{Newton} = 0.96498$

$f(x_{Secant}) = -2.7110 \times 10^{-9}$

4. **Fixed Point Iteration and Range of Convergence (10 pts) By-hand.** A fixed point is a point where $x = g(x)$. We will use fixed point iteration to find these points and then apply concepts from fixed point iteration in order to determine the range of convergence for root solving methods.

   (a) Given $x^2 - 4x = 6x + 1$ give two functions, $g_1(x)$ and $g_2(x)$, for which we can perform fixed point iteration to solve for the roots of the equation above.

   (b) Compute and draw the range of convergence (if any) on the interval $[-3, 3]$ for both functions, $g_1(x)$ and $g_2(x)$.

Solution:

5. Newton's Method (20 pts) Code. Given the function $f(x) : -x^4 + 2x^3 = e^{-x} - 1$,

(a) Find the root using Newton's method. Use $x_0 = 1.6$ and perform 5 iterations after the initial guess. Save the x_k from each iteration to a vector.

(b) Determine if Newton's Method converges for the range of $x \in [-3, 3]$ by using the convergence check for fixed point iteration. Provide a plot of $|g'(x)|$ to determine the range of convergence.

(c) Use the results from your code to provide a table of the estimated x_k values from your Newton method iterations. Compute the order of convergence (α) and the asymptotic error constant (λ) as accurately as possible using the x_k values you have saved.

Solution:

(a) Applying the Newton Method, we obtain:

$$\begin{cases} f(x) = -2x^3 + x - e^{-x} - 1 \\ f'(x) = -6x^2 + 1 + e^{-x} \\ x_{n+1} = x_k - \dfrac{f(x_k)}{f'(x_k)} \\ x_k = [-2.0000, -1.6406, -1.5372, -1.5283, -1.5282, -1.5282] \end{cases}$$

(b) Newton's Method converges when $x_{n+1} = x_k$ for $n \to \infty$. Therefore, if fixed point iteration converges when applied to Newton's Method, then Newton's Method must also converge. Writing Newton's Method in the recursive form:

$$x_{fpi} = g(x_{fpi}) = x_{fpi} - \frac{f(x_{fpi})}{f'(x_{fpi})}$$

Fixed point iteration converges for the interval $[a, b]$ if $|g'(x)| < 1$ for $x \in [a, b]$.

$$|g'(x)| = \left| \frac{f(x)f''(x)}{(f'(x))^2} \right|$$

$f(x) = -2x^3 + x - e^{-x} - 1$
$f'(x) = -6x^2 + 1 + e^{-x}$
$f''(x) = -12x - e^{-x}$

Plugging this equation into Python and plotting, we can see that $\exists x \in [-3, 3],\ |g'(x)| > 1 \implies$ FPI and the Newton Method are not guaranteed to converge on the interval [-3,3].

(c) The order of convergence and asymptotic error constant can be approximated as:

$$\alpha = \frac{\ln e_5 - \ln e_4}{\ln e_4 - \ln e_3} = 1.9977$$
$$\lambda = \frac{e_5}{e_4^{\alpha}} = 0.7989$$

6. Matrix Operations (15 pts) Code. Given the linear system,

$$A\,\boldsymbol{x} = \begin{bmatrix} 3 & 2 & 7 & -1 & 4 \\ 6 & -2 & 0 & 2 & -2 \\ 4 & 1 & -1 & 2 & 4 \\ 2 & 10 & -6 & -4 & 1 \\ 5 & 3 & -1 & -8 & 3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 3 \\ 5 \\ -6 \\ 3 \end{Bmatrix} = \boldsymbol{b}$$

Develop a code to:

(**a**) Perform the LU Decomposition of A, and compare to `scipy.linalg.lu(A)`.

(**b**) Compute A^{-1} using the Gauss-Jordan Method, and compare to `numpy.linalg.inv(A)`.

(**c**) Provide the solution of the system $A\,\boldsymbol{x} = \boldsymbol{b}$ using either LU Decomposition **or** the Gauss-Jordan Method.

Solution:

(**a**)

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -0.5556 & 0.7704 & 1 & 0 & 0 \\ -0.2222 & 0.2148 & 0.00077 & 1 & 0 \\ -0.8889 & 0.3259 & 0.1121 & -0.2159 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} -9 & 10 & 1 & -4 & 6 \\ 0 & 15 & -4 & -12 & 10 \\ 0 & 0 & 10.6370 & 12.0222 & -5.3704 \\ 0 & 0 & 0 & 4.5968 & 5.2263 \\ 0 & 0 & 0 & 0 & 0.8044 \end{bmatrix}$$

(**b**)

$$A^{-1} = \begin{bmatrix} -0.0171 & 0.1863 & 0.0367 & 0.0644 & -0.0077 \\ 0.1797 & -1.3661 & -0.1864 & -0.7424 & 0.4136 \\ -0.1573 & 2.2250 & 0.2345 & 1.2881 & -0.6544 \\ 0.1591 & -1.4134 & -0.0876 & -0.8305 & 0.3569 \\ -0.1414 & 1.2431 & 0.2684 & 0.7322 & -0.3539 \end{bmatrix}$$

(**c**)

$$x = \begin{bmatrix} 1.5484 & -15.2508 & 24.9680 & -15.9077 & 14.4742 \end{bmatrix}$$

7. Jacobi Iterative Method (10 pts) Code. Starting with the linear system given in Problem 2, transform the problem in the system, $A^*\boldsymbol{x} = \boldsymbol{b}$, by making A^* be a diagonal dominant matrix ($|A^*(i,i)| \geq \sum_{k=1}^{5} \|A(i,k)\|$. Solve the obtained system using the Jacobi method,

$$\boldsymbol{x}_{k+1} = D^{-1}\,\boldsymbol{b} - D^{-1}\,O\,\boldsymbol{x}_k$$

Stop iterating when the 2-norm of $||\boldsymbol{x}_{k+1} - \boldsymbol{x}_k||_2/||\boldsymbol{x}_{k+1}||_2$ is less than 10^{-12}.

(a) Print out the final values for $\boldsymbol{x}_k$, the corresponding 2-norm of the error, and the number of iterations required.

(b) Repeat this exercise with the original matrix, A, and report the number of iterations required. Describe what has changed.

Solutions

(a)

$$x_k = \begin{Bmatrix} -0.3000 \\ 0.4697 \\ 0.1587 \\ 0.0510 \\ -0.0144 \end{Bmatrix}$$

2-norm of error: $\varepsilon = 6.4674 \cdot 10^{-7}$. Number of iterations: $k = 22$.

(b)

$$x_k = \begin{Bmatrix} 2.0351 \\ 2.3689 \\ -7.2781 \\ -1.9688 \\ -inf \end{Bmatrix} \times 10^{307}$$

2-norm of error: $\varepsilon = NaN$. Number of iterations: $k = 504$.