Assignment 2: Practice Problems

Anuj Goyal

October 2025

1

Write a python code to solve the system of linear equations, given below, using Gaussian elimination with partial pivoting:

$$\begin{bmatrix} 1 & -2 & 4 \\ 8 & -3 & 2 \\ -1 & 10 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 4 \end{bmatrix}$$

(a) How many row interchanges are needed?

(b) Solve again but use only three significant digits of precision.

(c) Repeat part (b) without any row interchanges. Do you get the same result?

2

Apply the Newton-Raphson method to equation $x^3 = N$ to derive an iterative algorithm, for getting the cube root of a number (N), as:

$$x_{n+1} = \frac{2}{3}\left(x_n + \frac{N}{2x_n^2}\right) \tag{2.0}$$

(a) Plot the function $f(x)$, for your choice of a positive number N ($\neq 1$).

(b) Write a code to implement the above algorithm and find the cube root of N (your chosen positive number). Use ”relative error” criteria for convergence.

$$\frac{|x_{n+1} - x_n|}{|x_n|} < \epsilon, x_n \neq 0, \tag{2.1}$$

where, ϵ = tolerance = 10^{-7} is the prescribed tolerance.

3

Write a code to calculate the inverse of a matrix. Use your code to determine the inverse of the matrix given as:

$$\begin{bmatrix} 2.75 & -1.72 & 4.64 \\ 1.75 & -2.92 & -1.00 \\ 4.50 & -4.64 & 3.69 \end{bmatrix}$$

4

A) Write a code that returns the condition number κ of a matrix based on the $\|A\|_\infty$ norm (maximum row sum), which is defined as:

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}|, \tag{4.0}$$

The condition number κ is defined as: $\text{cond}(A) = \kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$

Verify your code on a matrix, A=

$$\begin{bmatrix} 1.99 & 2.01 \\ 2.01 & 1.99 \end{bmatrix}$$

B) Here is a system of equations (in augmented Matrix form) that is called "ill-conditioned", meaning that the solution is not easy to get accurately. You see that $x = [1, 1, 1]^T$ is the solution.

$$\begin{bmatrix} 3 & 2 & 4 & 9 \\ 8 & -6 & -8 & -6 \\ -1 & 2 & 3 & 4 \end{bmatrix}$$

(i.) Using double precision, confirm the solution by doing Gaussian elimination.

(ii.) Now, solve rounding to three significant digits. Do you observe the solution is different?

5

Consider an initial value problem, where rate of change of $u(t)$ is given as a non-linear ordinary differential equation:

$$\frac{du}{dt} = f(t, u) = -3e^{-t} - 3u, \tag{5.0}$$

with initial conditions $u = 1.5$ at $t = 0$.

Numerically integrate $u(t)$ in the time interval $t = [0, 5]$ using three different explicit schemes: (i) Euler scheme, (ii) Modified Euler, (iii) Runge-Kutta scheme.

(a) Plot and compare the solutions from the above schemes with the exact solution of Eq.(1.0) for different choices of time steps. Starting from a relatively large time step, say $dt = 0.5$, up to some small time step $dt = 0.01$.

(b) Plot the error (with respect to the exact solution) for the above schemes as a function of different time steps.

(c) Which scheme results in the lowest error and why?

6

Solve for the position $x(t)$ and momentum $p(t)$ of a simple harmonic oscillator, whose equation of motion is given as:

$$m\frac{d^2x}{dt^2} = -kx, \tag{6.0}$$

Using:

(i) Forward Euler (Explicit)

(ii) Backward Euler (Implicit)

(iii) Backward in position and forward in velocity (Modified scheme)

Assume, $x(t = 0) = 0$; $v(t = 0) = 1$; $m = k = 1$.

(a) Plot momentum $p(t)$ vs. position $x(t)$ for the respective schemes and compare it with the exact solution.

(b) Plot the total energy for the respective schemes and compare it with the exact solution. Comment on whether the energy is conserved or not depending on the scheme.

(c) Identify time step Δt range for which explicit and modified schemes are stable and unstable, respectively.

7

A basketball is at height h_0 and is going up with some initial velocity v_0. If the drag force (or friction) due to air is proportional to the momentum of the ball.

(a) Derive the partial differential equation defining the equation of motion of the basketball.

(b) Write a code to solve the equation of motion numerically, using the Verlet algorithm. You can assume that the ball bounces elastically when it hits the ground. Choose a suitable value for timestep. The drag force constant to be some value between $[0, 1]$, say 0.5 per sec. Assume, $h_0 = 20$ m, and $v_0 = 5$ m/s or any suitable value of your choice.

(c) Plot the height and velocity of the ball as a function of time. What is the maximum timestep value you should take to get a stable solution?

(d) Also, estimate the kinetic and potential energy of the ball as a function of time.

(e) Assume that the drag force is zero. Plot the height and velocity as a function of time. Does your code give you the answer you expect? Is your result sensitive to timestep?

8

The edges of a square plate are kept at the temperatures shown below:

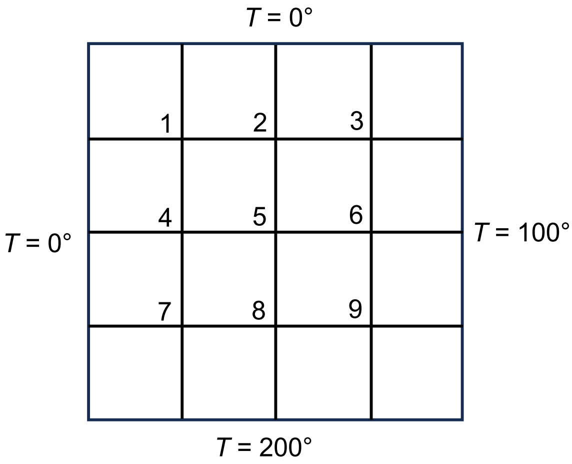

Figure 1:

Assuming steady state heat conduction, the differential equation governing the temperature T in the interior is given as:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, \tag{8.0}$$

(a) Using the finite-difference scheme reduce the above Laplace equation to $AT = B$ form.

(b) Solve for $T_1, T_2, ..., T_9$ using LU and Gauss-Seidel method and compare the solution.

9

Starting with an initial condition given by:

$$T(x,0) = 10\exp(-x^2), \quad \text{for} \quad 0 \leq x \leq 1,$$

solve the heat equation

$$\frac{\partial T}{\partial t} = K\frac{\partial^2 T}{\partial x^2} \tag{9.0}$$

with boundary conditions

$$T(x=0,t) = 25, \tag{9.1}$$

$$\frac{\partial T}{\partial x}(x=1,t) = 0, \tag{9.2}$$

Write down the python code:

(a) Discretize the heat equation using the explicit scheme (Forward in time, Centered in space) **OR** the implicit scheme (Backward in time, Centered in space).

(b) Use $K = 1$ and use two grid spacing $\Delta x = 1/10$, and $\Delta x = 1/100$. In each case, comment on the Fourier number, stability, and accuracy.

10

Let us consider a differential equation:

$$\frac{\partial c(\boldsymbol{r},t)}{\partial t} = \nabla \cdot D \nabla c(\boldsymbol{r},t) \tag{10.0}$$

in a two-dimensional square or rectangular domain with dimensions of your choice. Here, D denotes the diffusivity and c concentration. Consider two forms of diffusivity.

(i.) Constant diffusivity $D = D_0$ where D_0 is a physically meaningful constant of your choice.

(ii.) Diffusivity is a function of position: $D = D_0[(x-x_0)^2+(y-y_0)^2]$ where (x_0, y_0) denotes the coordinates of the center of your computational domain and (x, y) denotes the coordinates of any point in the domain. Note that we choose the lower left corner of the domain as the origin. However, you are free to choose the origin of your domain based on convenience.

For each case, solve the diffusion equation using the finite-difference method and clearly **explain your solutions.**

Given: Initial conditions:

At $t = 0$, $c(r, 0) = 1$ if $(x - x_0)^2 + (y - y_0)^2 \leq R^2$ where R = 20 (say); else $c(r, 0) = 0$.

You can choose any other value for R as long as the circular subdomain lies within the domain and its area is less than 50% of the entire computational domain.

Boundary condition:

Dirichlet boundary conditions on all sides. Choose a meaningful fixed value of c_0 on all edges of the computational domain with $c_{\text{left}} = c_{\text{right}} = c_{\text{bottom}} = c_{\text{top}} = c_0$.