In this sheet: problems on finite differencing, linear equations and matrices.

— — — — — — — — — — — — — — — ⋆ — — — — — — — — — — — — — —

1. Consider the spatial discretization with distance h between neighboring points.

   (a) A symmetric finite difference approximation for the second derivative is
   $$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}$$
   Show that the error in this approximation is of order h^2. You can do this by Taylor expanding $f(x \pm h)$ around x.

   (b) The forward difference approximation for the first derivative is
   $$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$
   Show that the error is linear in h. Define also the backward difference approximation for $f'(x)$ and find the order of the error in that case.

   (c) A better approximation for the first derivative is the centred difference
   $$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$
   The reason this is a better approximation? The error is of quadratic order in h. Show.

   (d) The forward difference formula
   $$\frac{4f(x+h) + f(x+2h) - nf(x)}{2h}$$
   is an approximation to the first derivative $f'(x)$ with $O(h^2)$ error. Find the integer n.

2. Consider the ordinary differential equation

$$\frac{d^2y}{dx^2} - x\frac{dy}{dx} + 5y \;=\; x + 1$$

defined on the interval $x \in [0, 1]$, satisfying boundary conditions

$$y(0) = 4, \quad y(1) = 3.$$

(a) Show how the system can be turned into a set of linear equations, by discretizing the interval $x \in [0, 1]$ into N equal pieces.

Present the equations that are obtained through this procedure. Please define any introduced notation clearly.

Important!! What is the size of the matrix? Is it $N \times N$? Is it $(N + 1) \times (N + 1)$? Or is it $(N - 1) \times (N - 1)$? Explain very clearly why.

(b) Considering the case $N = 5$ where the grid contains only four interior points (0.2, 0.4, 0.6, 0.8), write down the discretized equations in the form of a matrix equation. Write all elements explicitly.

(c) For arbitrary N, write a program that creates the matrix and the constant vector corresponding to the linear equation above. Solve this by explicit matrix inversion, for $N = 10$, $N = 50$ and $N = 100$. Plot your approximate solutions for $y(x)$, $x \in [0, 1]$.

(d) Imagine that the left boundary condition was replaced by a Neumann boundary condition: $y(0)$ is not known and $y'(0) = 1$.

i. Explain how this can be incorporated into your equations using a forward difference approximation for the first derivative, $y'(x)$. How does your first equation change if you use the linear-order approximation, $\frac{1}{h}\big[y(x + h) - y(x)\big]$?
How does your first equation change if you use the quadratic approximation, $\frac{1}{2h}\big[4y(x+h) - y(x+2) - ny(x)\big]$? ($n$ is an integer you were asked to find in an earlier question.)

ii. If you use the centred second-order approximation for $y'(x)$, how does your set of linear equations get modified?

iii. Write a program that constructs the matrix and constant vector for the linear set of equations you have for this modified boundary condition. Solve by matrix inversion, or by any other method provided by numpy/scipy, for various values of N. Plot your approximate solutions for $y(x)$, $x \in [0, 1]$.

3. Consider the system of equations

$$5x_1 - x_2 = -2$$
$$2x_1 + 7x_2 = 1$$

(a) Set up the iteration equations for Jacobi iteration.

(b) Program the iteration process to obtain the solution.

At each step (say at the n-th iteration), measure the Euclidean distance between the current values $(x_1^{(n)}, x_2^{(n)})$ and the exact solution (x_1^*, x_2^*). You can calculate the exact solution in some other way, e.g., by hand.

Plot the Euclidean distnce from the exact solution

$$d_n = \sqrt{\left(x_1^{(n)} - x_1^*\right)^2 + \left(x_2^{(n)} - x_2^*\right)^2}$$

against the iteration number, n.

(c) Program the iteration process to continue until d_n is smaller than some tolerance.

How many iterations are required to reach a tolerance of 10^{-4}?

A tolerance of 10^{-6}, 10^{-8}, 10^{-10}?

(d) Modify the iteration equations for Gauss-Seidel iteration.

Program the Gauss-Seidel iteration and plot d_n versus n. Is the convergence faster?

(e) Imagine that the system of equations is written as

$$2x_1 + 7x_2 = 1$$
$$5x_1 - x_2 = -2$$

Explain why Jacobi iteration will not converge. Write a program for the iteration and plot d_n versus n to show that it does not converge.

4. Krylov subspace.

(a) Given a large $N \times N$ matrix A and a $N \times 1$ vector $\mathbf{b}$, explain how you can build a K-dimensional Krylov subspace, where $K < N$.

(b) Explain how you can use Gram-Schmidt orthogonalization to construct an orthonormal basis for your Krylov subspace.

5. Gaussian elimination.

   (a) Given a system of N linear equations in N variables, count the number of operations required for Gaussian elimination, and the number of operations required for back-substitution. (Big-O notation is enough.)

   (b) Explain how Gaussian elimination can be reformulated in terms of LU decomposition. What are the operation counts for the different steps, in the LU language?

6. Describe how all eigenvalues of a matrix can be obtained by repeated use of the QR algorithm.