Assignment 9 (*revised*)

Due Wednesday 29 November 2023, at the start of class

Please read Lectures 17,20,21,22,23 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. (We are skipping Lectures 18 and 19.) This Assignment primarily covers Gauss elimination and pivoting.

DO THE FOLLOWING EXERCISES from Lecture 17:

- **Exercise 17.2** *Hint. Exactly what does Theorem 17.1 and Theorem 15.1 imply . . .*

DO THE FOLLOWING EXERCISES from Lecture 20:

- **Exercise 20.3** *Do part* (a) *only.*
- **Exercise 20.4**

DO THE FOLLOWING EXERCISES from Lecture 21:[1]

- **Exercise 21.1**

DO THE FOLLOWING EXERCISES from Lecture 22:

- **Exercise 22.1**

DO THE FOLLOWING ADDITIONAL EXERCISES.

P18. This question requires nothing but calculus as a prerequisite. It shows a major source of linear systems from applications.

(a) Consider these three equations, chosen for visualizability:

$$\begin{aligned} x^2 + y^2 + z^2 &= 4 \\ y &= \cos(\pi z) \\ x &= z^2 \end{aligned}$$

Sketch each equation individually as a surface in $\mathbb{R}^3$. (Do this by hand or in MATLAB. Accuracy is not important. The goal is to have a clear mental image of a nonlinear system as a set of intersecting surfaces.) Considering where all three surfaces intersect, describe informally why there are two solutions, that is, two points $(x, y, z) \in \mathbb{R}^3$ at which all three equations are satisfied. Explain why both solutions are inside the closed box $0 \le x \le 2, -1 \le y \le 1, -2 \le z \le 2$.

(b) Newton's method for a system of nonlinear equations is an iterative, approximate, and sometimes very fast, method for solving systems like the one above.

[1] Note that Exercise 21.5 has been removed from this revised version. Feel free to tell me how to do it, because I do not know!

Let $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$. Suppose there are three scalar functions $f_i(\mathbf{x})$ forming a (column) vector function $\mathbf{f}(\mathbf{x}) = (f_1, f_2, f_3)$, and consider the system

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}.$$

(It is easy to put the part **(a)** system in this form.) Let

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

be the Jacobian matrix: $J \in \mathbb{R}^{3\times 3}$. The Jacobian generally depends on location, i.e. $J = J(\mathbf{x})$, and it generalizes the ordinary scalar derivative.

Newton's method itself is

$$J(\mathbf{x}_n)\,\mathbf{s} = -\mathbf{f}(\mathbf{x}_n), \tag{1}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s} \tag{2}$$

where $\mathbf{s} = (s_1, s_2, s_3)$ is the *step* and $\mathbf{x}_0$ is an initial iterate. Equation (1) is a system of linear equations which determines s, and then equation (2) moves to the next iterate.

Using $\mathbf{x}_0 = (1, -1, 1)$, write out equation (1) in the $n = 0$ case, for the problem in part **(a)**, as a concrete linear system of three equations for the three unknown components of the step $\mathbf{s} = (s_1, s_2, s_3)$.

(c) Implement Newton's method in MATLAB to solve the part **(a)** nonlinear system. Show your script and generate at least five iterations. Use $\mathbf{x}_0 = (1, -1, 1)$ as an initial iterate to find one solution, and also find the other solution using a different initial iterate. Note that `format long` is appropriate here for showing iterates.

(d) In calculus you likely learned Newton's method as a memorized formula, $x_{n+1} = x_n - f(x_n)/f'(x_n)$. Rewrite equations (1), (2) for $\mathbb{R}^1$ to derive this formula.