Assignment # 2

Due Monday 9 February at start of class

Note "MOP" = MATLAB|OCTAVE|PYLAB, *from now on in this course.*

Exercise 2.1. (i.e., page 15, Lecture 2 in TREF & BAU)

Exercise 2.2.

Exercise 2.3.

P5. Read *Appendix: The Definition of Numerical Analysis*, page 321 of TREF & BAU.

Now answer a single question, formulated exclusively as encouragement to actually read it. Complete this statement of a "general principle of computing",

the faster the computer, the ________________________________

P6. On page 21 of TREF & BAU, equation (3.10) gives a formula for the ∞-norm of an $m \times n$ matrix. Prove it.

P7. *This question requires nothing but calculus as a prerequisite. Its purpose is twofold: (i) To illustrate very fast convergence of an iterative method—"quadratic convergence". We will see such excellent performance again in linear algebra problems in Lecture 25, 27, 28, . . . , though convergence this fast is rare in some sense. (ii) To illustrate a major source of linear systems in the science/engineering world.*

(a) Consider Newton's method for finding a solution ("root") of a single real equation in a single real unknown, $f(x) = 0$. Newton's method is the iteration which takes x_n, linearizes f there, finds the root x_{n+1} of the linearization, and then repeats. Use this description to derive the standard formula,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

(b) Check that $\tilde{x} = 0.739085133215161$ must be very close to an exact solution of the equation $\cos x = x$. Now use Newton's method, in MOP, to find $x_1, \dots, x_5$ if $x_0 = 1$. Describe how the error $e_n = |x_n - \tilde{x}|$ descreases. Illustrate that "$e_{n+1} \approx \lambda e_n$" does not capture the decay rate for any reasonable positive λ, but instead that $e_{n+1} \approx C e_n^2$; the right kind of plot shows this but so will the right table of numbers. (*This is "quadratic convergence". It is* so *fast that mere 16 decimal digit precision makes it hard to demonstrate.*)

CONTINUATION OF **P7**:

(c) Consider these three equations, chosen for pedagogical convenience:

$$x^2 + y^2 + z^2 = 4,$$
$$z = \sin(2\pi x),$$
$$y = x^2.$$

Convince yourself informally that there are two solutions, that is, exactly two points $(x, y, z) \in \mathbb{R}^3$ at which all three equations are satisfied. For example, you might attempt a sketch of each equation individually, as a surface in $\mathbb{R}^3$, and consider where all three surfaces intersect. Explain why such simultaneous solutions are, in any case, inside the box $-2 \le x \le 2, 0 \le y \le 2, -1 \le z \le 1$.

(d) Now, Newton's method for a system of three nonlinear equations. Let $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$ and describe three scalar functions as the vector function $\mathbf{f}(\mathbf{x}) = (f_1, f_2, f_3)$. For example, one can write the the system of equations above as $\mathbf{f}(\mathbf{x}) = 0$ with

$$\mathbf{f}(\mathbf{x}) = (x_1^2 + x_2^2 + x_3^2 - 4, \quad \sin(2\pi x_1) - x_3, \quad x_1^2 - x_2).$$

Let

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

be the Jacobian matrix, 3×3 in this case, which is a function of the location: $J(\mathbf{x})$. The Jacobian matrix $J(\mathbf{x})$ generalizes the ordinary scalar derivative $f'(x)$ in part (a). Newton's method (also "Newton-Raphson") is conveniently written as

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

where $\mathbf{s} = (s_1, s_2, s_3)$ is the *step*. The first of these equations is a system of three linear equations in three unknowns which determines $\mathbf{s}$, while the second uses $\mathbf{s}$ to move (step) to the next iterate. Demonstrate how these formulas reduce to the scalar Newton's method in part (a).

(e) As noted, $\mathbf{f}(\mathbf{x}) = (x_1^2 + x_2^2 + x_3^2 - 4, \sin(2\pi x_1) - x_3, x_1^2 - x_2)$ in part (c) above. Implement Newton's method to solve the system in part (c). You can do this at the command line in MOP, or in a script. Use $\mathbf{x}_0 = (1, 1, 1)$ as an initial iterate.