Assignment #2

Due Monday 21 September, 2015 at the start of class

Please read Lectures 2, 3, and 4 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. Then do these exercises:

P4. In your undergraduate linear algebra class, or elsewhere, you learned a method for computing determinants called "expansion by minors." Compute this determinant by hand to demonstrate that you know this algorithm:

$$\det\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right).$$

Check your work in MATLAB.

Now, for an arbitrary $A \in \mathbb{C}^{m\times m}$, count the exact number of multiplication operations needed to compute $\det(A)$ by this method. (*Hint*: How much more work is the $m \times m$ case than the $(m-1)\times(m-1)$ case?)

P5. *This question requires nothing but calculus as a prerequisite. Its purpose is to show a major source of linear systems from the science/engineering/mathematics world.*

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

$$\begin{aligned} x^2 + y^2 + z^2 &= 4, \\ \sin(2\pi y) - z &= 0, \\ x &= y^2. \end{aligned}$$

Sketch each equation individually as a surface in $\mathbb{R}^3$ (by hand or in MATLAB). 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 such solutions are inside the box $0 \le x \le 2, -2 \le y \le 2, -1 \le z \le 1$.

(b) Newton's method for a system of nonlinear equations is an iterative, approximate, and very fast (*when it works ...*) method for solving systems like that in part **(a)**. Considering $\mathbb{R}^3$ cases specifically, let $\mathbf{x} = (x_1, x_2, x_3) \in \mathbb{R}^3$. Suppose there are three scalar functions $f_i(x_1, x_2, x_3)$ forming a vector function $\mathbf{f}(\mathbf{x}) = (f_1, f_2, f_3)$. Let

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

be the Jacobian matrix: $J \in \mathbb{R}^{3\times 3}$. The Jacobian matrix is a nontrivial function of the location if the equations are nonlinear: $J = J(\mathbf{x})$. The Jacobian matrix function $J(\mathbf{x})$ generalizes the ordinary scalar derivative $f'(x)$.

Now, 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 three linear equations in three unknowns, in our case, which determines $\mathbf{s}$. The second equation (2) uses the step $\mathbf{s}$ to move to the next iterate; "$\delta\mathbf{x}$" is another common notation for $\mathbf{s}$.

In part **(a)** we have $\mathbf{f}(\mathbf{x}) = (x_1^2+x_2^2+x_3^2-4, \sin(2\pi x_2)-x_3, x_1-x_2^2)$. Using $\mathbf{x}_0 = (1,1,1)$, write out equation (1) in the $n=0$ case, 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 nonlinear system in part **(a)**. You can do this at the command line or in a script, but show me inputs (functions!), the commands/script, and at least three iterations. The command `format long` is appropriate here. Use $\mathbf{x}_0 = (1,1,1)$ as an initial iterate.

(d) In calculus you probably learned Newton's method as a memorized formula,

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

Rewrite this scalar case in the form of equations (1), (2), clearly identifying the Jacobian and the step, and what kind of objects they are.

P6. Write a MATLAB program which draws the unit balls shown in (3.2) on page 18 of Trefethen & Bau. That is, draw clean pictures of the unit balls of $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$, and $\|\cdot\|_p$. Following the aesthetic advice on page 18, use $p=4$ for the last one.

Exercise 1.4 in Lecture 1.

Exercise 2.1 in Lecture 2.

Exercise 2.3 in Lecture 2.