Assignment #5

Due Friday 8 March, 2013

Please read Lectures 6, 7, 8, 9, and 10 in the textbook *Numerical Linear Algebra* by Trefethen and Bau. Do these exercises:

P10. Apply the classical Gram-Schmidt process by-hand to the columns $\{a_1, a_2, a_3\}$ of

$$A = \begin{bmatrix} -1 & 1 & 1 \\ 2 & 0 & 1 \\ 0 & 1 & -2 \\ 1 & 1 & 1 \end{bmatrix} \in \mathbb{C}^{4\times 3}$$

and write the result as $A = \hat{Q}\hat{R}$ where $\hat{Q} \in \mathbb{C}^{4\times 3}$ and $\hat{R} \in \mathbb{C}^{3\times 3}$. Compare to the result of this line of MATLAB:

```
>> [Qhat, Rhat]= qr(A,0)
```

P11. *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:

$$x^2 + y^2 + z^2 = 4,$$
$$\sin(2\pi y) - z = 0,$$
$$x = y^2.$$

Sketch of 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 (also "Newton-Raphson") 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 $\mathbf{f}(\mathbf{x}) = (f_1, f_2, f_3)$. (For example, one can write the the system of equations in part **(a)** as $\mathbf{f}(\mathbf{x}) = 0$ with $\mathbf{f}(\mathbf{x}) = (x_1^2 + x_2^2 + x_3^2 - 4, \sin(2\pi x_2) - x_3, x_1 - x_2^2)$.) Let

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

be the Jacobian matrix, with $J \in \mathbb{R}^{3\times 3}$ in this case. The Jacobian is a nontrivial function of the location if the equations are nonlinear: $J = J(\mathbf{x})$. The Jacobian matrix $J(\mathbf{x})$ generalizes the ordinary scalar derivative $f'(x)$ (see part **(d)** below). Newton's method 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 s $= (s_1, s_2, s_3)$ is the *step* and $\mathbf{x}_0$ is an initial iterate. The inputs to the method are the functions to solve (the "residual") $\mathbf{f}(\mathbf{x})$ and the initial iterate $\mathbf{x}_0$. Equation (1) is a system of three linear equations in three unknowns, in our case, which determines s. The second equation (2) uses the step s to move to the next iterate; "δx" is common notation for s.

As noted, 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 $\mathbf{x}_n = (1, 1, 1)$ as a known iterate, write out equation (1) in this case, as a linear system for the unknowns s $= (s_1, s_2, s_3)$. It will be a concrete linear system of three equations in three unknowns.

(c) Implement Newton's method in MATLAB to solve the 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 memorizable 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.

Exercise 6.1 in Lecture 6.

Exercise 6.3 in Lecture 6.

Exercise 6.4 in Lecture 6.

Exercise 7.1 in Lecture 7.

Exercise 7.2 in Lecture 7.