MATH 4093/5093 **Homework 6** **Due Fri, 4/1/2011**

Problem 1. Consider the linear system

$$\begin{aligned} 2x_1 - x_2 &= -1 \\ -x_1 + 4x_2 + 2x_3 &= 3 \\ 2x_2 + 6x_3 &= 5 \end{aligned}$$

(a) Write the system in the form $A\mathbf{x} = \mathbf{b}$.

(b) Starting with the initial guess $\mathbf{x}^{(0)} = \mathbf{0}$, perform two iterations of the Jacobi method.

Problem 2. Consider the matrix $A = \begin{pmatrix} 2 & -1 \\ -1 & 3 \end{pmatrix}$.

(a) Find the Jacobi and Gauss-Seidel iteration matrices T_{Jac} and T_{GS}.

(b) Determine the spectral radius of each iteration matrix from (a).

(c) Will the Jacobi method converge for any choice of initial vector $\mathbf{x}^{(0)}$? Will the Gauss-Seidel method converge for any choice of initial vector $\mathbf{x}^{(0)}$? Explain.

Problem 3. Download the codes `jacobi.m`, `gauss_seidel.m`, and `sor.m` from the class web-site. They solve linear systems iteratively, using the Jacobi, Gauss-Seidel, or the SOR (successive over-relaxation) methods.

In this problem you will test these codes on the system $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 5 & 1 & 2 \\ -3 & 9 & 4 \\ 1 & 2 & -7 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 10 \\ -14 \\ -33 \end{pmatrix}.$$

In each case, start from the initial vector $\mathbf{x}^{(0)} = (0 \;\; 0 \;\; 0)^T$, use tolerance 10^{-10} (i.e., take `TOL=1e-10`), and allow 200 steps to be performed before the program exits with an error message. The arguments `b` and `xold` passed to the programs should be row vectors.

(a) Run the code `jacobi.m` and record how many steps are needed to achieve the desired accuracy (`TOL=1e-10`). Attach a printout (use `format long` to see more digits).

(b) Run the code `gauss_seidel.m` and record how many steps are needed to achieve the desired accuracy. No printout is necessary.

(c) Run the code `sor.m` with values of the parameter ω equal to 0.1, 0.2, 0.3, ..., 1.3, and in each case record the number of steps needed to achieve the desired accuracy. No printout is necessary.

Problem 4. This problem is about the concept of multiplicity of a zero of a function.

(a) The Taylor expansion of a function $\phi(x)$ around $x_0 = 3$ be

$$\phi(x) = \frac{1}{2}(x-3)^2 - \frac{1}{3}(x-3)^4 + \frac{1}{3}(x-3)^6 - \frac{23}{90}(x-3)^8 + \frac{181}{720}(x-3)^{10} \cdots .$$

What is the multiplicity m of 3 as a zero of $\phi(x)$? Explain why.

Hint: The easiest way to do this is to use the very definition of multiplicity.

(b) Directly from the definition of multiplicity of zero, show that if p is a zero of multiplicity m_1 of the function $f(x)$, and a zero of multiplicity m_2 of the function $g(x)$, then it is zero of multiplicity $m_1 + m_2$ of the product $f(x)\,g(x)$.

(c) Show that $p = 0$ is a zero of multiplicity 3 of the function $f(x) = x - \sin x$.

(d) Use your results from parts (b) and (c) to find the multiplicity of the zero $p = 0$ of the function

$$h(x) = x^4\,(x - \sin x)^2 \ .$$

Problem 5. Consider the system of equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, where $\mathbf{x} = \binom{x_1}{x_2}$, and

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 - x_2 \\ -3x_1 + 2x_2 - 1 \end{pmatrix}$$

and perform "by hand" two steps of the Newton's method starting from the initial guess $\mathbf{x}^{(0)} = \binom{0}{0}$. What do you observe? Why do you see this behavior? For what functions $\mathbf{f}(\mathbf{x})$ would you observe the same behavior?

Problem 6. Write the nonlinear system

$$\begin{aligned} x_1^3 - 2x_2 &= 2 \\ x_1^3 - 5x_3^2 &= -7 \\ x_2 x_3^2 &= 1 \end{aligned}$$

in the form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. The exact solution of this system is $\mathbf{x}_{\text{exact}} = \left(\sqrt[3]{3} \quad 1/2 \quad \sqrt{2}\right)^T$. Compute the Jacobian $J(\mathbf{x}) = \left(\frac{\partial f_i}{\partial x_j}(\mathbf{x})\right)_{i,j}$. Create the files `sys.m` and `sys_jac.m` that take as a argument a 3-dimensional column vector $\mathbf{x}$ and return $\mathbf{f}(\mathbf{x})$ and the Jacobian $J(\mathbf{x})$. Start from the vector $\mathbf{x}^{(0)} = (1 \quad 1 \quad 1)^T$ and perform several steps of the Newton's method "manually" in MATLAB, monitoring the error $\|\mathbf{x}^{(n)} - \mathbf{x}_{\text{exact}}\|_\infty$ by executing the line

```
x = x - inv(sys_jac(x))*sys(x);      norm(x-xexact,inf)
```

several times. Attach printouts of the MATLAB functions `sys.m` and `sys_jac.m` and a printout of your MATLAB session with the steps of the Newton method.