CS 357 / MATH 357
Fall 2013

Numerical Methods – Homework 3
Instructor: B. David Semeraro
University of Illinois at Urbana-Champaign

Due: 09/26/2013

Note: Homework is due **5pm** on the due date. Please submit your homework through the dropbox in the Siebel Center basement. Make sure to include your name and **netid** in your homework.

Problem 1 [**6pt**] Considering long operations only and assuming 1 microsecond execution time for all long operations, give the costs for solving $A\boldsymbol{x} = \boldsymbol{b}$ when $n = 10^3, 10^4, 10^5, 10^6$ with the following methods. Estimate the costs at 50 cents per minute and round to the nearest cent.

(a) Gaussian elimination with scaled partial pivoting.

(b) Tridiagonal (`Tri` procedure in book).

Problem 2 [**4pt**] True/False questions

(a) **(True/False)** Partial pivoting is especially useful for strictly diagonally dominant tridiagonal matrices.

(b) **(True/False)** A small residual guarantees an accurate solution.

(c) **(True/False)** A 638x638 tridiagonal matrix requires 1912 memory locations.

(d) **(True/False)** In scaled partial pivoting, the matrix is scaled to eliminate small elements.

Problem 3 [**10pt**] Consider the following system:

$$\begin{bmatrix} 2.130 & 6.325 & 10.11 \\ 1.100 & 3.912 & 3.351 \\ 3.345 & 14.18 & 28.33 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.000 \\ 2.322 \\ -8.500 \end{bmatrix}$$

(a) Solve this system using Gaussian elimination with scaled partial pivoting. At each step show the values of the index array and retain 4 significant figures for each calculation.

(b) Calculate the residual.

(c) Calculate the error using `numpy.linalg.solve` to compute the exact solution.

(d) Use `numpy.linalg.cond` to compute the condition number of this matrix.

Problem 4 [**20pt**] We wish to compare the time required to perform the level 1, level 2, and level 3 `Blas` operations as a function of size n. Where level 1 is the vector inner product, level 2 is the matrix vector product, and level three is the matrix matrix product. Write a Python program that times each of the three `Blas` methods for 10 different vector and matrix sizes n. For each of the three operators assume any matrix involved is square. Use the Python `time.clock()` to time the operations. Use the Python routine `random.rand(n)` to initialize the vector and matrix elements.

(a) Create a table of values of time vs n for each method. Hand written tables are fine.

(b) Use Matplotlib to plot the times as a function of n using one curve for each method. (ie. 3 curves on one plot)

(c) Turn in your printed code listing, the table, and the printed matplotlib output.