SOLUTIONS OF LINEAR SYSTEMS

Programming Assignment 6 (20 points)

Math 248 Computers and Numerical Algorithms–Spring 2007–Pruett

DATE ASSIGNED: Tuesday, 27 March, 2007 **DATE DUE:** Thursday, 05 April, 2007

BACKGROUND: Later this semester (and in Math 449), you will learn that matrix methods are sometimes used to solve ordinary differential equations (ODE's) of boundary-value (BV) type. The matrices that arise in these applications are usually *banded*. Banded matrices have the property that the only non-zero elements lie on the diagonal or in close proximity to the diagonal. For example, *tridiagonal* matrices have the form:

$$\begin{bmatrix} d_1 & u_1 & 0 & 0 & \dots & 0 \\ l_2 & d_2 & u_2 & 0 & \dots & 0 \\ 0 & l_3 & d_3 & u_3 & \dots & 0 \\ . & . & . & . & . & . \\ 0 & \dots & 0 & l_{n-1} & d_{n-1} & u_{n-1} \\ 0 & \dots & 0 & 0 & l_n & d_n \end{bmatrix}$$

In class we adapted Gaussian elimination (Algorithms 6.2-3) to tridiagonal systems of equations (Alg. 6.5).

1. **C-level ASSIGNMENT:** Write a SUBROUTINE that can solve $T\vec{x} = \vec{b}$ by Alg. 6.5, where T is a tridiagonal $n \times n$ matrix, and $\vec{x}$ and $\vec{b}$ are n-vectors.

   (a) Design your subroutine to use the following 5 arguments: an integer scalar giving the size of the system and 4 n-vectors containing the elements of the subdiagonal, the diagonal, the superdiagonal, and the right-hand-side vector, respectively. To save memory, if you wish, the solution can can replace the RHS vector upon return.

   (b) Write a main program that will call your subroutine to solve $T\vec{x} = \vec{b}$ for data provided in file **proj6.dat** on the **K:** drive (directory **COSM**, subdirectory **Math-Common**, and sub-subdirectory **Math248**). Use *compile-time* array declarations of fixed dimensions 256.

2. **B-level ASSIGNMENT:** *Do the C-level part first.* (To encourage a stepwise approach, I will give B-level help only to those students who have done the C-level part correctly!)

   (a) Modify your program above to use *run-time* array declarations and the ALLOCATE and DEALLOCATE commands. Your program should read the value of n and allocate memory accordingly.

   (b) Determine the operation count for your tridiagonal solver algorithm.

3. **A-level ASSIGNMENT:** (Quantum jump beyond B-level). In addition to the B-level assignment ...

   (a) Compute and print out T^{-1}. Is T^{-1} also tridiagonal? How many operations are required to compute the inverse?

   (b) Compute $\vec{x} = T^{-1}\vec{b}$; that is, find the solution to the linear system by first computing the inverse and then multiplying the RHS by the inverse. (HINT: You can adapt your original algorithm readily to compute the inverse of T.)

   (c) Compare the approximate total number of operations of the two methods for solving the linear system: 1) Gaussian elimination for triadiagonal matrices, vs. 2) matrix inversion plus matrix multiplication.

   (d) Mathematically the two methods give identical results. Are the computational results identical? Let $\vec{x}_1$ and $\vec{x}_2$ be the solutions computed by the two methods. Find $||\vec{x}_1 - \vec{x}_2||$.

   (e) Draw some conclusions about which method is prefered.

SPECIAL CONSIDERATIONS: The sub- and super-diagonals have one fewer elements than the diagonal. There are several ways to address this problem. I recommend the simplest "fix" that works.

This project is to be done alone or with one partner. As always, you are allowed to discuss programming issues with other students, but the actual coding of your program should be accomplished individually or with only your partner, and your program should be unique.