Announcements

• The first examination is Wednesday, February 15 in class (from 12:30 pm to 1:20 pm). It is a closed book exam, and only pens, pencils, and erasers are permitted. It covers Chapters 1 and 2 of the class notes. Questions are based on Assignments # 1 and # 2 and the “Review Questions” for Chapters 1 and 2 in the class notes. (It does not explicitly cover unassigned exercises.)

• Here in writing is what was announced verbally in class:
  
  – *Cheating.* “Evident collaboration is penalized in accordance with university rules and the instructor’s policy is to impose substantial penalties.” In practice this is enforced when the course staff believe that the collaboration goes beyond the sharing of ideas. The course staff reserve the exclusive right to make the judgment of what constitutes excessive collaboration. Students who collaborate can avoid violating the rule against cheating by including a written acknowledgment in their writeups, which is nothing more than good scholarship.
  
  – *Copying.* In the case of excessive acknowledged collaboration or excessive use of outside written material, the score for an assignment may be discounted. For example, 3 students turning in identical correct answers each may receive as little as one third credit. In the case of the use of outside written material, reduction in credit can be avoided by acknowledging the source, which is again proper scholarship. As stated previously, for problems from the textbook you may use facts presented in the textbook and for other problems you may use the class notes.

• Some general rules concerning computer programs: If provided by the programming language, the use of array addition and multiplication is encouraged. This includes sections of arrays. However, the use of more advanced operations such as matrix factorization, inversion, or division is generally not permitted.

  – If using C, do not use the features of C++; however, use of array features of C99 is encouraged if implemented by the compiler.

  – If using Python, you should install and use NumPy from [http://numeric.scipy.org/](http://numeric.scipy.org/) or one of its predecessors Numeric or Numarray.

Due Friday, February 10 in class

0. Look over Sections 1.5–2.5 of the class notes and Section 1.11 of the textbook.

1. (15 points) Write a recursive program in Matlab, Python or C that inverts a nonsingular lower triangular matrix. The inverse of an $n$ by $n$ matrix should be computed in terms of the
inverse of a $n - 1$ by $n - 1$ matrix unless $n = 1$, in which case it should be computed directly. The program should accept a two-dimensional array and return the inverse. You may assume valid input. Try your program on the 8 by 8 matrix

$$\begin{bmatrix}
1 & h^2 - 1 & 1 \\
\frac{h^2}{2} - 1 & h^2 - 2 & 1 \\
1 & h^2 - 2 & 1 \\
& & & \ddots & \ddots & \ddots \\
1 & h^2 - 2 & 1
\end{bmatrix}.$$  

First try it with $h = \frac{1}{2}$. Then try it on the inverse that you just computed to see if you get the original matrix back again. Finally try it with $h = 0$. (The above matrix arises in the discretization of the dynamics of a spring by an eighteenth century integrator known variously as the Störmer method, the leapfrog method, and the Verlet method.) Hand in a printed copy of your program and its output.

2. (4 points) (Outer product form of Cholesky factorization) By considering $(n-1,1)$ by $(n-1,1)$ partitionings of the matrices involved, give a recursive algorithm for the Cholesky factorization $GG^T$ of a symmetric positive definite matrix $A$.

3. (5 points) Show in the case of base ten floating-point arithmetic that if $\frac{1}{2} \leq x/y \leq 2$, the difference $x - y$ is computed exactly. (Your argument should be convincing to a person, but not a mechanical proof checker.) Show that this is untrue if (say) $0.4 \leq x/y \leq 2.5$.

4. (6 points) Suppose that a matrix $A$ has a factorization $A = M_1^{-1}P_2M_2^{-1}U$ where

$$M_1 = \begin{bmatrix}
1 & 0 & 0 \\
-1/2 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad P_2 = \begin{bmatrix}
0 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}, \quad M_2 = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 1/2 & 1
\end{bmatrix}, \quad U = \begin{bmatrix}
2 & -2 & 0 \\
0 & 4 & -2 \\
0 & 0 & 1
\end{bmatrix}.$$  

Without re-forming the original matrix $A$, use this information to solve $A^T x = b$ efficiently where $b = [5 \ 1 \ -4]^T$. 