Math 170A
Midterm 1

Name: Answer Key
Student ID number: 

Instructions: Answers without work may be given no credit at the grader's discretion. The test is out of 36 points.

This cover page may be used at scratch paper. However, all final work must be on the page with the related question. Do not remove this sheet.
1. What are the two main ways we can measure the speed of an algorithm/program? What are their respective strengths and weaknesses? (5 pts)

- **Actual runtime**: adv: time is money, so this is what we really care about.
  - disadv: not a good theoretical tool b/c it depends on implementation, hardware, and even chance, so heavily.

- **Number of operations (flops)**: Good theoretical tool, but it is only a stand in for time.
  - For instance, it can tell us how computation time should roughly scale as we increase the size of our problem.

2. Contrast the Cholesky and LU decompositions by explaining one advantage of each over the other decomposition. (4 pts)

- Cholesky runs in about $\frac{1}{2}$ the time of LU.

- However, LU decomposition works for any nonsingular matrix, while Cholesky only works for sym. pos. def. matrices.
3. Explain the process of how to solve $Ax = b_i$ efficiently, for multiple vectors $b_i$. (I want you to explain what quantities/matrices/vectors you calculate, and the names of the algorithms you use. Assume your matrix is not symmetric positive definite, but it is invertible.) (4 pts)

1. Calculate $LU, P$ (via the LU decomposition w/ partial pivoting) for $A$.

   Then, for each $b_i$

   2. permute the entries of $b_i$, as $P$ says.

   3. Solve $Ly = Pb_i$ via forward subs.

   4. Solve $Ux = y = L^{-1}Pb_i$ via back subs

   That $x$ then solves $Ax = b_i$.

4. Why is the process you described in the previous question more efficient than simply solving $Ax = b_i$ via row reduction? (2 pts)

   While solving for one $b$ would take about the same number of flops, w/ this method, each additional solve for $x_i$ only costs $O(n^2)$ vs. $O(n^3)$ for row reduction, since we can reuse $L, U, P$. 
5. Write a MATLAB function that solves $Lx = b$, for any matrix $L$ that is lower triangular. (Your inputs are $L$ and $b$, your output is $x$.) Your program must check for, and take advantage of, any leading zeros in $b$. (5 pts)

```matlab
function b = forsub(L, b)
K=0;
for i = 1:length(b)
    if b(i) == 0
        K = i;
        break
    else
        K = i;
    end
end
for j = K:length(b)
    for i = K:(j-1)
        b(i) = b(i) - L(i,j) * b(j);
    end
    if L(j,j) == 0
        error ('Matrix is singular')
    end
    b(j) = b(j) / L(j,j);
end
```

- $K$ becomes index of first non-zero entry in $b$.
- Only do for $i > K$, $j > K$ to take advantage of zeros.
- Error checking. You didn’t have to include this.
6. Explain the basic idea of a block version of an algorithm, and why they are useful. (3 pts)

A block version breaks the matrix and calculation into smallish pieces that can each be done independently. This is useful as it can help minimize memory access delay and makes the algorithm parallelizable.

7. If an algorithm is \( O(n^3) \), what does that mean? What happens if we double the size of the matrices? (3 pts)

\( O(n^3) \) means the algorithm takes about \( \text{constant} \cdot n^3 \) flops in order to complete. So, if we double the size, our work will go up by about a factor of 8.
8. Consider the one dimensional differential equation \( u''(x) - 5u'(x) = -1 \) on the interval \([-2, 0]\) and boundary conditions \( u(-2) = 0, u(0) = 0 \). Set up the equation \( Au = b \) to estimate the solution to this differential equation, by splitting the interval \([-2, 0]\) into subintervals of length 1/3. Do not solve the problem once it is set up. (5 pts)

\[
\begin{bmatrix}
  \frac{33}{2} & -18 & 0 & 0 & 0 \\
  -18 & 3/2 & 0 & 0 & 0 \\
  0 & 33/2 & -18 & 3/2 & 0 \\
  0 & 0 & -33/2 & -18 & -\frac{33}{2} \\
  0 & 0 & 0 & 33/2 & -18
\end{bmatrix}
\begin{bmatrix}
  u(-\frac{5}{3}) \\
  u(-\frac{4}{3}) \\
  u(-1) \\
  u(-\frac{3}{2}) \\
  u(-\frac{1}{2})
\end{bmatrix}
= \begin{bmatrix}
  -1 \\
  -1 \\
  -1 \\
  -1 \\
  -1
\end{bmatrix}
\]
9. Consider the matrix \( A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 8 & -2 \\ -1 & -2 & 1 \end{bmatrix} \). Is this matrix positive definite? Justify your answer using a method we have discussed in class. (5 pts)

Using Cholesky:
\[
\begin{align*}
    r_{11}^2 &= 1, & r_{11} &= 1 \\
    r_{12} &= \frac{2}{\sqrt{3}}, & r_{13} &= -\frac{1}{\sqrt{3}} \\
    r_{12}^2 + r_{22}^2 &= 8, & r_{22} &= \sqrt{4} = 2 \\
    r_{13}^2 + r_{23}^2 + r_{33}^2 &= 1 \\
    \Rightarrow r_{33} &= 0
\end{align*}
\]

Main way:
\[
R = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\]

Since a diagonal element is 0, the Cholesky decomposition fails and so \( A \) is not pos. def. (In fact, \( A \) is singular.)

Way 2: By row reduction, \( A \Rightarrow \begin{bmatrix} 1 & 2 & -1 \\ 0 & 4 & -4 \\ 0 & 0 & 0 \end{bmatrix} \) which is singular.

A singular matrix has a vector \( x \neq 0 \) so that \( Ax = 0 \).

Thus \( x^TAx = 0 \), so \( A \) is not pos. def.

Way 3: The vector \( x = [1, 0] \) satisfies \( x^TAx = 0 \), so \( A \) is not pos. def.
This page may be used for scrap paper. ALL FINAL WORK MUST BE ON THE APPROPRIATE PAGE.