Math 441 — Exam I
October 7, 1993

1. Suppose that when floating point arithmetic is performed on a certain machine, $c = 2^{-8}$ is the smallest positive machine representable number for which $2^{15} + c > 2^{15}$. What is machine epsilon on this particular machine? Justify your answer.

---

2. Let

$$A^{(2)} = \begin{bmatrix} 2 & -2 & -4 & 5 \\ 0 & -3 & 5 & -1 \\ 0 & 6 & 2 & 2 \\ 0 & 3 & -1 & 3 \end{bmatrix}.$$

a. Apply one step of Gaussian Elimination **with partial pivoting** to obtain a matrix $A^{(3)}$ with zeros below the first 2 diagonal entries. Use a row interchange strategy which guarantees that **none of the multipliers are larger than 1 in absolute value**.

b. Give the lower triangular matrix M and the permutation matrix P for which $A^{(3)} = MP\,A^{(2)}$.

---

3. The infinite series $\sum_{k=1}^{\infty} \frac{1}{k}$ diverges (i.e., the partial sums $S_n = \sum_{k=1}^{n} \frac{1}{k}$ tend to infinity as $n \to \infty$. However, when I implement the following algorithm in MATLAB on my computer,

$$n = 1; \quad S_{n-1} = 0; \quad S_n = 1;$$

while $S_{n-1} \neq S_n$,

$$n = n + 1;$$
$$S_{n-1} = S_n;$$
$$S_n = S_n + 1/n;$$

end

the algorithm terminates (after running for several days) with finite $n \approx 2.2983 \times 10^{14}$ and a finite sum $S_n \approx 19.5951$.

a. Explain exactly why this happens. Machine epsilon on my machine is $2^{-52} \approx 2.2204 \times 10^{-16}$.

b. What would be the effect of a **smaller** value of machine epsilon? Would the values of n and S_n at termination be larger or smaller? Explain your answer.

---

4. Let

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2} \end{bmatrix}, \quad \text{and } \mathbf{b} = \begin{bmatrix} -3 \\ -5 \\ 8 \end{bmatrix}.$$

Make intelligent use of the decomposition $A = LDL^T$ to solve the system $A\mathbf{x} = \mathbf{b}$.

---

5. Below is an algorithm to compute the Cholesky factorization of a tridiagonal symmetric positive definite $n \times n$ matrix A, i.e., $A = R^T R$, where R is upper bidiagonal. Assume the diagonal entries of A are stored in an $n \times 1$ vector A_{diag} and the upper bidiagonal entries of A are stored in an $(n-1) \times 1$ vector A_{upper}. There is no need to store the lower bidiagonal entries, since A is symmetric. This algorithm stores the diagonal entries of R in an $n \times 1$ vector R_{diag} and the upper bidiagonal entries in an $(n-1) \times 1$ vector R_{upper}.

$R_{diag}(1) = \sqrt{A_{diag}(1)}$
for $k = 1, \ldots, n-1$

$$R_{upper}(k) = A_{upper}(k)/R_{diag}(k)$$
$$R_{diag}(k+1) = \sqrt{A_{diag}(k+1) - R_{upper}(k)^2}$$

end

Assume that multiplications, divisions, and square roots have equal computational cost. Ignore the cost of additions, subtractions, and indexing.

a. Determine the computational cost (in terms of the number of multiplications, divisions, and square roots) of the above Cholesky factorization algorithm. Provide details to justify your answer.

b. Making **intelligent** use of the above algorithm, what is the **total cost** of solving the system $A\mathbf{x} = \mathbf{b}$, where $\mathbf{b}$ is an $n \times 1$ vector? Explain your answer. **Hint:** Use forward elimination and back substitution on bidiagonal matrices.

c. How does this compare to the cost of solving $A\mathbf{x} = \mathbf{b}$ when A is full and nonsymmetric? Give the asymptotic ratio of the costs.