Test 1 Advanced Computer Graphics

ID:

Name:

No computers or calculators.
If you need to make a simple, reasonable assumption to arrive at an answer then state any such assumption.
Write answers cleanly on the space provided. Use the back of the previous page if more space is needed.

1 Illustrate true or false

Are the following statements true or false? Give a brief example to illustrate your choice.

- (a) question like the posted review questions from the book .

- (z) another question like the posted review questions from the book .

2 Machine epsilon

Suppose ϵ is the smallest positive number so that the IEEE floating-point computation $1 + \epsilon$ differs from 1.

- (a) tough question .

- (z) another tough question .

3 Prove it!

- (15 points) The matrix A in $Ax = b$ is diagonally dominant. We apply LU factorization with partial pivoting. Do pivots occur? .

4 LU factorization

Consider LU factorization without pivoting, $LU = A$. Solving $Ax = b$ can be done by first solving $Ly = b$ for y, and then solving $Ux = y$ for x. In MATLAB notation, `y=L\b ; x=U\y`. When you solve this system, suppose you keep both `x` and `y`.

- (a) (5 points) Split all the matrices L, U, and A into 4 submatrices, each of size $n/2$ by $n/2$. Write out the $LU = A$ expression in this new form.

$$\begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

- (b) (5 points) Also split vectors x and y into two equal parts. Call the first $n/2$ entries of x by the name x_1 and the latter $n/2$ entries as x_2, and likewise for y. Show how to compute y and x using this splitting and your splitting in part (a), by writing the appropriate matrix expressions. Don't write MATLAB code yet.

First solve the system $Ly = b$, which is:

$$\begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

This can be solved in two steps:

1. Solve $L_{11}y_1 = b_1$ for y_1
2. Solve $L_{22}y_2 = b_2 - L_{21}y_1$ for y_2

Next solve the system, $Ux = y$:

$$\begin{bmatrix} U_{11} & U_{12} \\ & U_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

This can be solved in two steps:

1. Solve $U_{22}x_2 = y_2$ for x_2
2. Solve $U_{11}x_1 = y_1 - U_{12}x_2$ for x_2

- (c) (5 points) Now write MATLAB code in detail that computes x and y based on your equations in part (b). It should use colon notation explicitly, and should be syntactically correct. For example, x_1 is the MATLAB expression `x(1:n/2)` and x_2 is `x(n/2+1:n)`. You may use the MATLAB backslash to solve triangular systems (but not for the whole system).

```
y(1:n/2) = L(1:n/2, 1:n/2)  b(1:n/2) y(n/2+1:n) = L(n/2+1:n, n/2+1:n)
(b(n/2+1:n) - L(n/2+1:n, 1:n/2) * y(1:n/2)) x(n/2+1:n) = U(n/2+1:n,
n/2+1:n)  y(n/2+1:n) x(1:n/2) = U(1:n/2, 1:n/2)  (y(1:n/2) - U(1:n/2,
n/2+1:n) * x(n/2+1:n))
```

- (d) (5 points) Now suppose we want to solve a new linear system with the same A, and a new right-hand side $\overline{b}$, where the first $n/2$ entries of $\overline{b}$ are the same as b, and the last $n/2$ entries are completely different. That is,

$$b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \text{ and } \overline{b} = \begin{bmatrix} b_1 \\ \overline{b}_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 + \Delta b_2 \end{bmatrix}$$

where Δb_2 is the change in b_2. You are given Δb_2, and you are compute the new $\overline{y}$ and the new $\overline{x}$. Write out the matrix expressions, just like you did for part (b).

If we start with the solution to part (b), but only b_2 changes, we see that y_1 does not change. Thus, as a first pass we can do:

1. Skip solving $L_{11}y_1 = b_1$ for y_1, since we have y_1 already
2. Solve $L_{22}\overline{y}_2 = \overline{b}_2 - L_{21}y_1$ for $\overline{y}_2$
3. Solve $U_{22}\overline{x}_2 = \overline{y}_2$ for $\overline{x}_2$
4. Solve $U_{11}\overline{x}_1 = \overline{y}_1 - U_{12}\overline{x}_2$ for $\overline{x}_2$

The above solution saves some work, but not all (it's worth just 2 points). To save more, you need to see that the computation $L_{21}y_1$ can be skipped in step (2). The original solution was

$$y_2 = L_{22}^{-1}(b_2 - L_{21}y_1)$$

and the new system is

$$L_{22}\overline{y}_2 = \overline{b}_2 - L_{21}y_1$$

can be written as

$$\overline{y}_2 = L_{22}^{-1}(b_2 + \Delta b_2 - L_{21}y_1)$$
$$\overline{y}_2 = L_{22}^{-1}\Delta b_2 + L_{22}^{-1}(b_2 - L_{21}y_1)$$
$$\overline{y}_2 = L_{22}^{-1}\Delta b_2 + y_2$$

The last expression is less work because it removes the matrix-vector computation $L_{21}y_1$.

- (e) (5 points) Now write MATLAB code in detail that implements your computation in part (d). It should look like your solution to (c), **except** you must save as much work as possible by not recomputing things you already computed in your code in part (c).

```
if you get part (d) wrong, then part (e) is worth at most 3 points.
ynew (1:n/2) = y (1:n/2) ynew (n/2+1:n) = y (n/2+1:n) + L(n/2+1:n, n/2+1:n)
deltab ; xnew(n/2+1:n) = U(n/2+1:n, n/2+1:n)  ynew(n/2+1:n) xnew(1:n/2)
= U(1:n/2, 1:n/2)  (ynew(1:n/2) - U(1:n/2, n/2+1:n) * xnew(n/2+1:n))
or, overwriting 'x' and 'y' with the new solution:  y (n/2+1:n) = y
(n/2+1:n) + L(n/2+1:n, n/2+1:n)  deltab ; x(n/2+1:n) = U(n/2+1:n, n/2+1:n)
y(n/2+1:n) x(1:n/2) = U(1:n/2, 1:n/2)  (y(1:n/2) - U(1:n/2, n/2+1:n)
* x(n/2+1:n)) Either one is fine.
```