Assignment 3

General Instructions

- Solutions due by March 3, Tue in the lab class.

Hand calculations

1. (10 points) The algorithm for Gaussian elimination with partial pivoting is given by

$U = A, L = I, P = I$
for $k = 1 : n - 1$ **do**
  Select $q : |u_{q,k}| = \max_{p \geq k} |u_{p,k}|$
  $u_{k,k:n} \leftrightarrow u_{q,k:n}$ (Row swap in U matrix)
  $l_{k,1:k-1} \leftrightarrow l_{q,1:k-1}$ (Row swap in L matrix)
  $p_{k,1:n} \leftrightarrow p_{q,1:n}$ (Row swap in permutation matrix)
  for $j = k + 1 : n$ **do**
    $l_{j,k} = u_{j,k}/u_{k,k}$ (Multiplication factor)
    $u_{j,k} = 0$ (Element underneath pivot)
    $u_{j,k+1:n} = u_{j,k+1:n} - l_{j,k} \cdot u_{k,k+1:n}$ (Row operations)
  end
end

   (a) Using the above algorithm, given matrix A as,

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 3 \\ 8 & 7 & 9 \end{bmatrix}$$

   obtain the L, U, and P matrices.

   (b) Verify that $PA = LU$.

2. (10 points) Gaussian elimination for banded matrices

   (a) Write down a modified form of the general Gaussian Elimination algorithm to solve for a scalar tri-diagonal system. This matrix has a bandwidth $B = 2$, *i.e.* 1 main diagonal + 1 super-diagonal (and sub-diagonal) populated with non-zeros.

   (b) Count the number of flops (multiplications and divisions) to solve for $\bar{x}$. Is this larger, smaller, or the same compared to that of the Thomas algorithm?

   (c) Generalize this algorithm to solve for a general banded system with bandwidth B

3. (10 points) Consider solving $Ax = b$, with entries of $A_{n\times n}$, $b_{n\times 1} \in \mathcal{C}$ (complex space).

   (a) Convert this problem to that of solving a real square system of order $2n$. *Hint:* Write $A = A_1 + iA_2$, $b = b_1 + ib_2$, $x = x_1 + ix_2$, with $A_1, A_2, b_1, b_2, x_1, x_2$ all real. Determine the number of operations required for solving the complex system $Ax = b$ using real arithmetic and as a real square system of order $2n$.

   (b) Compare these number of operations with those based on directly solving $Ax = b$ using Gaussian elimination and complex arithmetic. Note: One multiplication in complex arithmetic is equivalent of four multiplications in real arithmetic.

   (c) Reduce the real square system of order $2n$ using Gaussian elimination on the blocks of sub-matrices and express the unknown vectors in the following format: $[\ldots]\{x_1\} = \{\ldots\}$ and $[\ldots]\{x_2\} = \{\ldots\}$.

4. (15 points) **Cost of multiplication on a computer:**
Consider a number n represented in $k (= 2^m)$ bits on a computer. Splitting mid way, n can be represented as $n = n_a \times 2^{(k/2)} + n_b$, where n_a and n_b are $k/2$-bit long each. For example if $k = 8\,(i.e.\, 2^3)$ and $n = 10111001$, then $n_a = 1011$ and $n_b = 1001$.

(a) Conventional multiplication: Consider multiplying two numbers n_1 and n_2.
Step 1: $n_1 = n_{1a} \times 2^{(k/2)} + n_{1b}$, $n_2 = n_{2a} \times 2^{(k/2)} + n_{2b}$
Step 2: $n_1 \times n_2 = (n_{1a}n_{2a})2^k + (n_{1a}n_{2b} + n_{1b}n_{2a})\, 2^{(k/2)} + n_{1b}n_{2b}$
Step 3: Calculate the *four* products (colored) in Step 2 with four multiplications. To obtain these products, we recursively apply Steps 1–3 to $n_{1a}, n_{2a}, \ldots$

(i) How is the cost to multiply two k-bit numbers related to the cost to multiply two $(k/2)$-bit numbers?
(ii) Extending this further, how is the cost to multiply two k-bit numbers related to the cost to multiply two 1-bit numbers? *Hint: Use* $k = 2^m$.
(iii) Thus, obtain the computational complexity for conventional multiplication of two k-bit numbers in terms of k.

(b) Fast multiplication (Karatsuba algorithm): Consider multiplying two numbers n_1 and n_2.

Step 1, 2: Same as (a)

Step 3: Calculate the products required in Step 2 with three multiplications as:

A. Calculate $p_1 = n_{1a}n_{2a}$
B. Calculate $p_2 = n_{1b}n_{2b}$
C. Calculate $q = (n_{1a} + n_{1b})(n_{2a} + n_{2b})$
D. Calculate $p_3 = q - p_1 - p_2$

To obtain the products p_1, p_2, q, we recursively apply Steps 1–3 to $n_{1a}, n_{2a}, \ldots$

(i) How is the cost to multiply two k-bit numbers related to the cost to multiply two $(k/2)$-bit numbers?
(ii) Extending this further, how is the cost to multiply two k-bit numbers related to the cost to multiply two 1-bit numbers? *Hint: Use* $k = 2^m$.
(iii) Thus, obtain the computational complexity for fast multiplication of two k-bit numbers in terms of k.

5. (10 points) Calculate the 1-, ∞-, and Frobenius norms for the following matrices

$$A = \begin{bmatrix} 2 & -2 \\ 1 & -3 \end{bmatrix}, \; B = \begin{bmatrix} 1 & -2 & 3 \\ 1 & 5 & 6 \\ 2 & -1 & 3 \end{bmatrix}, \; C = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Programming

1. (15 points) *Solve 2(a) (hand calculations) before getting to this one*
   Consider the following scalar-tridiagonal system,

$$
\begin{pmatrix}
1 & 2 & 0 & . & . & 0 \\
1 & 4 & 1 & 0 & . & 0 \\
0 & 1 & 4 & 1 & . & 0 \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
. & . & . & 1 & 4 & 1 \\
0 & . & . & 0 & 2 & 1
\end{pmatrix}
\begin{pmatrix}
y_0 \\ y_1 \\ x_2 \\ . \\ . \\ . \\ y_{N-1} \\ y_N
\end{pmatrix}
=
\begin{pmatrix}
\frac{N}{3}\left(-\frac{5}{2}f(x_0) + 2f(x_1) + \frac{1}{2}f(x_2)\right) \\
N\left(f(x_2) - f(x_0)\right) \\
N\left(f(x_3) - f(x_1)\right) \\
. \\ . \\ . \\
N\left(f(x_N) - f(x_{N-2})\right) \\
\frac{N}{3}\left(\frac{5}{2}f(x_N) - 2f(x_{N-1}) - \frac{1}{2}f(x_{N-2})\right)
\end{pmatrix}
$$

   where $f(x_j) = \sin(5x_j)$ with $x_j = \frac{3j}{N}, \quad j = 0, 1, ..., N$. Compute the solution y_j and plot y_j versus x_j for $N = 15, 25$ and 50.

2. (30 points) Write a program for LU decomposition of the block tridiagonal matrix in the system shown below for arbitrary N and N_{blk}. D_is are scalar tridiagonal matrices of size N_{blk} and has a structure shown in eq. (1) (shown for $N_{blk} = 5$). A_is and B_is are identity matrices of size N_{blk}. Use either Gaussian elimination or Thomas algorithm (tridiagonal matrix algorithm) for each block. Solve the block tridiagonal system for three right hand side vectors ($N = 10, 20, 30$) given by,

$$
f_i = \begin{cases} 1 & j = 1 \\ 1/N_{blk} & 2 \leq j \leq (N_{blk} - 1) \\ 2 & j = N_{blk} \end{cases}
$$

   for $1 \leq i \leq N$. For all three cases, plot $\bar{x}_i$ vs j for $i = N/2$. Plot the time taken for computation as a function of N, for $N = 10$, 20, and 30 with $N_{blk} = 5$.

$$
\begin{pmatrix}
D_1 & A_1 & . & . & . & . \\
B_2 & D_2 & A_2 & . & . & . \\
. & B_3 & D_3 & A_3 & . & . \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
. & . & . & B_{N-1} & D_{N-1} & A_{N-1} \\
. & . & . & . & B_N & D_N
\end{pmatrix}
\begin{pmatrix}
\bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ . \\ . \\ . \\ \bar{x}_{N-1} \\ \bar{x}_N
\end{pmatrix}
=
\begin{pmatrix}
\bar{f}_1 \\ \bar{f}_2 \\ \bar{f}_3 \\ . \\ . \\ . \\ \bar{f}_{N-1} \\ \bar{f}_N
\end{pmatrix}
$$

$$
D_i = \begin{bmatrix}
-4 & 1 & 0 & 0 & 0 \\
1 & -4 & 1 & 0 & 0 \\
0 & 1 & -4 & 1 & 0 \\
0 & 0 & 1 & -4 & 1 \\
0 & 0 & 0 & 1 & -4
\end{bmatrix} \tag{1}
$$