This assignment is due at the start of the lecture on Friday, 15 Feb. 2019.

For each question that requires you to write a MatLab program, hand in the program and its output as well as any written answers requested in the question. Your program should conform to the usual CS standards for comments, good programming style, etc. Try to format the output from your program so that it is easy to read and to understand. (Your TA may take off marks if your output is really bad.)

Before writing your MatLab programs, you might find it useful to read the MatLab documentation on the course webpage `http://www.cs.toronto.edu/~krj/courses/446-2310/`. You should use sparse matrices as much as possible. Read help in MatLab on sparfun, sparse and spdiags. After you have initialized a matrix using the MatLab sparse matrix routines, you can solve a system $Ax = b$ in MatLab by `x = A \ b`. MatLab will use an efficient sparse matrix factorization to solve the system.

Everyone who has registered for this course should have an account on the CS Teaching Labs Computer System (i.e., formerly called the CDF System). If you are not familiar with the CS Teaching Labs Computer System, take a look at the webpage `https://www.teach.cs.toronto.edu/`.

If you haven't logged in to the CS Teaching Labs Computer System before, see "What do I need to know to start using my Teaching Labs account" on the webpage `https://www.teach.cs.toronto.edu/faq.html`.

You should be able to access the system remotely over the internet. If you haven't done this before, see "If I'm at home, how do I log in to the Teaching Lab machines" on the webpage `https://www.teach.cs.toronto.edu/faq.html`. If you would prefer to access the CS Teaching Labs Computer System by going to one of the Computer Labs, the room numbers of the Labs are BA2200, BA2210, BA2220, BA2230, BA2240, BA2270, BA3175, BA3185, BA3195, BA3200, BA3219.

You don't have to use the CS Teaching Labs Computer System for your CSC 446/2310 assignments, you can use MatLab on your own computer if you prefer. However, the CS Teaching Labs Computer System is available to you if you want to use it.

The main part of this assignment consists of questions 1 to 6, which are out of a total of 70 marks. There are also three bonus questions, which are optional. If you do any or all of the bonus questions, you can earn back marks that you lost on questions 1 to 6. However, the maximum mark that you can earn on this assignment is 70.

Throughout this assignment, I refer to MatLab, but you can use Octave or one of the other MatLab clones instead. (See "MatLab Clones" on the course webpage.) However, MatLab clones are not 100% compatible with MatLab. So, run each of your final programs through MatLab to make sure that your program really runs under MatLab, since, when your TA marks your program, he will want to see a working MatLab program.

1. [15 marks: 5 marks for each part]

Your textbook uses the relation $\Delta_+^s = \mathcal{O}(h^s)$ in section 8.1 to simplify expressions. What Iserles really means by this is that, if $s$ is a positive integer (i.e., $s \in \{1, 2, 3, \ldots\}$) and $z(x)$ has at least $s$ continuous derivatives, then

$$\Delta_+^s z(x) = \mathcal{O}(h^s) \tag{1}$$

where, for a function $z(x)$,

$$\Delta_+ z(x) = z(x+h) - z(x)$$

and, for $s \geq 2$, we can define $\Delta_+^s$ recursively by

$$\Delta_+^s z(x) = \Delta_+ \left(\Delta_+^{s-1} z(x)\right)$$

In parts (a) and (b) below, you will prove (1) in two straightforward, but tedious, steps. In part (c), I ask you to find another proof of (1) that is rigorous, but shorter and more elegant, than the proof given in parts (a) and (b).

(a) As a first step in proving (1), show that

$$\Delta_+^s z(x) = \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} z(x+ih) \tag{2}$$

for any positive integer $s$.

[Note that (2) is quite a useful relationship in its own right.]

(b) Taylor's Theorem with an error term gives

$$z(x+ih) = \sum_{j=0}^{s-1} \frac{(ih)^j}{j!} z^{(j)}(x) + \frac{(ih)^s}{s!} z^{(s)}(\eta_i) \tag{3}$$

where $\eta_i$ is some point in $[x, x+ih]$, assuming $z^{(s)}$ exists and is continuous on $[x, x+ih]$. Substituting the Taylor series (3) into (2), we get

$$\begin{aligned}
\Delta_+^s z(x) &= \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} \left[ \sum_{j=0}^{s-1} \frac{(ih)^j}{j!} z^{(j)}(x) + \frac{(ih)^s}{s!} z^{(s)}(\eta_i) \right] \\
&= \sum_{i=0}^{s} \sum_{j=0}^{s-1} (-1)^{s-i} \binom{s}{i} i^j \, \frac{h^j z^{(j)}(x)}{j!} + h^s \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} \frac{i^s}{s!} z^{(s)}(\eta_i) \\
&= \sum_{j=0}^{s-1} \left[ \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} i^j \right] \frac{h^j z^{(j)}(x)}{j!} + h^s \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} \frac{i^s}{s!} z^{(s)}(\eta_i)
\end{aligned}$$

Hence, if you can show that

$$\sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} i^j = 0 \tag{4}$$

for $j = 0, 1, \ldots, s-1$, you'll have

$$\Delta_+^s z(x) = h^s \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i} \frac{i^s}{s!} z^{(s)}(\eta_i) = \mathcal{O}(h^s)$$

as required. To prove (4), note that

$$(x-1)^s = \sum_{i=0}^{s} \binom{s}{i} x^i (-1)^{s-i} \tag{5}$$

Hence, setting $x = 1$ in (5), we get

$$0 = (1-1)^s = \sum_{i=0}^{s} (-1)^{s-i} \binom{s}{i}$$

which proves (4) for $j = 0$. Extend this approach to prove (4) for $j = 1, \ldots, s-1$.

(c) Give another rigorous, but shorter and more elegant, proof of (1).

2. [5 marks]

Do question 8.1 on page 166 of your textbook.

3. [10 marks]

Do question 8.5 on page 167 of your textbook.

4. [10 marks]

Write a MatLab program that uses the 5-point centered difference approximation to the Laplacian $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ that we discussed in class to compute a numerical approximation to Poisson's equation $\nabla^2 u = f$ on the unit square $\Omega = (0,1) \times (0,1)$ with Dirichlet boundary conditions. Your program should use an evenly spaced mesh with $\Delta x = \Delta y = 1/(n+1)$, where $n$ is an integer. (See below for the choices of $n$.)

As a particular example, let

$$f(x,y) = \frac{2}{(1+x)^3} + \frac{2}{(1+y)^3}$$

and let the Dirichlet boundary conditions be

$$\begin{aligned} u(x,0) &= 1 + \tfrac{1}{1+x} && \text{for } x \in [0,1] \\ u(x,1) &= \tfrac{1}{2} + \tfrac{1}{1+x} && \text{for } x \in [0,1] \\ u(0,y) &= 1 + \tfrac{1}{1+y} && \text{for } y \in [0,1] \\ u(1,y) &= \tfrac{1}{2} + \tfrac{1}{1+y} && \text{for } y \in [0,1] \end{aligned}$$

It is easy to verify that the solution to this problem is

$$u(x,y) = \frac{1}{1+x} + \frac{1}{1+y}$$

Let $\tilde{u}_{i,j} = u(i\Delta x, j\Delta y)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$.

For each of $n = 9, 19, 39, 79$, use your program to compute the numerical solution $u_{i,j}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ to this problem and compute and print the maximum error in the numerical solution

$$\max\{|u_{i,j} - \tilde{u}_{i,j}| : i = 1, \ldots, n, \ j = 1, \ldots, n\}$$

How does this error decrease with $\Delta x = \Delta y = 1/(n+1)$?

Discuss whether or not this rate of convergence agrees with the theory that we discussed in class.

5. [15 marks]

Write another MatLab program that uses the 5-point centered difference approximation to the Laplacian $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ that we discussed in class to compute a numerical approximation to Poisson's equation $\nabla^2 u = f$ on the unit square $\Omega = (0,1) \times (0,1)$ with Dirichlet boundary conditions on three sides and a Neumann boundary condition on the fourth side. (See below for the actual boundary conditions.)

Your program should use an evenly spaced mesh with $\Delta x = \Delta y = 1/(n+1)$, where $n$ is an integer. (See below for the choices of $n$.)

As a particular example, let

$$f(x,y) = \frac{2}{(1+x)^3} + \frac{2}{(1+y)^3}$$

and let the Dirichlet boundary conditions be

$$\begin{aligned} u(x,0) &= 1 + \tfrac{1}{1+x} \qquad &\text{for } x \in [0,1] \\ u(x,1) &= \tfrac{1}{2} + \tfrac{1}{1+x} \qquad &\text{for } x \in [0,1] \\ u(0,y) &= 1 + \tfrac{1}{1+y} \qquad &\text{for } y \in [0,1] \end{aligned}$$

and let the Neumann boundary condition be

$$u_x(1,y) = -\frac{1}{4} \quad \text{for } y \in [0,1]$$

It is easy to verify that the solution to this problem is

$$u(x,y) = \frac{1}{1+x} + \frac{1}{1+y}$$

Let $\tilde{u}_{i,j} = u(i\Delta x, j\Delta y)$ for $i = 1,\ldots,n$ and $j = 1,\ldots,n$.

Try two different ways to approximate the Neumann boundary condition.

(a) Approximate $u_x(1,y)$ by the first-order backward difference $\frac{1}{\Delta x}\Delta_{-,x}$. That is, use the boundary condition

$$\frac{u_{n+1,j} - u_{n,j}}{\Delta x} = -\frac{1}{4}$$

for $j = 1,\ldots,n$.

(b) Approximate $u_x(1,y)$ by the second-order backward difference $\frac{1}{\Delta x}(\Delta_{-,x} + \frac{1}{2}\Delta^2_{-,x})$. That is, use the boundary condition

$$\frac{\frac{3}{2}u_{n+1,j} - 2u_{n,j} + \frac{1}{2}u_{n-1,j}}{\Delta x} = -\frac{1}{4}$$

for $j = 1,\ldots,n$.

For each of the two approaches described above to approximate the Neumann boundary condition and for each of $n = 9, 19, 39, 79$, use your program to compute the numerical solution $u_{i,j}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ to this problem and compute and print the maximum error in the numerical solution

$$\max\{|u_{i,j} - \tilde{u}_{i,j}| : i = 1, \ldots, n+1,\ j = 1, \ldots, n\}$$

Note the $n+1$ in the line above.

For each of the two methods described above to approximate the Neumann boundary condition, how does this error decrease with $\Delta x = \Delta y = 1/(n+1)$? That is, does the error appear to be first-order accurate, second-order accurate or something in between?

Discuss whether or not this rate of convergence agrees with the theory that we discussed in class.

6. [15 marks]

Write another MatLab program that uses the 5-point centered difference approximation to the Laplacian $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ that we discussed in class to compute a numerical approximation to Poisson's equation $\nabla^2 u = f$ in $\Omega = \{(x,y) : x^2 + y^2 < 1\}$, the open disk of radius one centered at the origin, with Dirichlet boundary conditions on the boundary of the disk, $\partial\Omega = \{(x,y) : x^2 + y^2 = 1\}$.

Your program should use an evenly spaced mesh with $\Delta x = \Delta y = 1/(n+1)$, where $n$ is an integer. (See below for the choices of $n$.)

As a particular example, let

$$f(x,y) = 16(x^2 + y^2)$$

and let the Dirichlet boundary conditions be

$$u(x,y) = 1 \quad \text{for } (x,y) \in \partial\Omega$$

It is easy to verify that the solution to this problem is

$$u(x,y) = (x^2 + y^2)^2$$

Let $\tilde{u}_{i,j} = u(i\Delta x, j\Delta y)$ for all $i$ and $j$ such that $(i\Delta x, j\Delta y) \in \Omega$.

The difficulty with this problem is that you have to handle *near-boundary points.* (See Figure 8.3 and the discussion on pages 155–156 of the textbook.) The textbook describes two ways to handle the near-boundary points.

(a) A first-order method described at the bottom of page 155.

(b) A second-order method described on page 156.

For each of the two approaches described in the textbook to handle the near-boundary points and for each of $n = 9, 19, 39, 79$, use your program to compute the numerical solution $u_{i,j}$ for all $i$ and $j$ such that $(i\Delta x, j\Delta y) \in \Omega$ and compute and print the maximum error in the numerical solution

$$\max\{|u_{i,j} - \tilde{u}_{i,j}| : (i\Delta x, j\Delta y) \in \Omega\}$$

For each of the two methods described in the textbook to handle the near-boundary points, how does this error decrease with $\Delta x = \Delta y = 1/(n+1)$? That is, does the error appear to be first-order accurate, second-order accurate or something in between?

Discuss whether or not this rate of convergence agrees with the theory that we discussed in class.

7. Bonus Question [5 marks if you solve this problem yourself; 2.5 marks if you can find a published solution[1]]

   Consider the $n \times n$ matrix

$$A = \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -2 \end{bmatrix}$$

   where $\Delta x = 1/(n+1)$. Show that there exists a constant $C$, independent of $n$, such that $\|A^{-1}\|_\infty \leq C$ for all $n \geq 3$.

8. Bonus Question [5 marks if you solve this problem yourself; 2.5 marks if you can find a published solution[1]]

   Consider the block tridiagonal matrix

$$A = \frac{1}{\Delta x^2} \begin{bmatrix} T & I & 0 & 0 & 0 & \cdots & 0 \\ I & T & I & 0 & 0 & \cdots & 0 \\ 0 & I & T & I & 0 & \cdots & 0 \\ 0 & 0 & I & T & I & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & I & T & I \\ 0 & 0 & 0 & \cdots & 0 & I & T \end{bmatrix}$$

[1]Note that it is plagiarism if you find a published solution (whether in a book or in a paper or online) and present the work as your own.

where $\Delta x = 1/(n+1)$, each 0 in the matrix above is an $n \times n$ zero submatrix (i.e., all elements in the submatrix are zero), $I$ is the $n \times n$ identity matrix and $T$ is the $n \times n$ tridiagonal matrix

$$T = \begin{bmatrix} -4 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -4 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -4 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & -4 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & \cdots & 1 & -4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -4 \end{bmatrix}$$

Each row and each column of $A$ consists of $n$ submatrices and each submatrix is $n \times n$. Therefore, $A$ is an $n^2 \times n^2$ matrix.

Is there a constant $C$, independent of $n$, such that $\|A^{-1}\|_\infty \leq C$ for all $n \geq 3$?

9. Bonus Question [5 marks if you solve this problem yourself; 2.5 marks if you can find a published solution[1]]

   In Lemma 8.3 on page 157 of your textbook, the author claims that

   > "Moreover, $\lambda \in \sigma(B)$ may lie on $\partial S_{i^o}$ for some $i^o \in \{1, 2, \ldots, d\}$ only if $\lambda \in \partial S_i$ for all $i \in \{1, 2, \ldots, d\}$."

   Give an example to show that this claim is not true.

   Explain why your example shows that this claim is not true.