Project 1 - FYS3150

Josef Ayman M.
(Dated: September 12, 2024)

List a link to your github repository here!

PROBLEM 1

We check that $u(x) = 1 - (1 - e^{-10})x - e^{-10x}$ is an exact solution to our Poisson equation.
We simplify it:

$$u(x) = 1 - x + e^{-10}x - e^{-10x}$$

Then we try to find the second derivative:

$$u'(x) = -1 + e^{-10} + 10e^{-10x}u''(x) = -100e^{-10x}$$

Which when fits the source term $-(-100e^{-10x}) = f(x) = 100e^{-10x}$
And the boundary conditions:

$$u(0) = 1 - 0 + 0 - 1 = 0u(1) = 1 - 1 + e^{-10} - e^{-10} = 0$$

PROBLEM 2

In problem 2 we will write a program that:

- Defines a vector of x values.
- Evaluates the exact solution at each point.
- Writes the results to a file.
- It's recommended to use the *armadillo* library.

We can include figures using the `figure` environment. Whenever we include a figure or table, we *must* make sure to actually refer to it in the main text, e.g. something like this: "In figure 1 we show …".

PROBLEM 3

Discretization of the Domain

- The domain x ranges from 0 to 1.
- We divide the domain into N equally spaced intervals, where h is the spacing between grid points, $h = \frac{1}{N}$.
- Define grid points x_i as:

$$x_i = ih \quad \text{for } i = 0, 1, 2, \dots, N$$

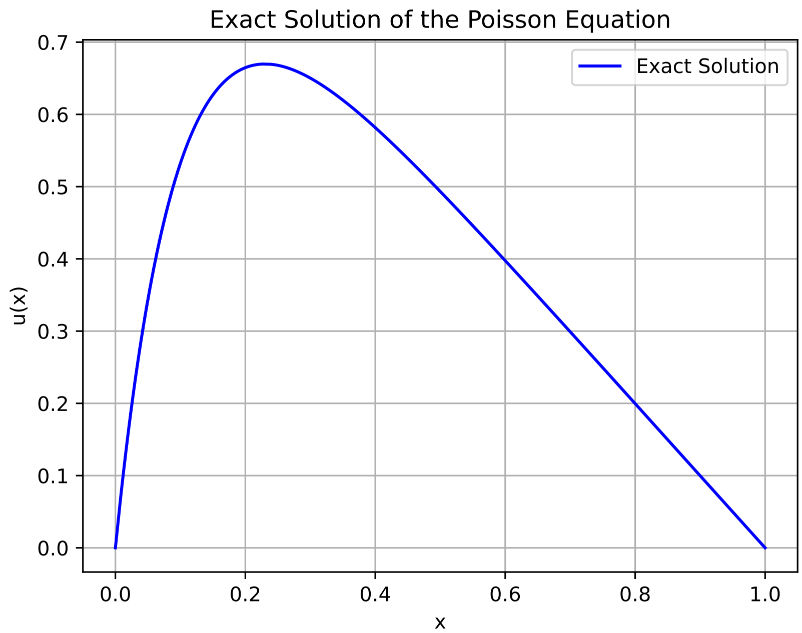

FIG. 1. Write a descriptive caption here that explains the content of the figure. Note the font size for the axis labels and ticks — the size should approximately match the document font size.

Let v_i be the approximation to $u(x_i)$.

To approximate the second derivative $\frac{d^2u}{dx^2}$ at a point x_i, we use the **finite difference method**. The central difference approximation for the second derivative is:

$$\frac{d^2u}{dx^2} \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}$$

Substitute into the Poisson Equation

$$-\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} = f(x_i)$$

Simplify

Rearrange the equation to solve for the discrete approximation:

$$v_{i+1} - 2v_i + v_{i-1} = -h^2 f_i$$

or

$$-\frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} = f_i \tag{1}$$

Boundary Conditions

In the problem, the boundary conditions are $u(0) = 0$ and $u(1) = 0$. Therefore:

$$v_0 = 0 v_N = 0$$

Putting it all together, the discretized version of the Poisson equation is:

$$v_{i+1} - 2v_i + v_{i-1} = -h^2 \cdot 100e^{-10x_i}$$

for $i = 1, 2, \ldots, N-1$, with boundary conditions:

$$v_0 = 0 \quad \text{and} \quad v_N = 0$$

PROBLEM 4

To rewrite the discretized Poisson equation as a matrix equation, we start from the discretized second derivative equation 1:

This can be written in matrix form as $\mathbf{A}\vec{v} = \vec{g}$, where $\mathbf{A}$ looks like:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

g_i is our right hand of the equation.

$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

$$\begin{bmatrix} 2v_1 & -v_2 & 0 & 0 \\ -v_1 & 2v_2 & -v_3 & 0 \\ 0 & -v_2 & 2v_3 & -v_4 \\ 0 & 0 & -v_3 & 2v_4 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix}$$

We now showed that we can rewrite the equation in the matrix form $\mathbf{A}\vec{v} = \vec{g}$

PROBLEM 5

Problem a

The matrix equation for the internal points can be written as:

$$\mathbf{A}\vec{v} = \vec{g}$$

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}, \quad \vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \vec{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$$

Writing out the first and last rows, we see that:

$$2v_1 - v_2 = g_1, \quad -v_{n-1} + 2v_n = g_n$$

These correspond to the internal points of $\vec{v}^*$, excluding the boundary values. Since v_1 corresponds to the second element of $\vec{v}^*$and v_n corresponds to the second-to-last element, we conclude that $m = n + 2$, accounting for the two boundary conditions $v_1^* = 0$ *and $v_m^* = 0$*.

Problem b

The vector $\vec{v}$, which we solve for in $\mathbf{A}\vec{v} = \vec{g}$, represents the internal points of $\vec{v}^*$. The boundary points $v_1^* = 0$ and $v^m = 0$ are known and are excluded from the matrix equation. Additionally, g_i corresponds to $f(xi+1)$, where x_{i+1} is the corresponding discretized point.

PROBLEM 6

Problem a - Algorithm for Solving the System

To solve the system $\mathbf{A}\vec{v} = \vec{g}$ where $\mathbf{A}$ is a general tridiagonal matrix, we can use a method called *Thomas Algorithm* (a specialized form of Gaussian elimination for tridiagonal matrices).

Here's the algorithm broken down step by step:

Algorithm 1 Thomas Algorithm for Tridiagonal Matrices

$c_1 \leftarrow \frac{c_1}{b_1}$
$g_1 \leftarrow \frac{g_1}{b_1}$
for $i = 2$ to n **do**
  $b_i \leftarrow b_i - a_i c_{i-1}$ ▷ Update diagonal
  if i < n **then**
    $c_i \leftarrow \frac{c_i}{b_i}$ ▷ Update superdiagonal (except for the last row)
  $g_i \leftarrow \frac{g_i - a_i g_{i-1}}{b_i}$ ▷ Update right-hand side
$v_n \leftarrow g_n$ ▷ Backward substitution starts here
for $i = n-1$ to 1 **do**
  $v_i \leftarrow g_i - c_i v_{i+1}$ ▷ Solve for remaining v_i

Problem b - Number of Floating-Point Operations (FLOPs)

We count the number of operations in each phase.

- **Forward substitution:** For each i from 2 to n, we have 3 FLOPs (2 divisions and 1 subtraction). This gives a total of $3(n-1)$ FLOPs.

  1 division for c_i

  1 division for g_i

  1 subtraction for b_i

- **Backward substitution:** For each i from $n-1$ to 1, we have 2 FLOPs (1 multiplication and 1 subtraction). This gives a total of $2(n-1)$ FLOPs.

  1 multiplication for $c_i v_{i+1}$

  1 subtraction for $g_i - c_i v_{i+1}$

- **Total:** The total number of FLOPs is $3(n-1) + 2(n-1) = 5(n-1) = 5n - 5$.

PROBLEM 7