FYS3150 - Project 1

Andrew Quan, Oskar Idland, Hishem Kløvnes, Håvard Skåli
(Dated: August 24, 2023)

GitHub Repository

PROBLEM 1

We have the one-dimensional Poisson equation

$$-\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = 100e^{-10x}, \qquad x \in [0, 1] \tag{1}$$

where the right hand side of the equation is the source term $f(x)$. We have the boundary conditions $u(0) = 0$ and $u(1) = 0$. Integrating both sides of (1) with respect to x twice we get

$$\begin{aligned}
\frac{\mathrm{d}^2 u}{\mathrm{d}x^2} &= -100e^{-10x} \\
\iint \left(\frac{\mathrm{d}^2 u}{\mathrm{d}x^2}\right) \mathrm{d}x^2 &= -100 \iint e^{-10x} \mathrm{d}x^2 \\
\int \left(\frac{\mathrm{d}u}{\mathrm{d}x}\right) \mathrm{d}x &= -100 \int \left(-\frac{1}{10}e^{-10x} + C\right) \mathrm{d}x \\
u(x) &= -100 \left(\frac{1}{100}e^{-10x} + Cx + D\right) \\
u(x) &= Cx + D - e^{-10x} \qquad (2)
\end{aligned}$$

Where C and D are some arbitrary constants. Invoking the boundary conditions we get

$$\begin{aligned}
u(0) = D - 1 = 0 \quad &\Rightarrow \quad D = 1 \\
u(1) = C + 1 - e^{-10} = 0 \quad &\Rightarrow \quad C = e^{-10} - 1
\end{aligned}$$

Thus, plugging these constants into the general solution and rearranging the terms, the final solution becomes

$$u(x) = 1 - x\left(1 - e^{-10}\right) - e^{-10x} \tag{3}$$

just like we wanted to show.

PROBLEM 2

After creating a program that defines a vector of x-values between 1 and 100 and evaluates the exact solution $u(x)$ defined as in (3), we wrote these numbers with four decimals into a text file and plotted them against each other in Python. The resulting plot is shown in FIG. 1.

PROBLEM 3

Our aim is to derive a discretized version of the Poisson equation which lets us calculate an approximate value v of the exact value u. To do this we must calculate the change in the change in v per change in x to calculate the change in v. In other words, the first step is to write

```
arma::vec x {arma::linspace <arma::vec>(0, 1, 100)};
arma::vec v {0};

double dx = x.at(1) - x.at(0);       // difference in x between data points

```

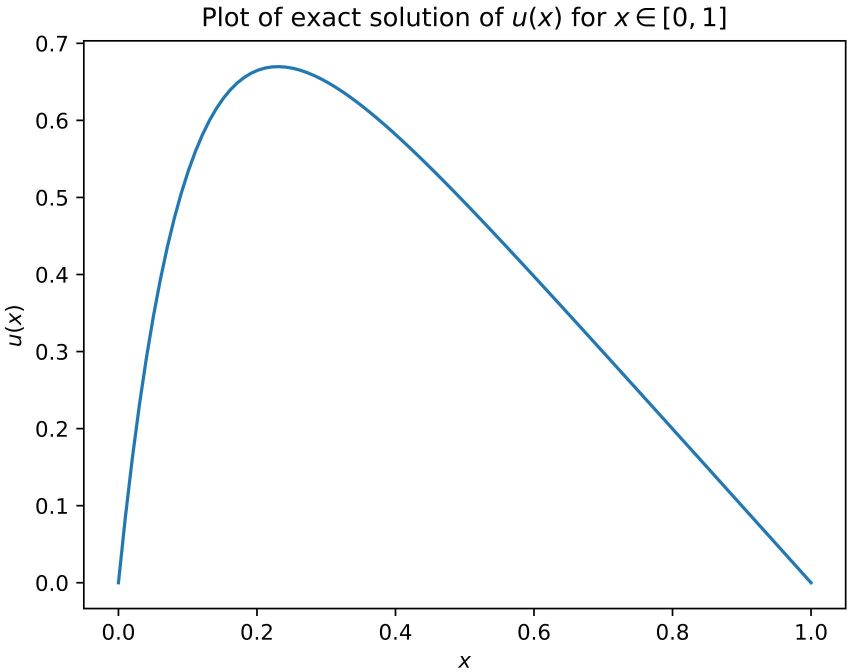

FIG. 1. Plot of the exact solution of $u(x)$ for $x \in [0, 1]$. We used 100 data points for the plot.

```
double ddvddx {- 100*exp(- 10*x.at(0))};        // double derivative of v(x)
double dvdx {ddvddx*dx};                         // derivative of v(x)

for (int i = 0, i < 100, ++i){
    ddvddx = - 100*exp(- 10*x.at(i));
    dvdx += ddvddx*dx;
    v.at(i+1) = v.at(i) + dvdx*dx;
}
```

PROBLEM 4

PROBLEM 5

Problem 5a

Problem 5b

PROBLEM 6

Problem 6a

Problem 6b

PROBLEM 7

Problem 7a

Problem 7b

PROBLEM 8

Problem 8a

Problem 8b

Problem 8c

PROBLEM 9

Problem 9a

Problem 9b

Problem 9c

PROBLEM 10

We write equations using the LaTeX `equation` (or `align`) environments. Here is an equation with numbering

$$\mathbf{F} = \frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}, \tag{4}$$

and here is one without numbering:

$$\oint_C \mathbf{F} \cdot \mathrm{d}\mathbf{r} = 0.$$

Sometimes it is useful to refer back to a previous equation, like we're demonstrating here for equation 4.

Also, note the LaTeX code we used to get correct quotation marks in the previous sentence. (Simply using the `"` key on your keyboard will give the wrong result.) Figures should preferably be vector graphics (e.g. a `.pdf` file) rather than raster graphics (e.g. a `.png` file).

By the way, don't worry too much about where LaTeX decides to place your figures and tables — LaTeX knows more than we do about proper document layout. As long as you label all your figures and tables and refer to them in the text, it's all good. Of course, in some cases it can be worth trying to force a specific placement, to avoid the figure/table appearing many pages away from the main text discussing it, but this isn't something you should spend time on until the very end of the writing process.

TABLE I. Write a descriptive caption here, explaining the content of your table.

Number of points	Output
10	0.3086
100	0.2550

Next up is a table, created using the `table` and `tabular` environments. We refer to it by table I.
Finally, we can list algorithms by using the `algorithm` environment, as demonstrated here for algorithm 1.

Algorithm 1 Some algorithm

Some maths, e.g $f(x) = x^2$. ▷ Here's a comment
for $i = 0, 1, ..., n-1$ **do**
  Do something here
while Some condition **do**
  Do something more here
Maybe even some more math here, e.g $\int_0^1 f(x)\mathrm{d}x$