Math 5620 Numerical Solution of Differential Equation Homework 7

Staple your solutions to this sheet Score:_____

Problem 1: Write a code that will compute the solution of the heat equation using the explicit Euler method for time-stepping and the usual second order spatial discretization as discussed in class. Make sure the time step honors the stability condition for parabolic problems.

Problem 2: Repeat the work in Problem 1 using your implicit Euler method. Compare the results to those in Problem 1.

Problem 3: Instead of using the step 10^{-3} in your implicit Euler method, try increasing the size of the step to 10^{-2} and so on. What happens to the accuracy of your results?

Problem 4: Use the predictor-corrector method you developed in Homework 5 on the same problems as in Problem 1. Compare your results to the results from the implicit and explicit Euler methods.

Problem 5: Use the fourth order Runge Kutta method you developed earlier in class to try the same problems. Compare the results and also determine the difference in time needed to use the Runge Kutta method of order four.

HW7 George Staples

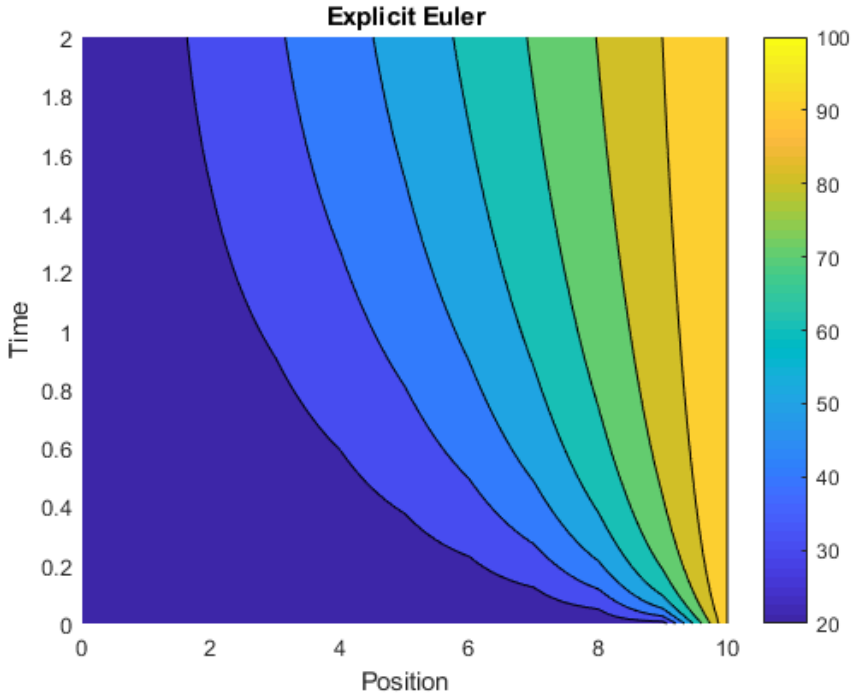
P1-

https://georgest347.github.io/MATH-5620/softwareManual/HW7/heatEE1D
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g1
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g2

This code was used to solve the steady state temperatures of a rod with fixed temperatures at the ends. The table below shows the final temperatures for the given inputs.
Inputs:
X(t,0)=20;
X(t,L)=100;
X(0,x)=25;
L=10;
Final time =2;
k=10;
dt=0.001;
dx=1.0;

20	26.0806	32.3471	38.9681	46.0784	53.7665	62.0647	70.9459	80.3249	90.0669	100

A contour plot shown below provides a visual representation of how the rod temperatures change with time. Plots for each of the subsequent problems were generated, but no noticeable changes were observed, therefore numerical data for the t=2 values will be compared.



P2-

https://georgest347.github.io/MATH-5620/softwareManual/HW7/heatIE1D
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g1
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g2

This code was used with the same inputs as problem 1. The final temperatures are given in the table below:

20	26.0771	32.3404	38.9587	46.0673	53.7545	62.0531	70.9359	80.3176	90.063	100

The Implicit Euler method is an under approximation compared to the Explicit Euler method. The two methods get similar results for a dt of 0.001 and dx of 1.

P3-

The following table shows how the dt affects the results of the Implicit Euler method.

dt											
10^-3	20	26.0771	32.3404	38.9587	46.0673	53.7545	62.0531	70.9359	80.3176	90.063	100
10^-2	20	26.0615	32.3104	38.9168	46.0171	53.7008	62.0011	70.8909	80.2844	90.0454	100
10^-1	20	25.9127	32.0234	38.513	45.5296	53.173	61.4847	70.4405	79.9508	89.8678	100

As the dt goes closer to 0 the solution produced by the Implicit Euler method gets farther from the actual solution. The Accuracy goes down.

P4-

https://georgest347.github.io/MATH-5620/softwareManual/HW7/g1
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g2
https://github.com/georgest347/MATH-5620/blob/master/softwareManual/HW7/heatPC.md
https://github.com/georgest347/MATH-5620/blob/master/softwareManual/HW7/functInital.md

Using the modified Predictor Corrector method from HW 5 and the same input values as P1 and P2, the function yielded:

20	26.0788	32.3438	38.9634	46.0728	53.7605	62.0589	70.9409	80.3212	90.0649	100

These results are similar to those of the Implicit and Explicit Euler methods. Since the Explicit Euler method tends to over predict the values and the Implicit Euler under predicts, the PC method should be closer to the exact solution.

P5-

https://georgest347.github.io/MATH-5620/softwareManual/HW7/g1
https://georgest347.github.io/MATH-5620/softwareManual/HW7/g2
https://github.com/georgest347/MATH-5620/blob/master/softwareManual/HW7/heatRK4.md
https://github.com/georgest347/MATH-5620/blob/master/softwareManual/HW7/functInital.md

Using the RK4 function which was modified with method of lines, the following data was collected using the same input values as in problem 1.

20	26.0791	32.3443	38.9642	46.0738	53.7615	62.0599	70.9417	80.3219	90.0652	100

The time step needed to run the RK4 method must satisfy the following equation:

$$k\frac{\Delta t}{\Delta x^2} \leq 1$$