ME4565

Project A

Ryan Huang

Theory Section

1) $\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$ From $A\phi_{xx} + B\phi_{xy} + C\phi_{yy} + D\phi_x + E\phi_y + F\phi = G$

$A = \alpha,\ B = 0,\ C = 0$

$B^2 - 4AC = 0 - 0 = \boxed{0 \rightarrow \text{Parabolic Equation}}$

Slope: $\frac{dx}{dt} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} = 0$ $\boxed{\frac{dx}{dt} = 0}$

2) $\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$

$A = 1,\ B = 0,\ C = 1$

$B^2 - 4AC = 0 - 4(1)(1) = \boxed{-4 < 0 \rightarrow \text{Elliptic Equation}}$

3) $\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$

$A = c^2,\ B = 0,\ C = -1$

$B^2 - 4AC = 0 - 4(c^2)(-1) = \boxed{4c^2 > 0 \rightarrow \text{Hyperbolic Equation}}$

Slope: $\frac{dx}{dt} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} = \frac{2c}{2c^2} = \pm \frac{1}{c}$ $\boxed{\frac{dx}{dt} = \pm \frac{1}{c}}$

4) $\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$

This equation is 1st-Order and does not belong to the classification equation above, but it has hyperbolic features

Transform the PDE into an ODE along the curve $(x(s), t(s))$ where $\frac{d}{ds}(u(x(s), t(s))) = F(u, x(s), t(s))$

$\frac{d}{ds}(u(x(s), t(s))) = \frac{\partial u}{\partial x}\frac{dx}{ds} + \frac{\partial u}{\partial t}\frac{dt}{ds}$

Let $\frac{dx}{ds} = c,\ \frac{dt}{ds} = 1$ so that $\frac{du}{ds} = \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$

Solving for t, x, and u

$\frac{dt}{ds} = 1 \rightarrow t = s$

$\frac{dx}{ds} = c \rightarrow x = cs + C_1 = ct + C_1$

$\frac{du}{ds} = 0 \rightarrow F(u(0), x(0), t(0)) = F(C_1) = \boxed{F(x - ct)}$

$\boxed{\text{The slope of its characteristic lines is } c \quad \frac{dx}{dt} = c}$

3.3) $\frac{\partial^2 u}{\partial x \partial y} = 0$

$$A = 0,\ B = 1,\ C = 0$$

$$B^2 - 4AC = 1 - 0 = \boxed{1 > 0 \text{ Hyperbolic Equation}}$$

$$(x^2-1)\frac{\partial^2 u}{\partial x^2} + 2\frac{\partial^2 u}{\partial y^2} = 25(x^3-1)\frac{\partial u}{\partial x}$$

$$A = (x^2-1) \quad B = 0 \quad C = 2$$

$$B^2 - 4AC = 0 - 4(x^2-1)(2) = -8x^2+8$$

$$\boxed{\begin{array}{lll} -8x^2+8=0 \rightarrow & x=-1,1 & \text{Parabolic} \\ -8x^2+8>0 \rightarrow & -1<x<1 & \text{Hyperbolic} \\ -8x^2+8<0 \rightarrow & x<-1,\ x>1 & \text{Elliptic} \end{array}}$$

$$\frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial t} = \cos(5t)$$

$$A = 1,\ B = 0,\ C = 1$$

$$B^2 - 4AC = 0 - 4(1)(1) = \boxed{-4 < 0 \text{ Elliptic}}$$

3.4) $\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$

$$\underbrace{\frac{\partial q(x,t)}{\partial t} + R\frac{\partial q(x,t)}{\partial x} = 0}$$

$$\left.\begin{array}{l} v = \frac{\partial u}{\partial t},\quad w = \frac{\partial u}{\partial x}a \\ \frac{\partial v}{\partial t} = \frac{\partial^2 u}{\partial t^2},\quad \frac{dw}{dx} = a\frac{\partial^2 u}{\partial x^2} \end{array}\right\} \frac{dv}{dt} = a\frac{dw}{dx}$$

$$\left\{\begin{array}{l} q = [u_t,\ au_x] \\ q_t = [u_{tt},\ au_{xt}]^T \\ q_x = [u_{tx},\ au_{xx}]^T \end{array}\right. \qquad R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$$

$$\begin{bmatrix} u_{tt} \\ au_{xt} \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}\begin{bmatrix} u_{tx} \\ au_{xx} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$u_{tt} + u_{tx}R_{11} + au_{xx}R_{12} = 0$$

$$au_{xt} + u_{xt}R_{21} + au_{xx}R_{22} = 0$$

$$R_{11} = R_{22} = 0 \qquad R_{12} = R_{21} = -a$$

$$R = \begin{bmatrix} 0 & -a \\ -a & 0 \end{bmatrix} \quad \text{Eigenvalues of } R \quad |R - \lambda I| = \begin{vmatrix} -\lambda & -a \\ -a & -\lambda \end{vmatrix} = 0$$

$$\lambda^2 - a^2 = 0 \rightarrow \boxed{\lambda = \pm a}$$

Since the eigenvalues are real and distinct, the system is hyperbolic which matches the original equation

3.5) $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y)$

$\frac{\partial}{\partial x}(q(x,t)) + R\frac{\partial}{\partial y}(q(x,t)) = 0$

$$\left.\begin{array}{ll} v = \frac{du}{dx} & w = \frac{du}{dy} \\ \frac{dv}{dx} = \frac{\partial^2 u}{\partial x^2} & \frac{dw}{dy} = \frac{\partial^2 u}{\partial y^2} \end{array}\right\} \rightarrow \frac{dv}{dx} + \frac{dw}{dy} = f(x,y)$$

$$\left\{\begin{array}{l} q = [u_x, u_y] \\ q_x = [u_{xx}, u_{yx}]^T \\ q_y = [u_{xy}, u_{yy}]^T \end{array}\right.$$

$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$

$f(x,y) = u_{xx} + u_{yy}$

$$\begin{bmatrix} u_{xx} \\ u_{xy} \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}\begin{bmatrix} u_{xy} \\ u_{yy} \end{bmatrix} = \begin{bmatrix} u_{xx} + u_{yy} \\ 0 \end{bmatrix}$$

$u_{xx} + R_{11} u_{xy} + R_{12} u_{yy} = u_{xx} + u_{yy}$

$u_{xy} + R_{21} u_{xy} + R_{22} u_{yy} = 0$

$R_{22} = R_{11} = 0 \quad R_{12} = 1 \quad R_{21} = -1$

$R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Eigenvalues of R $|R - \lambda I| = \begin{vmatrix} -\lambda & 1 \\ -1 & -\lambda \end{vmatrix} = 0$

$\lambda^2 + 1 = 0 \quad \boxed{\lambda = \pm i}$

Since all eigenvalues are complex, the system is elliptic as Poissons equation is

Numerical Programming Section: $\frac{\partial T}{\partial t} = \alpha\frac{\partial^2 T}{\partial x^2}$, $T^* = \frac{T - T_\infty}{T_i - T_\infty}$ $\quad s = \frac{x}{L} \quad \tau = \frac{\alpha}{L^2}t$

BC: $x=0, \frac{\partial T}{\partial x} = 0$

$x = L, -k\frac{\partial T}{\partial x} = h(T - T_\infty) \quad T(x, t=0) = T_i$

a)
$$\left.\begin{array}{l} T = T^*(T_i - T_\infty) + T_\infty \\ x = sL \\ t = \frac{L^2}{\alpha}\tau \end{array}\right\} \quad \frac{\partial(T^*(T_i - T_\infty) + T_\infty)}{\partial(\frac{L^2}{\alpha}\tau)} = \alpha\frac{\partial^2(T^*(T_i - T_\infty) + T_\infty)}{\partial(sL)^2}$$

$$\frac{\alpha(T_i - T_\infty)}{L^2}\frac{\partial T^*}{\partial \tau} = \frac{\alpha(T_i - T_\infty)}{L^2}\frac{\partial^2 T^*}{\partial s^2}$$

Nondimensionalized Heat Equation $\boxed{\frac{\partial T^*}{\partial \tau} = \frac{\partial^2 T^*}{\partial s^2}}$

BCs: $\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \rightarrow \boxed{\frac{\partial T^*}{\partial s} = 0 \text{ at } s=0}$ $\leftarrow$ Nondimensionalized BCs

$$-\frac{k(T_i - T_\infty)}{L}\frac{\partial T^*}{\partial s} = h(T^*(T_i - T_\infty) - T_\infty) \rightarrow \boxed{\frac{\partial T^*}{\partial s} = -\frac{Lh}{k}T^* + \frac{Lh}{k}\frac{T_\infty}{T_i - T_\infty} \text{ at } s=1}$$

b) Discretized equations and BCs

~~Let~~ Writing T^* as T for simplicity

$$\boxed{\begin{array}{l} \text{Explicit Method: } T_i^{n+1} = \frac{\Delta t}{\Delta x^2}T_{i-1}^n + (1 - 2\frac{\Delta t}{\Delta x^2})T_i^n + \frac{\Delta t}{\Delta x^2}T_{i+1}^n \\ \text{Implicit Method: } T_i^n = -\frac{\Delta t}{\Delta x^2}T_{i-1}^{n+1} + (1 + 2\frac{\Delta t}{\Delta x^2})T_i^{n+1} - \frac{\Delta t}{\Delta x^2}T_{i+1}^{n+1} \end{array}}$$

BCs: $\frac{T_{i+1}^n - T_i^n}{\Delta x} = 0$ at $x=0 \rightarrow$ $\boxed{T_1^n = T_2^n}$

$\frac{T_i^n - T_{i-1}^n}{\Delta x} = -\frac{Lh}{k}T_i^n$ at $x=L \rightarrow$ $\boxed{-T_{N-1}^n + T_N^n(1 + \frac{Lh}{k}\cdot\Delta x) = 0}$

c) Using $\frac{T_i^{n+1}-T_i^n}{\Delta t} = \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \leftarrow T_i^n = \varepsilon_i^n = e^{at}e^{ik_m x}$ (Explicit Method)

$$\frac{e^{a(t+\Delta t)}e^{ik_m x} - e^{at}e^{ik_m x}}{\Delta t} = \frac{1}{\Delta x^2}\left(e^{at}e^{ik_m(x+\Delta x)} - 2e^{at}e^{ik_m x} + e^{at}e^{ik_m(x-\Delta x)}\right)$$

$$\frac{e^{a\Delta t}-1}{\Delta t} = \frac{1}{\Delta x^2}\left(e^{ik_m\Delta x} - 2 + e^{-ik_m\Delta x}\right)$$

$$e^{a\Delta t} = 1 + \frac{\Delta t}{\Delta x^2}\left(e^{ik_m\Delta x} + e^{-ik_m\Delta x} - 2\right) \leftarrow e^{ik_m\Delta x} + e^{-ik_m\Delta x} = 2\cos(k_m\Delta x)$$

$$e^{a\Delta t} = 1 + 2\frac{\Delta t}{\Delta x^2}\left(\cos(k_m\Delta x) - 1\right) \leftarrow \cos(k_m\Delta x) - 1 = -2\sin^2\left(\frac{k_m\Delta x}{2}\right)$$

$$e^{a\Delta t} = 1 - 4\frac{\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right)$$

$$\left|\frac{\varepsilon_i^{n+1}}{\varepsilon_i^n}\right| = \left|e^{a\Delta t}\right| = \left|1 - 4\frac{\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right)\right| \le 1$$

$$0 \le \frac{2\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right) \le 1 \rightarrow \boxed{\frac{\Delta t}{\Delta x^2} \le \frac{1}{2}}$$ Stability Criterion for the explicit method

$\sin^2(x)$ is bounded between 0 and 1

Using $\frac{T_i^{n+1}-T_i^n}{\Delta t} = \frac{1}{\Delta x^2}\left(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}\right)$ (Implicit Method)

Let $T_i^n = \varepsilon_i^n = e^{at}e^{ik_m x}$

$$\frac{e^{a(t+\Delta t)}e^{ik_m x} - e^{at}e^{ik_m x}}{\Delta t} = \frac{1}{\Delta x^2}\left(e^{a(t+\Delta t)}e^{ik_m(\Delta x+x)} - 2e^{a(t+\Delta t)}e^{ik_m x} + e^{a(t+\Delta t)}e^{ik_m(x-\Delta x)}\right)$$

$$\frac{e^{a\Delta t}-1}{\Delta t} = \frac{1}{\Delta x^2}\left(e^{a\Delta t}e^{ik_m\Delta x} - 2e^{a\Delta t} + e^{a\Delta t}e^{-ik_m\Delta x}\right) \leftarrow \begin{cases} e^{ik_m\Delta x} + e^{-ik_m\Delta x} = 2\cos(k_m\Delta x) \\ \cos(k_m\Delta x) - 1 = -2\sin^2\left(\frac{k_m\Delta x}{2}\right) \end{cases}$$

$$\cancel{e^{a\Delta t} = \frac{\Delta t}{\Delta x^2}(e^{a\Delta t})}$$

$$e^{a\Delta t} = 1 - 4e^{a\Delta t}\frac{\Delta t}{\Delta x^2}\left(\sin^2\left(\frac{k_m\Delta x}{2}\right)\right)$$

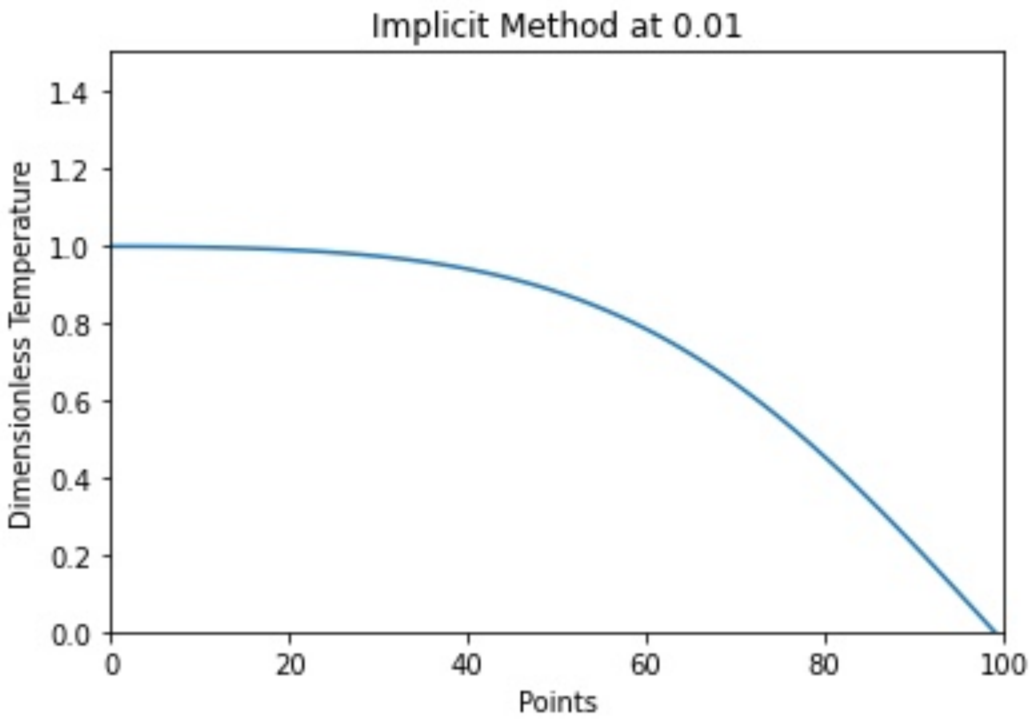
$$e^{a\Delta t} = \left(1 + 4\frac{\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right)\right)^{-1}$$

$$\left|e^{\Delta t}\right| = \left|\left(1 + 4\frac{\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right)\right)^{-1}\right| \le 1$$

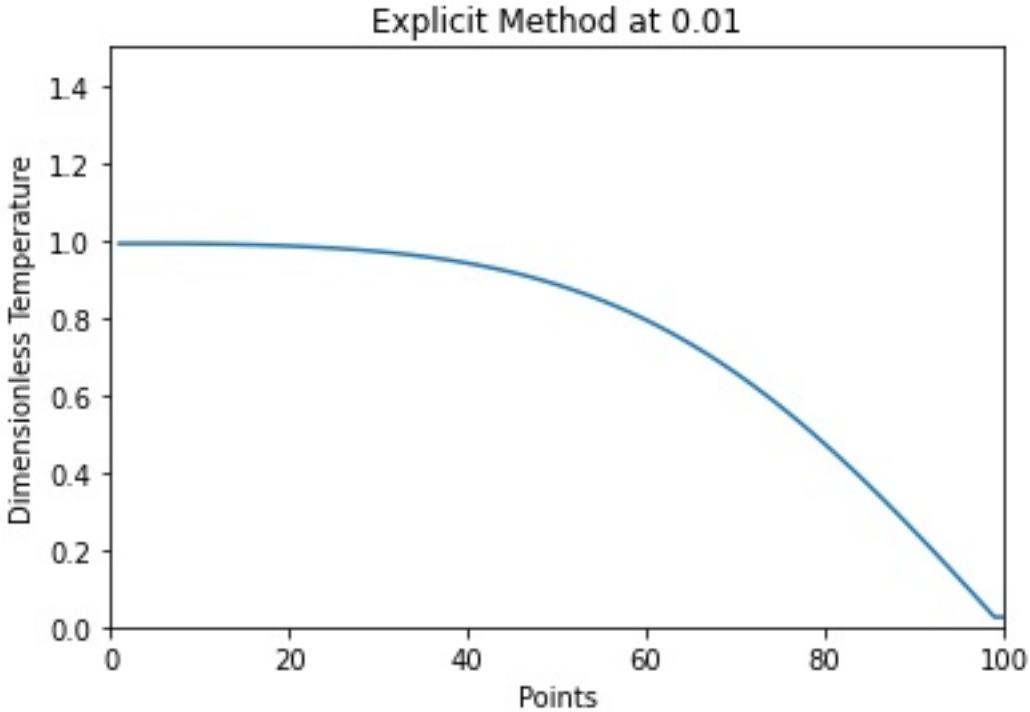
→ This expression has to be less than 1. As it is similar to the form $\frac{1}{x}$, for $\frac{1}{x}$ to be ~~bettend~~ less than 1, $x \ge 1$. As $1 + 4\frac{\Delta t}{\Delta x^2}\sin^2\left(\tfrac{1}{2}k_m\Delta x\right)$ has a minimum value of 1, the implicit scheme is always numerically stable

Ansys Fluent Section

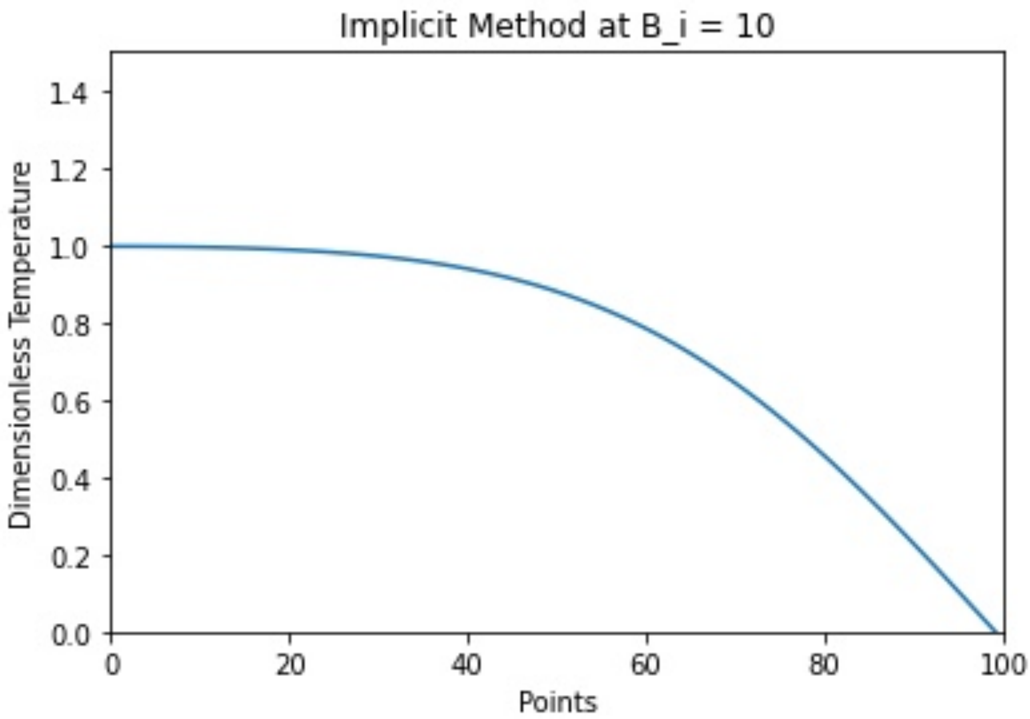
My simulation did not end up working and I could not find out how many particles went through the exhaust channel/engine duct. Sorry.

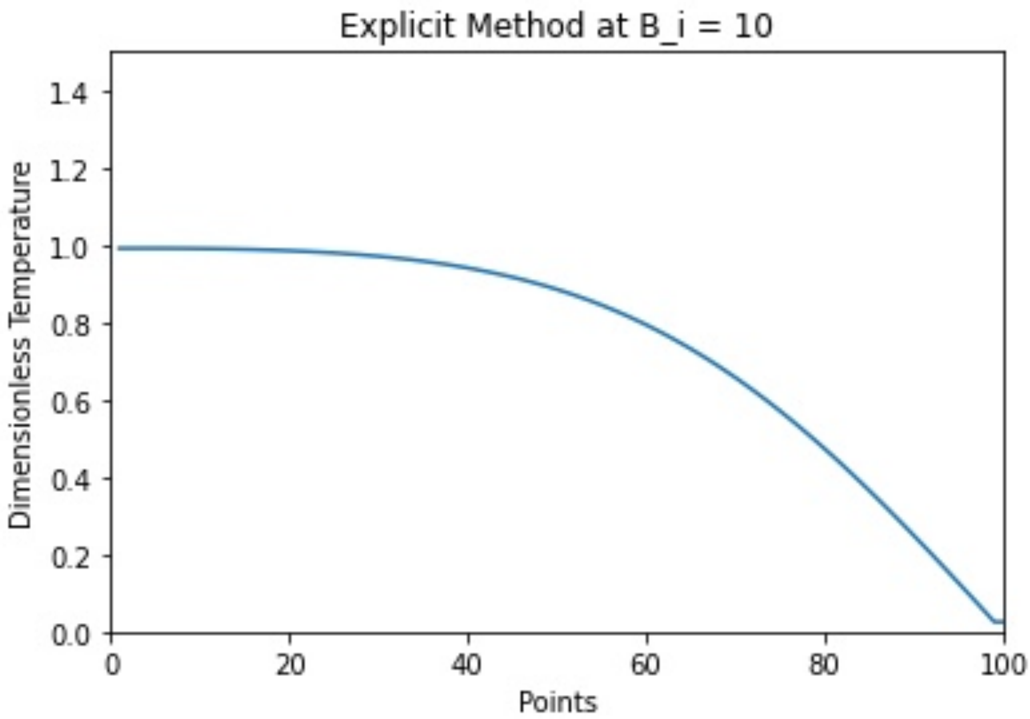


Implicit Method at 0.01
Dimensionless Temperature
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
Points



Explicit Method at 0.01
Dimensionless Temperature
Points
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100


Implicit Method at B_i = 10
Dimensionless Temperature
Points
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
20
40
60
80
100



Explicit Method at B_i = 10
Dimensionless Temperature
Points
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tues Apr 19 17:39:00 2022

@author: ryanhuang
"""

import matplotlib.pyplot as plt
import numpy as np
from numpy.linalg import inv


# number of gridpoints
N = 100

# boundary conditions
T1, T2 = 0, 0

# time step and space step
DT, DX = 0.00001, 0.01

r = DT / (DX**2)   # r = 0.1

# constants
T_i     = 1
B_i     = 10


def heat_eq(T1, T2, T3, non_homo):
    '''
    Discretized heat equation to calculate the next temperature at a specific grid point
    '''
    if non_homo:
        new_T = r * T1 + (1 - 2*r) * T2 + r * T3
    else:
        new_T = r * T1 + (1 - 2*r) * T2 + r * T3 - 2 * DT

    return new_T


def update(temperatures, non_homo):
    '''
    Input a list of temperature values

    Output a new list of temperature values for the next timestep based on the discretized heat

    '''
    temp = []

    for i in range(1, len(temperatures) - 1):
        if i - 1 == 0:
            temp.append(heat_eq(T1, temperatures[i], temperatures[i+1], non_homo))
```

```
        elif i + 1 == 100:
            temp.append(heat_eq(temperatures[0], temperatures[i], T2, non_homo))
        else:
            temp.append(heat_eq(temperatures[i-1], temperatures[i], temperatures[i+1], non_homo)

    return temp


def main():
    '''
    ---------------------------------------------------------------------------
    implicit method

    '''



    # setup the RHS and LHS matrices
    a = np.zeros((N+1, N+1))
    b = np.zeros((N+1,1))
    b.fill(T_i)


    for i in range(0, N-1):
        a[i+1][i] = -r
        a[i+1][i+1] = (1 + 2*r)
        a[i+1][i+2] = -r


    # boundary conditions
    a[0][0] = 1
    a[0][1] = -1
    a[1][0] = 0
    a[1][1] = 1 + r

    a[N][N-1] = 1
    a[N][N] = 1 + DX*B_i


    a_inv = inv(a)


    for i in range(5000):
        new = np.matmul(a_inv, b)
        b = new
        b[0] = b[1]
        print(i)


    plt.figure(1)
    plt.title(f"Implicit Method at B_i = {B_i}")
    plt.xlabel("Points")
    plt.ylabel("Dimensionless Temperature")
    plt.plot(list(range(len(b.transpose()[0]))), b.transpose()[0])
    plt.axis([0, N, 0, 1.5])
    plt.show()
```

```
    '''
    ---------------------------------------------------------------------------
    explicit method

    '''

    # initialize temperature values at T_i
    T = [T1] + [1 for i in range(N - 2)] + [T2]

    # run for 5000 steps
    for i in range(5000):

        new = [T1] + update(T, non_homo = True) + [T2]
        T = new
        T[0] = T[1]

        T[N-1] = T[N-2]/(1+0.01*DX)

    plt.figure(2)
    plt.title(f"Explicit Method at B_i = {B_i}")

    # plot the final temperature distribution
    plt.plot(list(range(1,N+1)), list(reversed(T)))
    plt.axis([0, N, 0, 1.5])
    plt.xlabel("Points")
    plt.ylabel("Dimensionless Temperature")
    plt.show()



if __name__ == "__main__":
    main()
```

PDF document made with CodePrint using Prism