Computational Engineering - Engr 8103
Problem Set #8

Allen Spain
avs81684@uga.edu

University of Georgia — 26 November 2019

Problem 1

(10 pts.) Consider the following transport PDE

$$\begin{aligned} u_t &= D_{u_{xx}} - F u_x \\ u(0,x) &= \frac{2x}{1+x^4} \quad 0 \le x \le 20 \\ u(t,0) &= 0 \quad 0 \le t \le 3 \\ u(t,20) &= 0 \quad 0 \le t \le 3 \end{aligned}$$

Where $u(t,x)$ represents the drug concentration at time t seconds and x inches away from a reference point on the vein, located slightly behind the point of injection. $u(0,x)$ represents the initial drug concentration through the blood vein a moment after the injection:

Discretize this PDE using backward time for ut, backward space for ux and the regular second derivative method for uxx, the same one we used in class for the diffusion equation. Assuming $D = 0.5$,$F = 2$,$dt = 2$ and $dx = 4$,write down the matrix M where:

$$M \begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ u_5^{n+1} \end{bmatrix} = b$$

Discretizing using $B_t B_x$

$$u_t = u_k^n - u_k^{n-1} u_x = u_n^k - u_{k-1}^n u_{xx} = u_{k-1}^n - 2u_k^n + u_{k+1}^n$$

Given that

$$u_t = Du_{xx} - Fu_x$$

The aforementioned discretization u_t yields:

$$\begin{aligned} u_t &= \frac{u_k^n - u_k^{n-1}}{dt} \\ u_k^n - u_k^{n-1} &= \frac{Ddt}{(dx)^2}[u_{k-1}^n - 2u_k^n + u_{k+1}^n] - \frac{Fdt}{dx}[u_k^n - u_{k-1}^n] \end{aligned}$$

Define: $L_1 = \frac{Ddt}{(dx)^2}$ and $L_2 = \frac{Fdt}{dx}$

$$\therefore u_k^{n-1} = -(L_1 + L_2)u_{k-1}^{n+1} + (2L_1 - L_2 + 1)u_k^n - L_1 u_{k+1}^n$$

$$\text{Assume: } n \to n+1$$

$$L_1 = 0.5(\frac{1}{2}) = \frac{1}{4}$$

$$L_2 = 2(\frac{1}{2}) = 1$$

$$u_k^n = -(L_1 + L_2)u_{k-1}^{n+1} + (2L_1 - L_2 + 1)u_k^n - L_1 u_{k+1}^n$$

So the matrix M can be defined as:

$$\begin{bmatrix} (L_2 + L_2 - 1) & -L_1 & 0 & 0 \\ -(L_1 + L_2) & (L_2 + L_2 - 1) & -L_1 & 0 \\ 0 & -(L_1 + L_2) & (2L_1 + L_2 + -1) & -L_1 \\ 0 & 0 & -(L_1 + L_2) & (L_2 + L_2 - 1) \end{bmatrix}$$

(b) (10 pts.) Write a Matlab code to solve this PDE using the discretization you developed in part (a). Use $D = 0.5$, $F = 2$, $dt = 0.02$, $dx = 0.02$. Note that since backward time methods tend to be more stable, you can afford to use a finer mesh for the space dimension (smaller dx values.) Your code should plot the initial drug concentration and the drug concentrations after one, two and three seconds on the same figure. In other words, plot $u(0, x)$, $u(1, x)$, $u(2, x)$ and $u(3, x)$ vs x. Identify each plot using a legend. Include a hard copy of this figure with your HW solutions.

(b)

```
function drug_Bt
    dt = 0.02;
    dx = 0.02;

    x = 0:dx:20;
    t = 0:dt:3;

    Tmax = length(t);
    Xmax = length(x);

    D = 0.5;
    F = 2.0;

    L1 = D*dt/(dx)^2;
    L2 = dt/dx;

    u(:,1) = 0;
    u(:,Xmax) = 0;

    u(1 ,:) = (2.*x)./(1 + (x.^4)); plot(x,u(1 ,:) , 'linewidth',2);
        axis ([0 10 0 1]);
    hold on;

    for n = 1:(Tmax-1) for k = 2:(Xmax - 1)
        u(n,k) = -(L1 + L2)*u(n+1,k-1) + (2*L1 - L2 + 1)*u(n+1,k) -L1*u(
            n+1,k)
    end
        graph = n*dt
        if (graph==1 || graph==2 || graph==3)
            axis ([0 10 0 2]);
            plot(x,u(n,:),'linewidth',2);
            hold on;
        end
    end
```

```
title('u(t,x) vs x');
legend( 'u(0,x)' , 'u(1,x)', 'u(2,x)' ,'u(3,x)')
```

Had some errors coding despite the math seeming correct

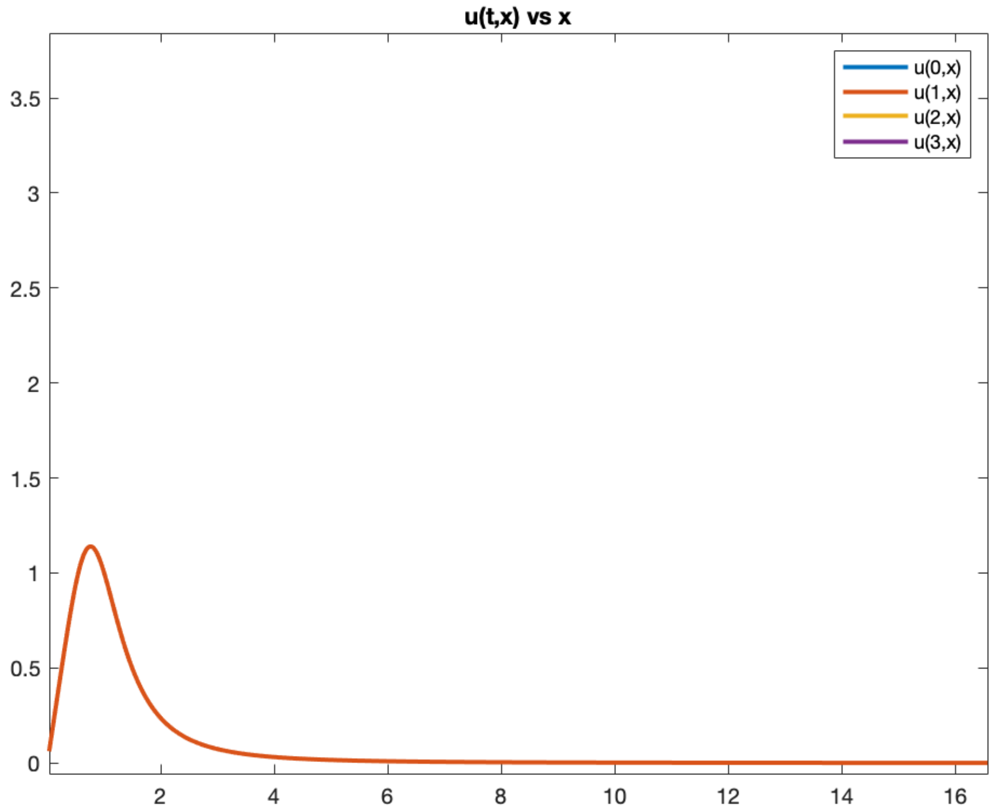

Figure 1: 1b

(c) (5 pts.) Now assume faster blood flow; that is $F = 6$. Repeat part (b) for this case. Does the figure look realistic? Why not? Explain. - The concetration would have an abnormal retractio rate, and would therefore diminish faster then realistic.

Problem 2

(10 pts.) Consider the same PDE where one of the boundary conditions is changed from Dirichlet to Neumann:

$$M \begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ u_5^{n+1} \\ u_6^{n+1} \end{bmatrix} = b$$

(a)

$$\begin{bmatrix} (L_2+L_2-1) & -L_1 & 0 & 0 \\ -(L_1+L_2) & (L_2+L_2-1) & -L_1 & 0 \\ 0 & -(L_1+L_2) & (2L_1+L_2+-1) & -L_1 \\ 0 & 0 & -(L_1+L_2) & (L_2+L_2-1) \\ 0 & 0 & 0 & -(L_1+L_2) \end{bmatrix}$$

(b)

```
function drug_Bt_Nbc
    dt = 0.02;
    dx = 0.02;

    x = 0:dx:20;
    t = 0:dt:3;

    Tmax = length(t);
    Xmax = length(x);

    D = 0.5;
    F = 2.0;

    L1 = D*dt/(dx)^2;
    L2 = dt/dx;

    u(:,1) = 0;
    u(:,Xmax) = 0;

    u(1 ,:) = (2.*x)./(1 + (x.^4)); plot(x,u(1 ,:) , 'linewidth',2);
       axis ([0 10 0 1]);
    hold on;

    for n = 1:(Tmax-1) for k = 2:(Xmax - 1)
        u(n,k) = -(L1 + L2)*u(n+1,k-1) + (2*L1 - L2 + 1)*u(n+1,k) -L1*u(
           n+1,k)
    end
        graph = n*dt
        if (graph==1 || graph==2 || graph==3)
            axis ([0 10 0 2]);
            plot(x,u(n,:),'linewidth',2);
            hold on;
        end
    end
    title('u(t,x) vs x');
    legend( 'u(0,x)' , 'u(1,x)', 'u(2,x)' ,'u(3,x)')
```

Had some errors coding despite the math seeming correct