

## Sample Partial Differential Equation

Suppose given the PDE (Diffusion)

$$\partial_t u = D \partial_x^2 u, \text{ for } x \in [0, 1], \text{ and } t > 0$$

Exact Solution (can be found in multiple PDEs textbooks):

$$u(x, t) = \sum_{k=1}^{\infty} \frac{4}{(k\pi)^2} \sin\left(\frac{k\pi}{2}\right) \sin(k\pi x) e^{-0.5(k\pi)^2 t}$$

We are going to compute the numerical solution with the given information below.

for  $x \in [0, 1]$  and  $t > 0$  :

$$\text{the initial condition } u(x, 0) = \begin{cases} x, & \text{if } 0 \leq x \leq 1/2 \\ 1 - x, & \text{if } 1/2 < x \leq 1 \end{cases}$$

the boundary condition  $u(0, t) = u(1, t) = 0$

The Explicit Difference Schema:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}, \text{ where } D = \frac{1}{2}$$

Take  $\Delta x = 0.1, \Delta t = 1/100$  to proceed the calculations;

## Solving Idea for a particular t

Use of the finite difference scheme for discretizing partial derivatives reduces the problem to vector-matrix multiplication.

Where the unknown vector  $U^n = (U_1^n, U_2^n, \dots, U_N^n)^T$  refers to sample of grid values of the solution function  $u(x, t)$  evaluated at  $t = t_n$ .

Then what we focus on for  $u_j^n = u(x_j, t_n)$  is in a vector of  $U^n$  to  $U^{n+1}$

To calculate the  $U^{n+1}$ , we must use the explicit difference scheme

## A Look of Explicit Difference Scheme Equation

As we said we interested the vector  $U^{n+1}$  = a set of  $u_j^{n+1}$   $j \in [1, 11]$ ,  
for the case that  $\Delta x = 0.1, x \in [0, 1]$

Hence we would like to rearrange the equation to:

$$\begin{aligned} u_j^{n+1} &= \frac{1}{2} \frac{\Delta t}{(\Delta x)^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) + u_j^n \\ &= u(x_j, t_{n+1}) = \frac{1}{2} \frac{\Delta t}{(\Delta x)^2} [u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)] + u(x_j, t_n) \end{aligned}$$

for easier programming purpose, we shift the index of t left by one

$$= u(x_j, t_n) = \frac{1}{2} \frac{\Delta t}{(\Delta x)^2} [u(x_{j+1}, t_{n-1}) - 2u(x_j, t_{n-1}) + u(x_{j-1}, t_{n-1})] + u(x_j, t_{n-1})$$

## A Look of How Initial and Boundary Condition are Important

The expression of  $u(x, t)$  in 2D matrix:

	1	2	3	...	$n$
1	$u(0, 0)$	$u(0, 1\Delta t)$	$u(0, 2\Delta t)$	...	$u(0, n\Delta t)$
2	$u(1\Delta x, 0)$	...	...	...	$u(1\Delta x, n\Delta t)$
3	$u(2\Delta x, 0)$	...	...	...	$u(2\Delta x, n\Delta t)$
	.				
	.				
11	$u(1, 0)$				

The initial condition given the cover the values for first columns

And the boundary condition cover the values for first and last rows

Now, the expression of  $u(x, t)$  should be:

	1	2	3	...	$n$
1	0	0	0	...	0
2	0.1	$u(1\Delta x, 1\Delta t)$	...	...	$u(1\Delta x, n\Delta t)$
3	0.2	$u(2\Delta x, 1\Delta t)$	...	...	$u(2\Delta x, n\Delta t)$
	.				
5	0.5	...	...	...	
6	0.4	...	...	...	
7	0.3	...	...	...	
	.				
11	0	0	0	...	0

Now from this matrix, we see the pattern of how explicit difference help us to calculate the  $u(x_j, t_n)$ ;

For example:  $u(x_2, t_2)$ , we need  $u(x_1, t_1)$ ,  $u(x_2, t_1)$ ,  $u(x_3, t_1)$ 's values to explicit difference equation:

$$u(x_j, t_n) = \frac{1}{2} \frac{\Delta t}{(\Delta x)^2} [u(x_{j+1}, t_{n-1}) - 2u(x_j, t_{n-1}) + u(x_{j-1}, t_{n-1})] + u(x_j, t_{n-1})$$

Hence we know we must calculated the current columns's val then proceed to proceed to the row,  $U^n$  = each columns value, and  $U^{n+1}$  = next columns value

## MATLAB Implementation

```
clear
close all

prompt = 'Enter the value of  $\Delta t$ : ';
% set t_max= x then coloumns needed = x / delta_t + 1
% row = 1/ (0.01) + 1 = 11 (by default x[0,1]
delta_t = input(prompt);

prompt1 = 'Enter the value of T(max): ';
column = input(prompt1) / delta_t + 1 ;
row = 11;

% 2-D array with all zeros for exact solution
arr1 = zeros(row,column);

% initial variables for exact solution
result = 0;
arr1_x = 0;
arr1_t = 0;

% exact solution
% taking 14 terms
for i = 1:row
    for j = 1:column
        for k = 1:14
            result = result +
4/(k*pi)^2*sin(k*pi/2)*sin(k*pi*arr1_x)*exp(-0.5*(k*pi)^2*arr1_t);
        end
        arr1(i,j) = result;
        arr1_t = arr1_t + delta_t;
        result = 0;
    end
    arr1_t = 0;
    arr1_x = arr1_x + 0.1;
end

% Numerical Computing Solution
% (Initial Condition)
arr2 = zeros(row,column);
x_num = 0;
for i = 1 : row
    if(x_num>0.5)
        arr2(i,1) = 1-x_num;
    else
        arr2(i,1) = x_num;
    end
    x_num = x_num + 0.1;
```

```

end

% applied explicit differences to calculate other points
% applied boundary condition as well
dt = delta_t;
dx = 0.1;
for j = 2: column
    for i = 1:row
        if (i == 1 || i == row)
            arr2(i,j) = 0;
        else
            arr2(i,j) = 0.5 *dt/(dx)^2*(arr2(i+1, j-1)-2*arr2(i,j-1)+arr2(i-1,j-1))+arr2(i,j-1);
        end
    end
end

% Error |Exact - Numerical|
error = zeros(row, column);
for i = 1 : row
    for j = 1 : column
        error(i,j) = abs(arr1(i,j)- arr2(i,j));
    end
end

%% plot Portion
x_ = 0:0.1:1;

prompt_1 = 'Enter the value of t to get the graph u(x,t) at t =: ' ;
t = input(prompt_1);
exctVal = arr1(:, t/delta_t+1 );
numVal = arr2(:, t/delta_t+1 );
plot (x_,numVal,x_,exctVal,'lineWidth', 2);
hold off
xlabel('x');
s = num2str(t);
txt = ['u(x,t), t= ', s];
ylabel(txt);
legend('Numerical','Exact');

```

## Graphical output

// solved

Enter the value of  $\Delta t$ : 0.01

Enter the value of  $T(\max)$ : 5

Enter the value of  $t$  to get the graph  $u(x, t)$  at  $t =$ : 1

