## 11.3: Heat Equation in 1D

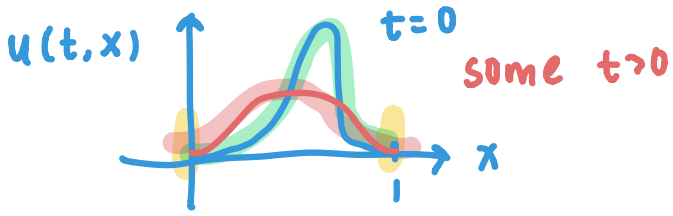
We consider a simple heat equation, where the unknown function $u(t, x)$ is the temperature at time $t$ at the point $x$, in a rod of unit length. This equation has the form

"partial derivatives"

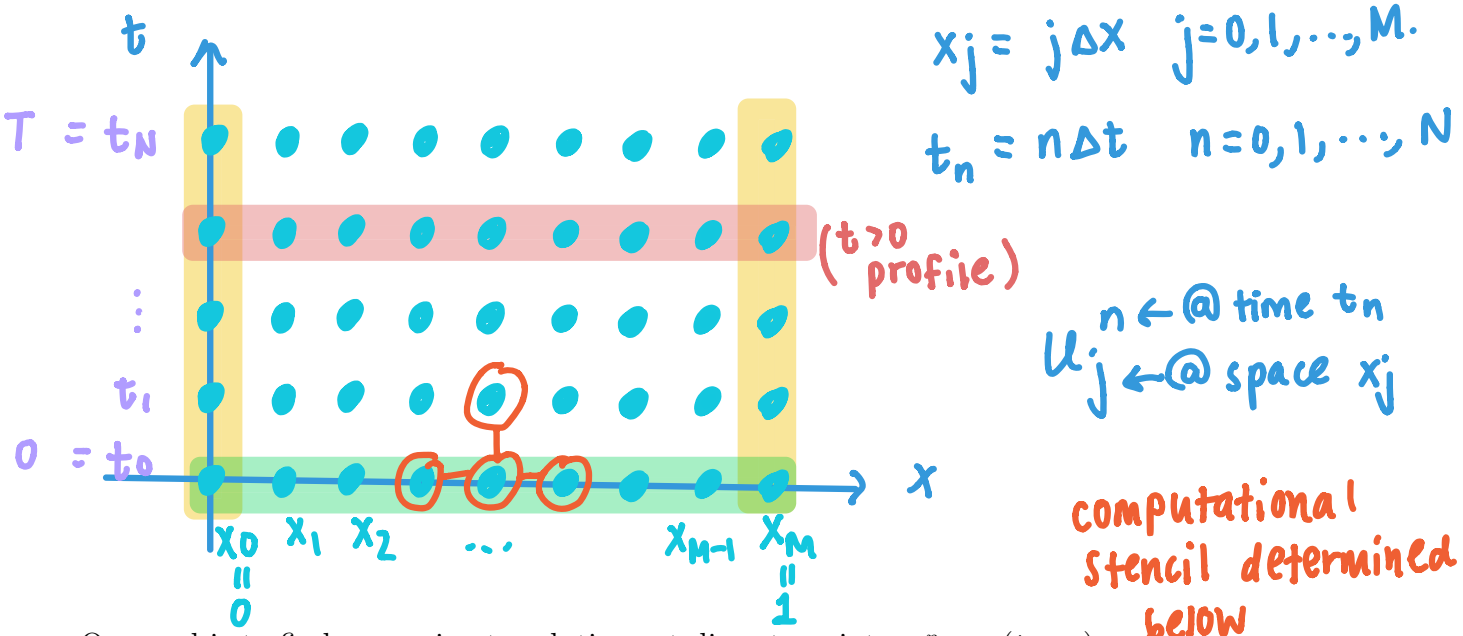
$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} \quad \Longleftrightarrow \quad u_t = u_{xx} \quad 0 \le x \le 1 \quad \text{for } t \in [0, T]\,.$$

How many conditions do we need to impose to solve this? Hint: what's the 'order of the equation' with respect to $x$? And $t$?

- 2nd order eqn. in $x$: 2 Boundary cond. for $x$ $\rightarrow$ $u(t,0) = u(t,1) = 0$ $t > 0$
- 1st order eqn. in $t$: 1 initial cond. for $t$ $\rightarrow$ $u(0,x) = f(x)$ for $0 \le x \le 1$




We want to find the temperature $u(t, x)$ for $t \in [0, T]$ and $x \in [0, 1]$. To solve this problem, we set up a uniform grid on the rectangle $[0, T] \times [0, 1]$.




Our goal is to find approximate solutions at discrete points, $u_j^n \approx u(t_n, x_j)$.

Heat eqn.: $u_t = u_{xx}$



## 1. Forward Euler scheme for the heat equation

We use a central finite difference for the double derivative in space, and forward Euler for the time derivative. What do these look like, and what order accuracy do they have?

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{t=t_n, x=x_j} \approx u_{xx}(t_n, x_j) = \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{(\Delta x)^2}$$

$$\left.\frac{\partial u}{\partial t}\right|_{t=t_n, x=x_j} \approx u_t(t_n, x_j) = \frac{u_j^{n+1} - u_j^n}{\Delta t}$$

What do we get for our approximation when we substitute those into the PDE?

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

- given $u_j^n$, want to solve for $u_j^{n+1}$:

$$u_j^{n+1} = u_j^n + \Delta t \cdot \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{(\Delta x)^2}.$$

- define $\gamma = \frac{\Delta t}{(\Delta x)^2}$

$$\Rightarrow \quad u_j^{n+1} = \gamma u_{j-1}^n + (1-2\gamma) u_j^n + \gamma u_{j+1}^n$$

To compute $u_j^{n+1}$, we only need values of $u$ at the previous time-step $n$ with the given formula. Algorithms where the *next* time step is given solely in times of the previous time-steps is called an **explicit method.**

**Example 1.1.** *Create some pseudo-code to approximate solutions $u(t,x)$ of*

$$\begin{cases} u_t = u_{xx} \quad \text{on } 0 \le x \le 1 \quad \text{for } t \in [0,T] \\ u(t,0) = u(t,1) = 0 \quad \text{for } t \ge 0, \\ u(0,x) = f(x) \end{cases} \qquad (*)$$

*with $\Delta x = 1/N$ and $\Delta t = T/M$.*

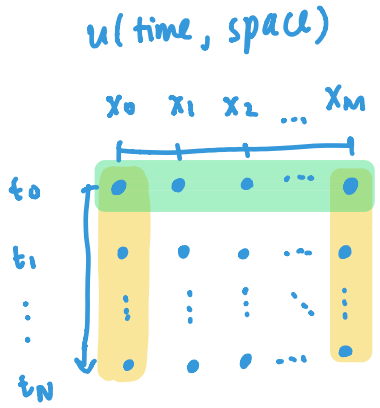
- specify initial & Boundary conditions, final time
- discretize mesh in space/time

x = linspace(0,1,M+1)  u = zeros(N+1, M+1)

t = linspace(0,T,N+1)

- implement Boundary conditions:

u(:,1) = 0
u(:,N+1) = 0  ( u(t,0) = 0, u(t,1) = 0 )




- implement initial condition:

u(1,:) = f(x(:))  ( u(0,x) = f(x) )

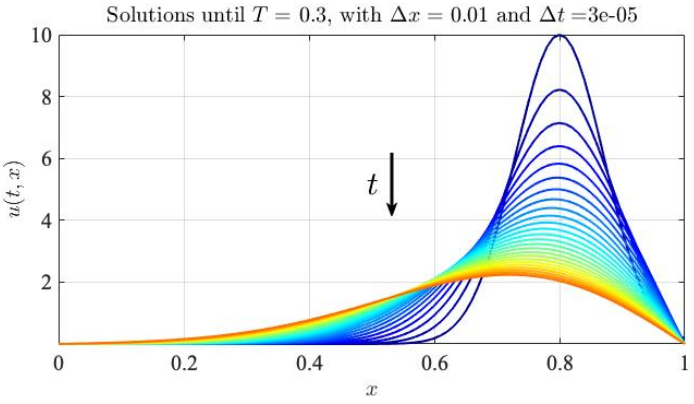
- set $\gamma = \frac{\Delta t}{(\Delta x)^2}$

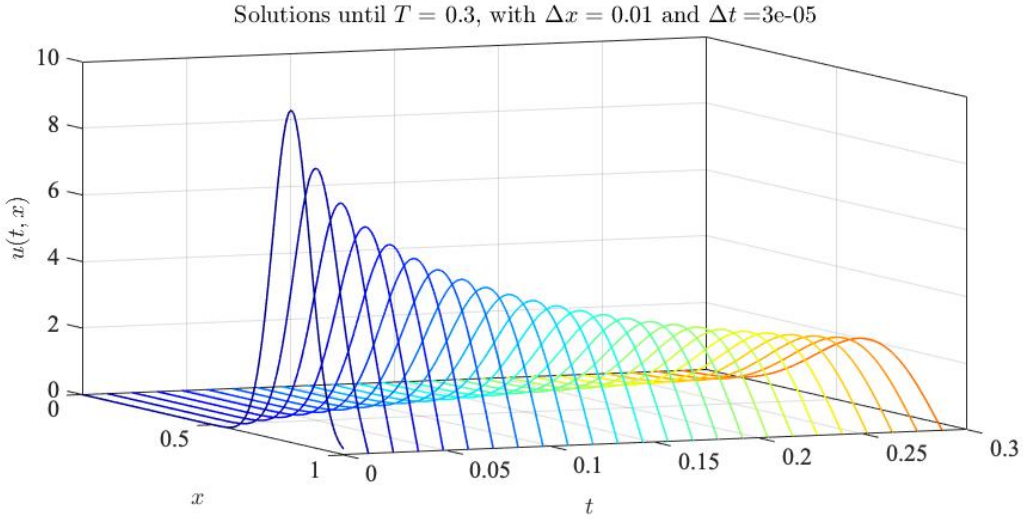
```
for n = 2 : N
    for j = 2 : M
        u(n,j) = γ·u(n-1,j-1) + (1-2γ)·u(n-1,j)
                              + γ·u(n-1,j+1)
    end
end
```

**Example 1.2.** *Use your code to approximate the solutions $u(t,x)$ to the problem:*

$$\begin{cases} u_t = .1\, u_{xx} \quad \text{on } 0 \le x \le 1 \quad \text{for } t \in [0,.3] \\ u(t,0) = u(t,1) = 0 \quad \text{for } t \ge 0, \\ u(0,x) = 10e^{-((x-.8)/.1)^2} \quad \text{for } 0 \le x \le 1, \end{cases}$$

*with the parameters:* $\Delta x = 1/100$ *and* $\Delta t = 3 \times 10^{-5}$.

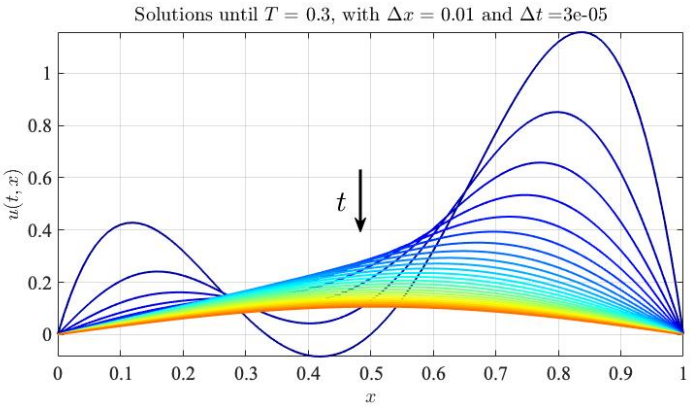


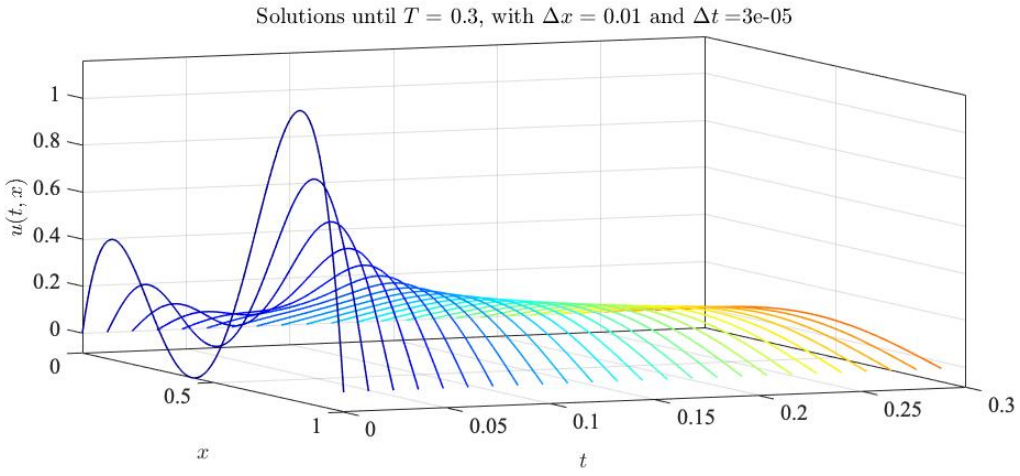



**Example 1.3.** *Use your code to approximate the solutions $u(t,x)$ to the problem:*

$$\begin{cases} u_t = .5\, u_{xx} \quad \text{on } 0 \le x \le 1 \quad \text{for } t \in [0,.3] \\ u(t,0) = u(t,1) = 0 \quad \text{for } t \ge 0, \\ u(0,x) = -50x(x-1)(x-\frac{1}{3})(x-\frac{1}{2}) \quad \text{for } 0 \le x \le 1, \end{cases}$$

*with the parameters:* $\Delta x = 1/100$ *and* $\Delta t = 3 \times 10^{-5}$.



## 2. Maximum principle for the heat equation

The solution $u(t,x)$ of the heat equation with the boundary conditions from $(*)$ satisfies the following maximum principle:

$$\min_{0\le y\le 1} u(t_1,y) \le u(t_2,x) \le \max_{0\le y\le 1} u(t_1,y) \qquad \text{for all } x,$$

for any $t_1 \le t_2$.

By looking at the maximum values, we can obtain a more 'intuitive' statement of the **maximum principle**:

$$\max_{0\le x\le 1} |u(t_2,x)| \le \max_{0\le x\le 1} |u(t_1,x)|$$

for any $t_1 \le t_2$. This means that the maximum value of $|u(t,x)|$ over $x$ is non-increasing in time $t$.

With Dirichlet boundary conditions set to 0, what does this say about the solution $u(t,x)$ as $t$ increases?

Since we're working with discrete approximations of the solution, we have to satisfy a **discrete version** of the maximum condition, namely

$$\max_j |u_j^{n+1}| \le \max_j |u_j^n| \quad \text{for every } n\,.$$

This is not always satisfied by numerical solutions. For some methods, we have to satisfy certain stability criteria for this condition to be imposed.

To satisfy the discrete maximum principle of the heat equation with the Dirichlet boundary conditions, we need to choose a time-step that satisfies:

$$\Delta t \le \frac{1}{2}(\Delta x)^2\,.$$

This is called a **stability condition.**

If $\Delta x = .01$, how small does $\Delta t$ have to be?

$$\Delta t \le \frac{1}{2}(.01)^2 = \frac{1}{2}(.001) = .0005$$