Tutorial 03:
Numerical Aspect I: Finite Differences

STEP 1: Logging onto the HPC cluster

- From a terminal/konsole:

```
ssh -X login@scp.chpc.ac.za
```

- Request one node with the alias command `qsubi1`

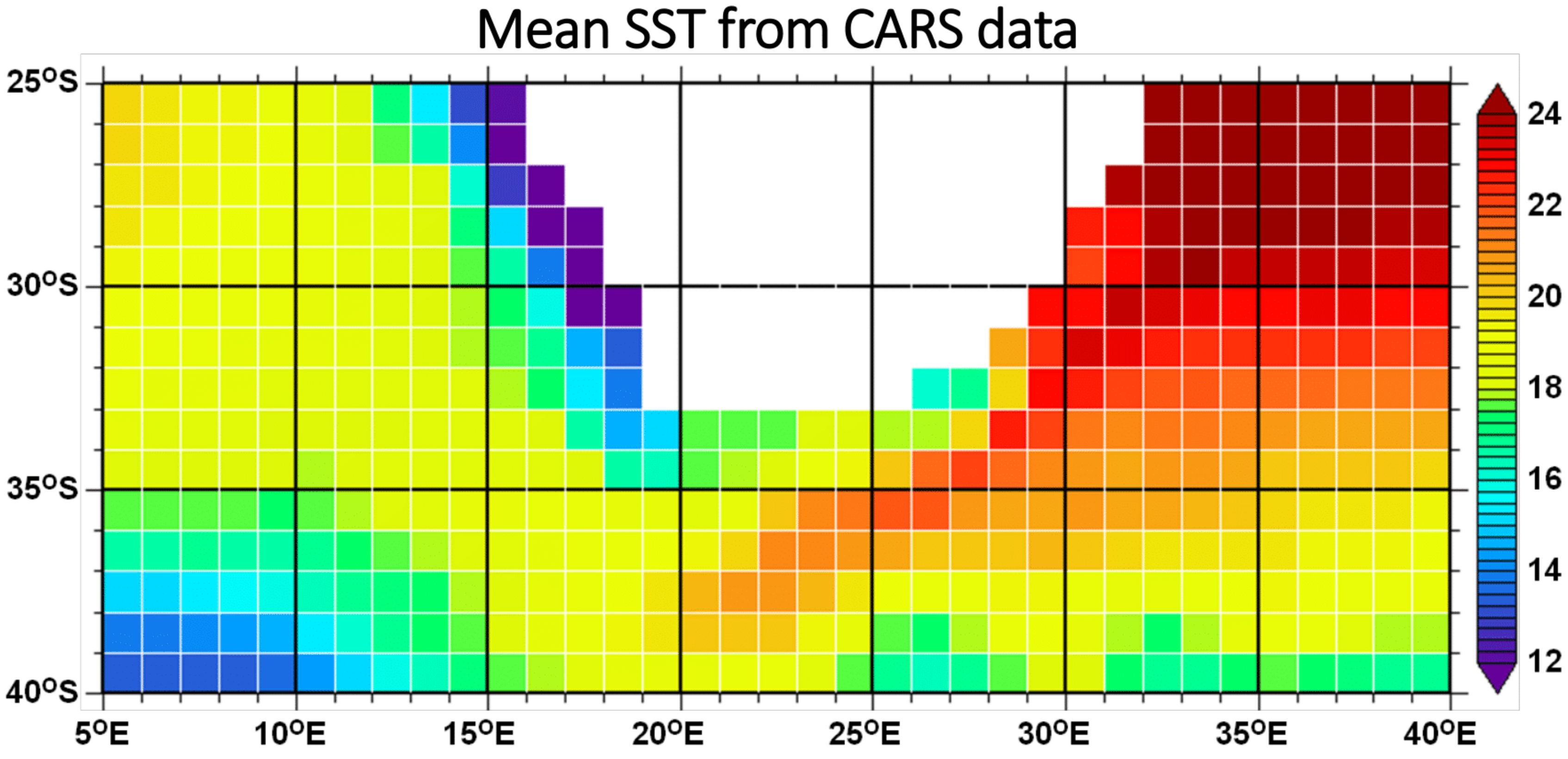
```
qsubi1
```

OBJECTIVES

- Analyse the temperature equation
- Admire my dream swimming pool
- Discretize the swimming into a regular a mesh grid
- Transform continuous derivatives by finite difference approximations
- Solve the 1D-Diffusion equation in MATLAB

The Grid

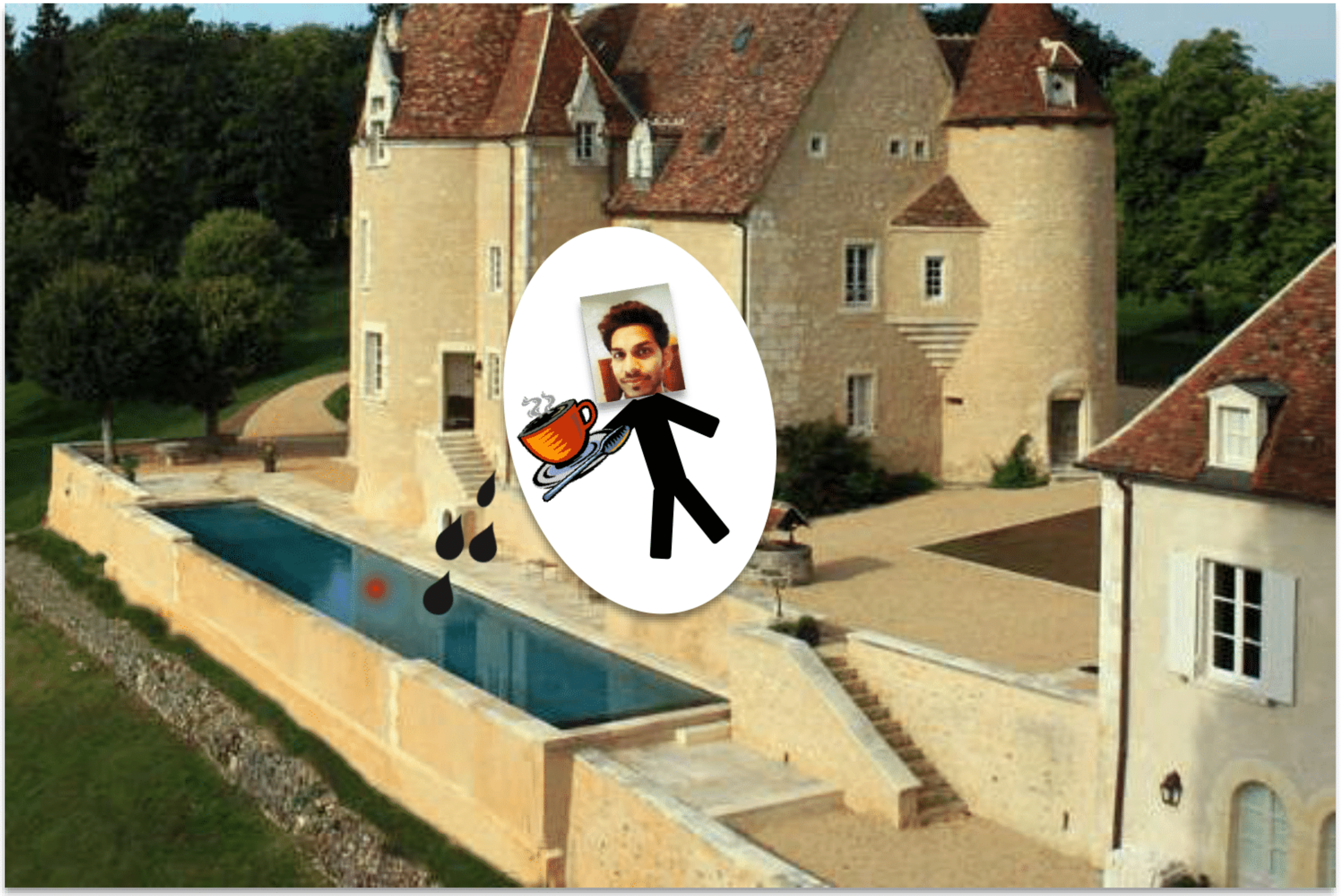
- Let's consider Temperature data (T) discretized on a grid :



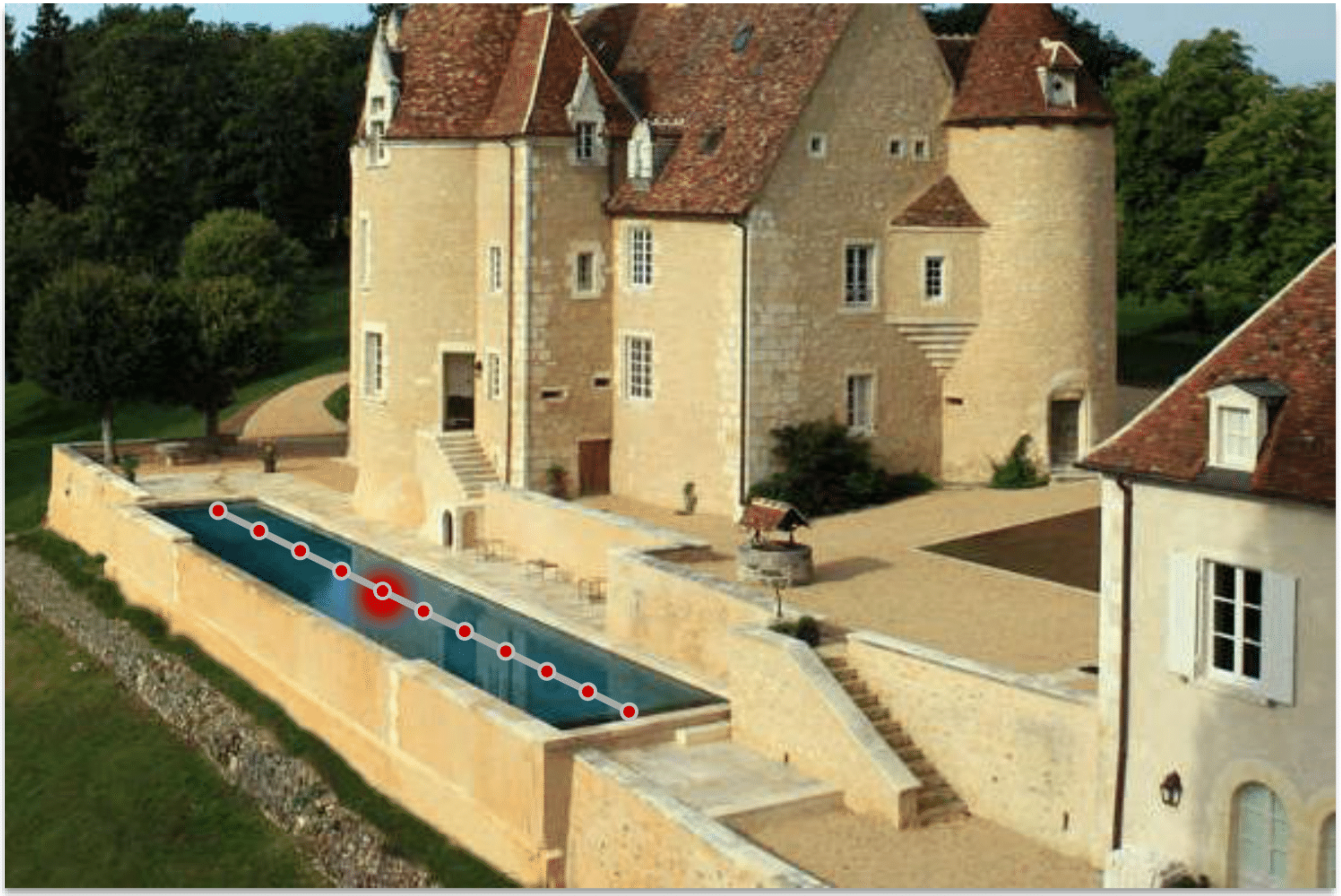
$$\frac{\partial T}{\partial t} + \vec{u}.\nabla T \;=\; \nabla_h \left(K_{Th}.\nabla_h T\right) + \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right)$$

My Dream Swimming Pool

My Dream Swimming Pool

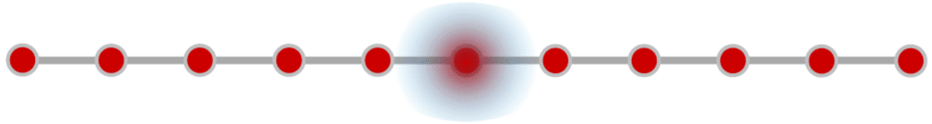


My Dream Swimming Pool



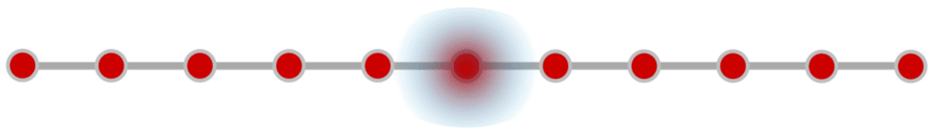
Solving the 1D Diffusion equation

- Let’s model how the temperature will evoluate in the swimming pool



Solving the 1D Diffusion equation

- Let’s model how the temperature will evoluate in the swimming pool

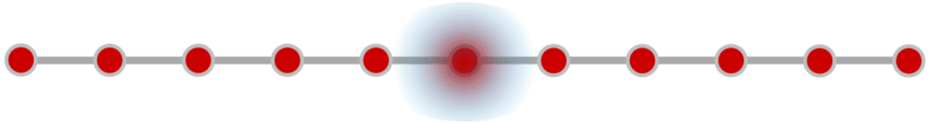


- To do so, you will solve the Temperature équation :

$$\frac{\partial T}{\partial t} + \vec{u}.\nabla T \;=\; \nabla_h \left(K_{Th}.\nabla_h T\right) + \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right)$$

Solving the 1D Diffusion equation

➢ Let’s model how the temperature will evoluate in the swimming pool



➢ To do so, you will solve the Temperature équation :

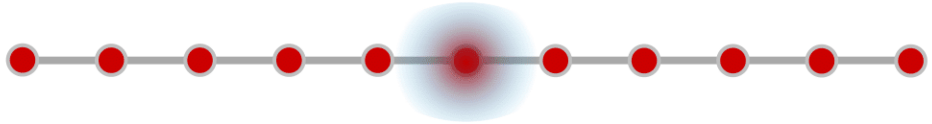
$$\frac{\partial T}{\partial t} + \vec{u}.\nabla T = \nabla_h \left(K_{Th}.\nabla_h T\right) + \frac{\partial}{\partial z}\left(K_{Tv}\frac{\partial T}{\partial z}\right)$$

➢ We can simplify the equation, under particular hypothesis:

- There is no currents in the swimming pool
- We do not consider variation of temperature with depth
  - ↳ It becomes a 1 Dimensional (1D) problem in the x coordinate
- The diffusion coefficient (K_{Th}) is a constant

Solving the 1D Diffusion equation

- Let's model how the temperature will evoluate in the swimming pool



- To do so, you will solve the 1D diffusion equation :

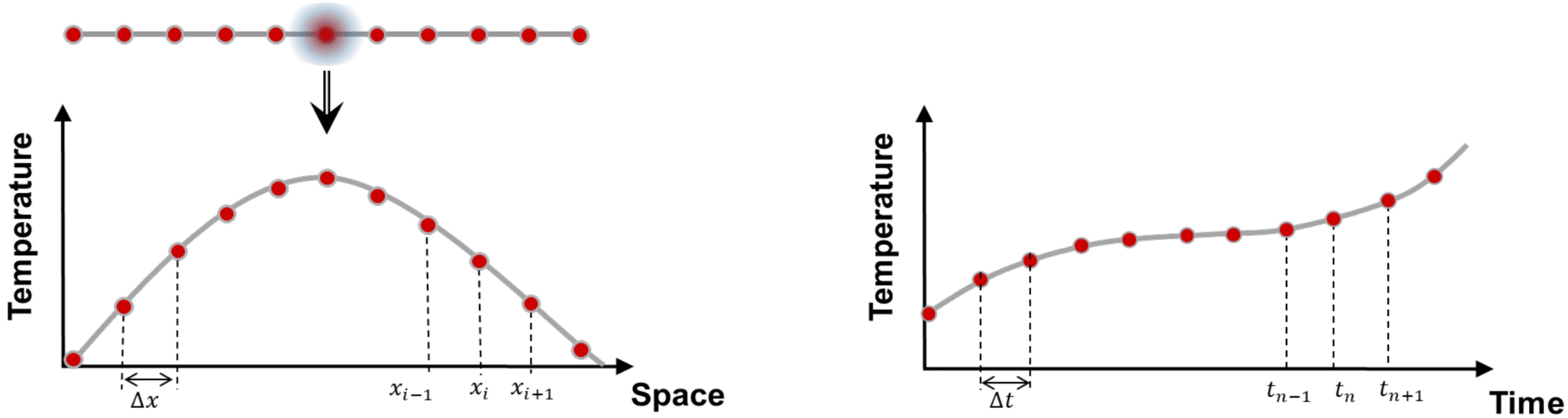
$$\frac{\partial T}{\partial t} = K_{Th} \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right)$$

- $\frac{\partial T}{\partial t}$ is the temperature increment during the period of time **dt**
- K_{Th} s the diffusion coefficient (ex: 0.001 m²/s)
- $\frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right)$ the **Laplacian** operator
plied to the temperature field with an horizontal scale **dx**

Finite Difference Definition

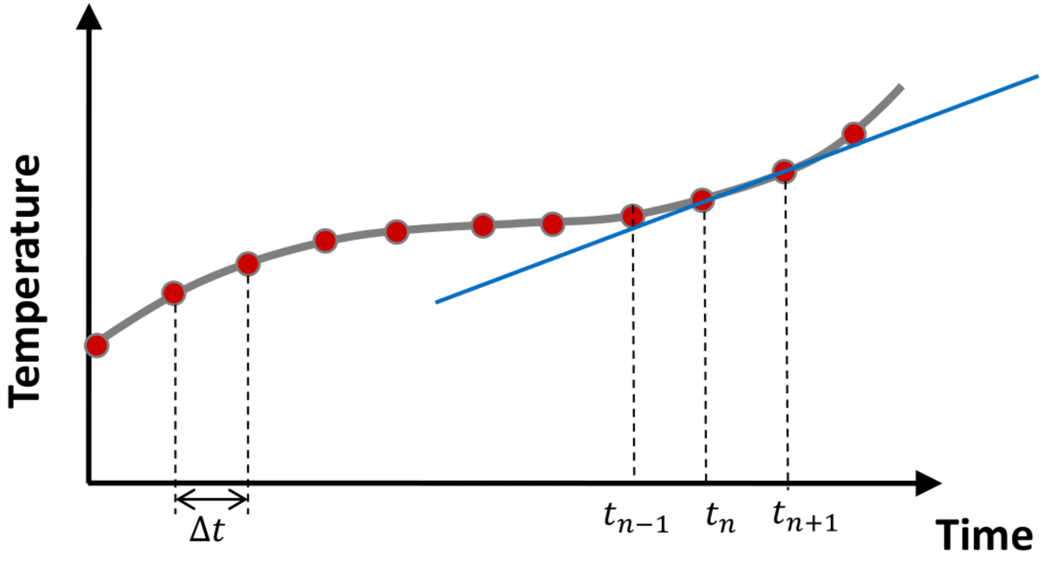
→ The first step in the discretization procedure is to replace the domain $[0, L] \times [0, T]$ by a set of mesh points. Here we apply equally spaced mesh points :

$$x_i = i\Delta x, i = 1, \dots, N_x \quad \text{and} \quad t_n = n\Delta t, n = 1, \dots, N_t$$



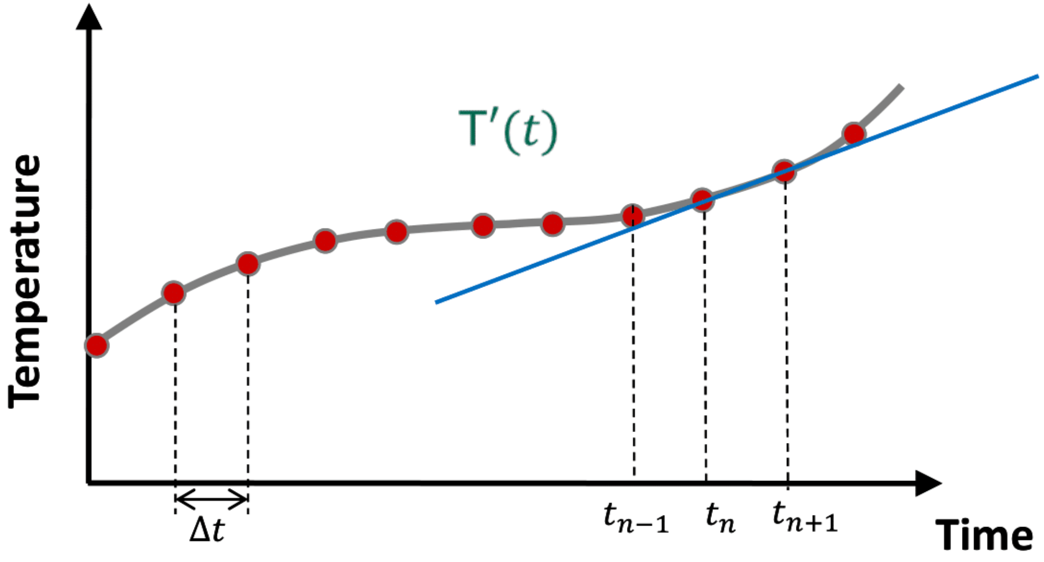
Finite Difference Definition

- Let's model the temporal derivative $\frac{\partial T}{\partial t}$



Finite Difference Definition

- Let's model the temporal derivative $\frac{\partial T}{\partial t}$

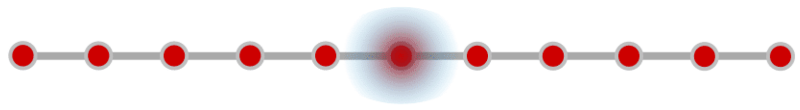


$$\frac{\partial T}{\partial t} = \frac{T_i^{n+1} - T_i^n}{\Delta t}$$

Forward difference

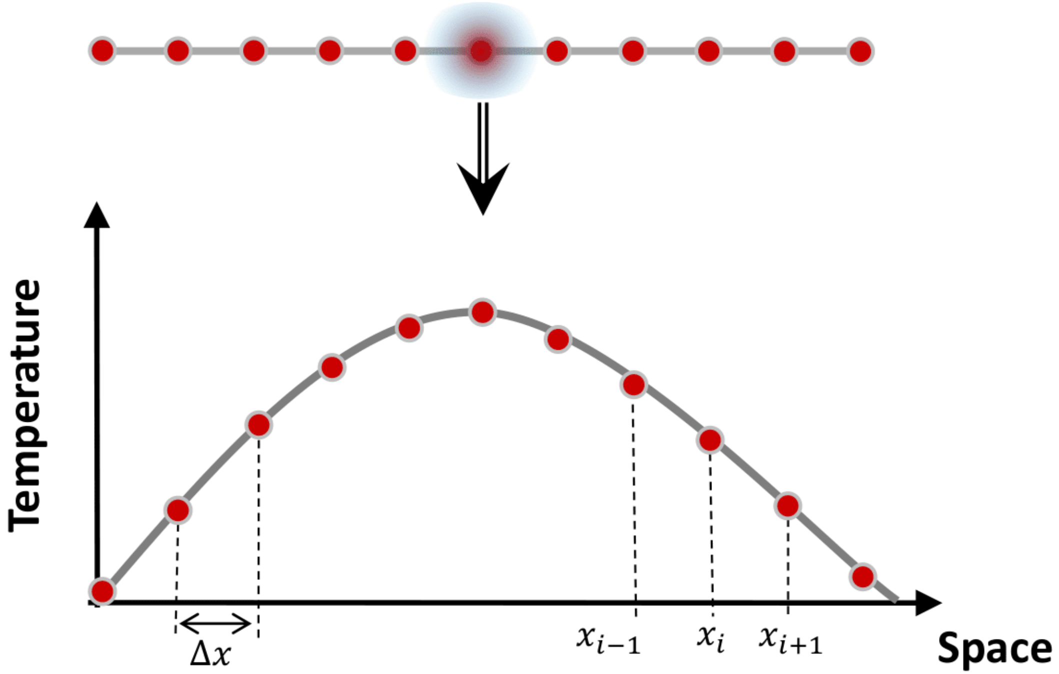
Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \frac{\partial^2 T}{\partial x^2}$



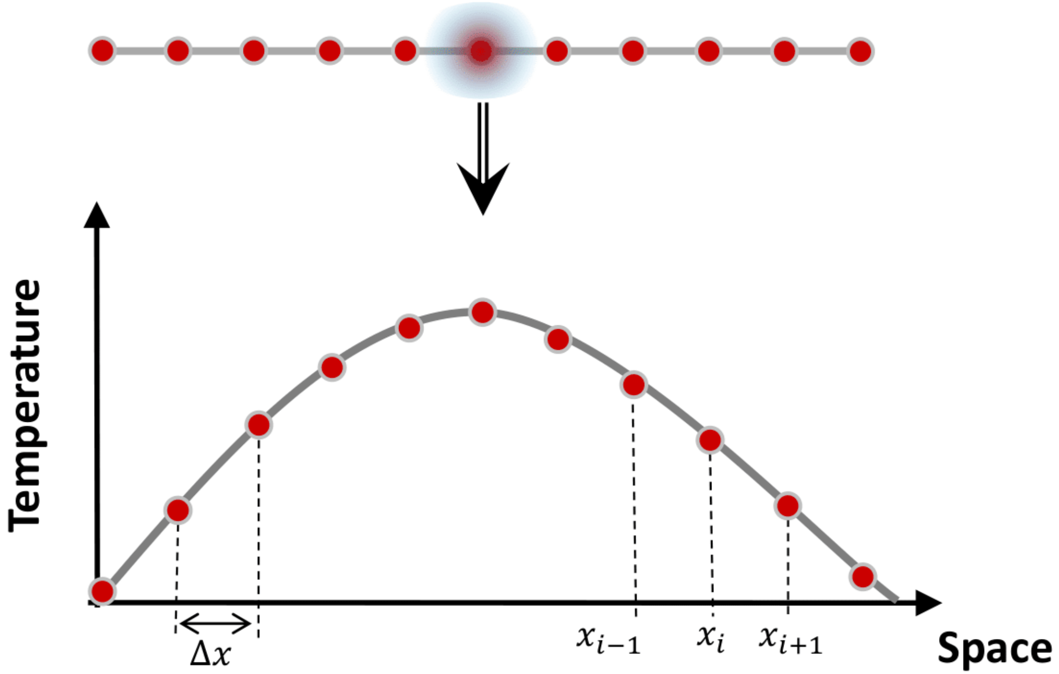
Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \frac{\partial^2 T}{\partial x^2}$



Finite Difference Definition

➢ Let's model the spatial derivative $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2}$

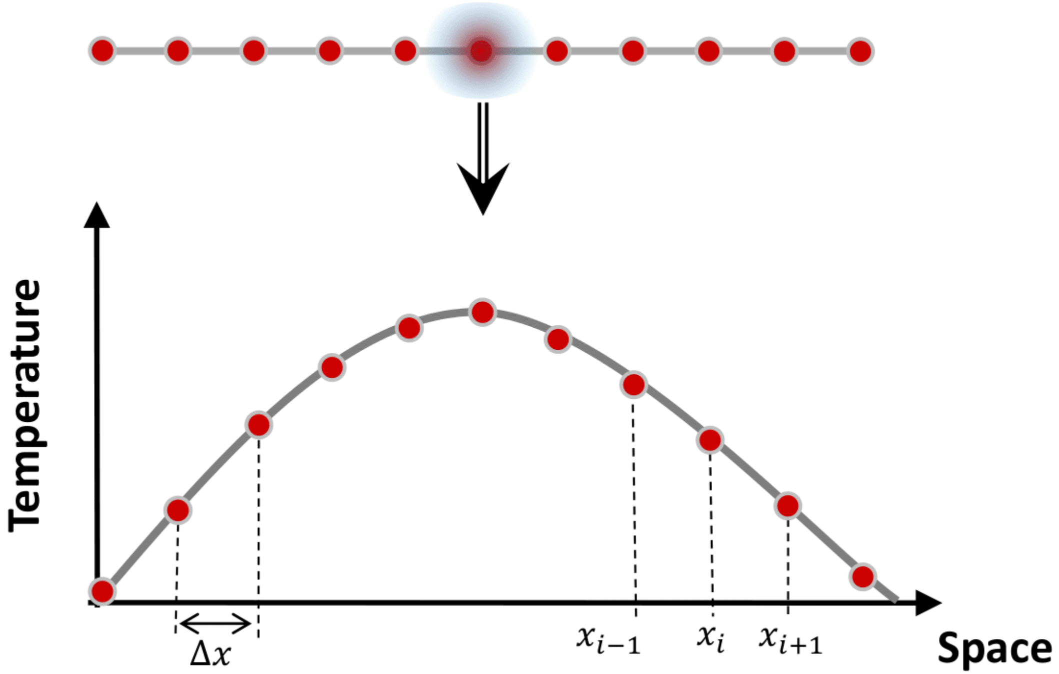


❶ $\dfrac{\partial T}{\partial x} =$

❷ $\dfrac{\partial}{\partial x}\left(\dfrac{\partial T}{\partial x}\right) =$

Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2}$



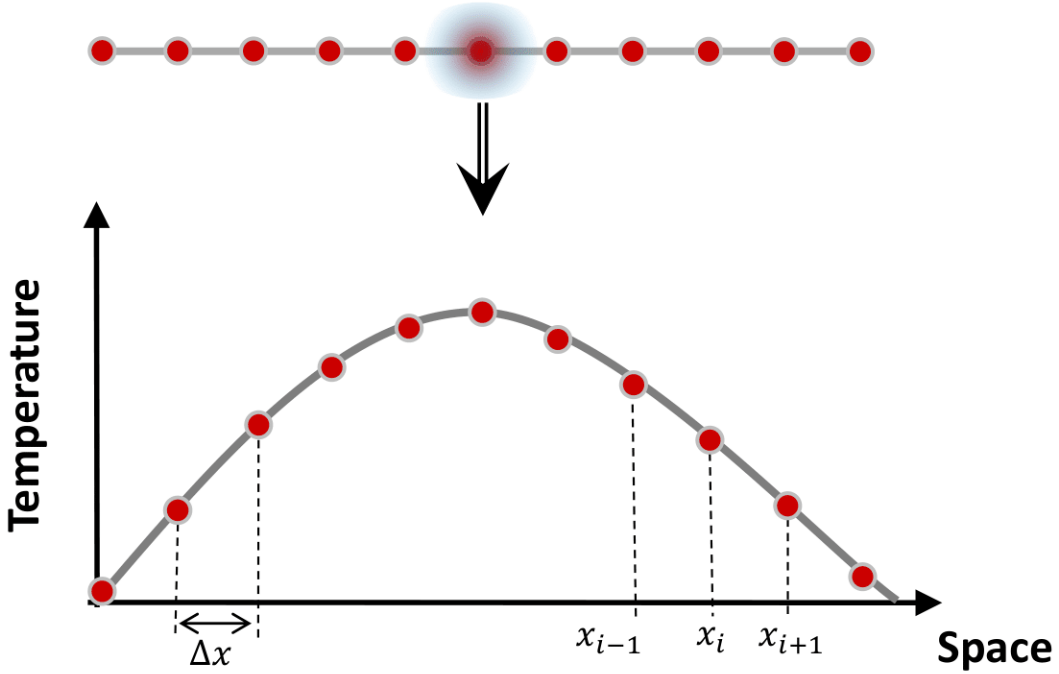
❶ $$\frac{\partial T}{\partial x} \approx \frac{T_{i+1}^{n} - T_{i-1}^{n}}{2\Delta x}$$

Centered difference

❷ $$\frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) \approx$$

Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2}$

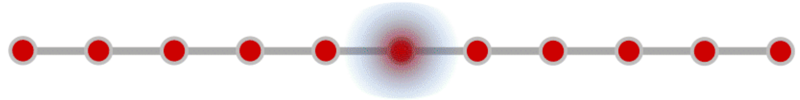


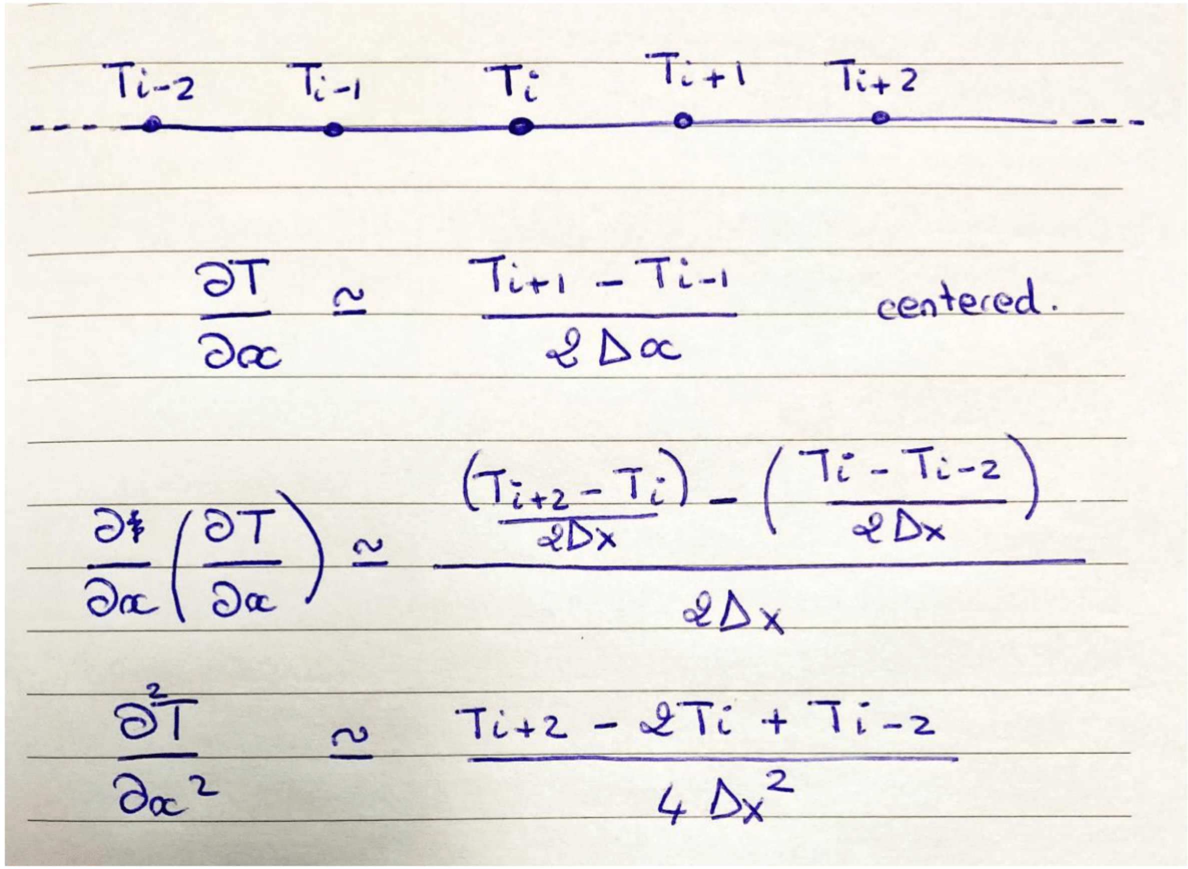
❶ $$\frac{\partial T}{\partial x} = \frac{T^n_{i+1/2} - T^n_{i-1/2}}{\Delta t}$$ **Centered difference**

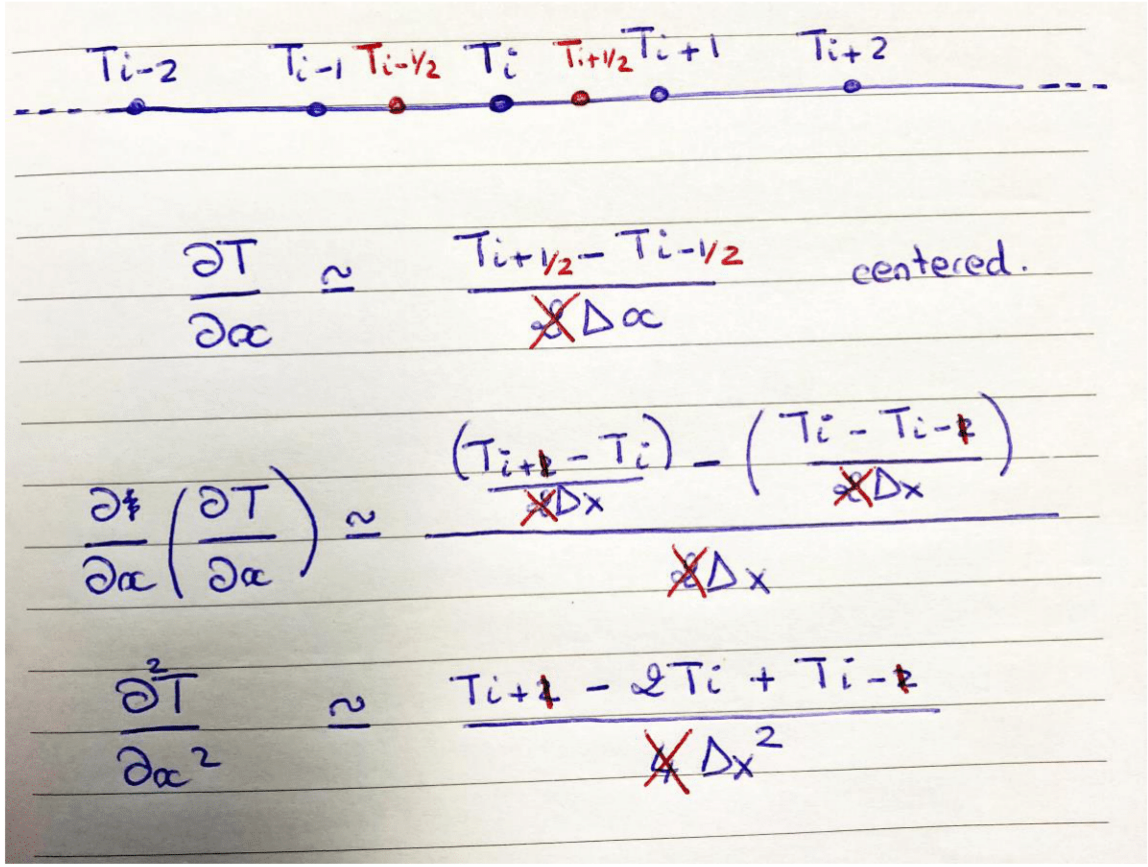
❷ $$\frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) = \frac{\partial^2 T}{\partial x^2} \approx \frac{T^n_{i+1} - 2T^n_i + T^n_{i-1}}{\Delta x^2}$$

Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2}$

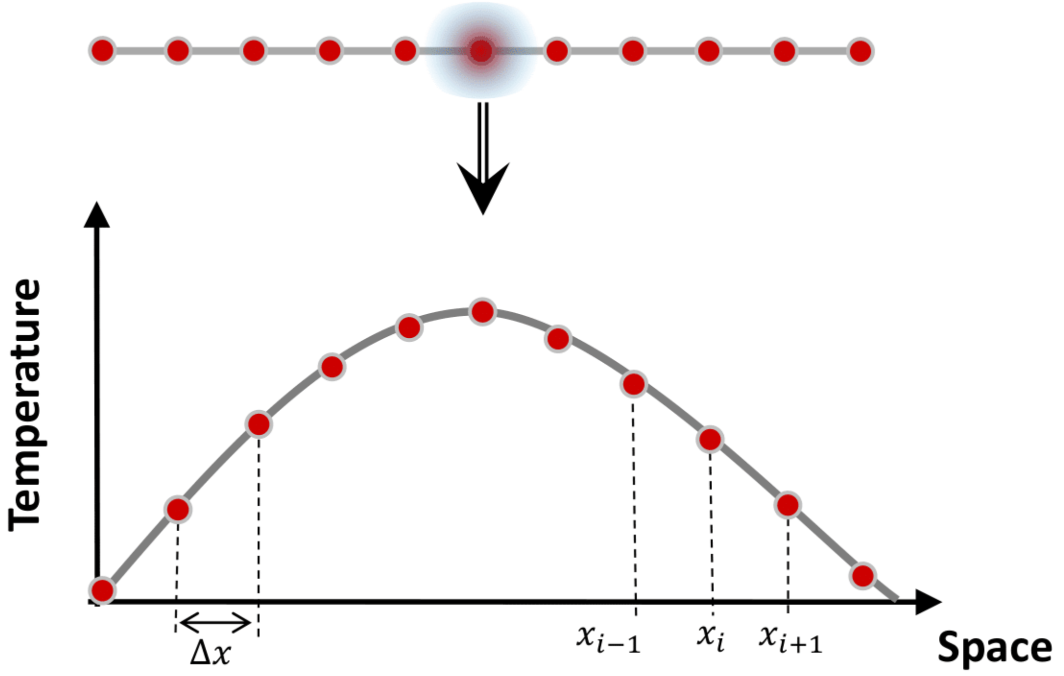






Finite Difference Definition

- Let's model the spatial derivative $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2}$



$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

Centered scheme

Solving the 1D Diffusion Equation

- Back to the diffusion equation

$$\boxed{\frac{\partial T}{\partial t}} = K_{Th} \boxed{\frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right)}$$

Solving the 1D Diffusion Equation

- Back to the diffusion equation

$$\frac{\partial T}{\partial t} = K_{Th} \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right)$$

becomes..

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = K_{Tv} \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

Forward Euler scheme

Solving the 1D Diffusion Equation

- Back to the diffusion equation

$$\frac{\partial T}{\partial t} = K_{Th} \frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right)$$

becomes..

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = K_{Th} \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

Forward Euler scheme

$$T_i^{n+1} = T_i^n + K_{Th} \frac{\Delta t}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

Explicit method : Direct calculation of the temperature at a later time from the current time

Solving the 1D Diffusion Equation

- We have an explicit formulation:

  - $$T_i^{n+1} = T_i^n + K_{Th}\frac{\Delta t}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

  - The initial condition: T at $t = 0$ for all x

The Computational Algorithm

➢ We have an explicit formulation :

- $$T_i^{n+1} = T_i^n + K_{Th}\frac{\Delta t}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

- The initial condition: T at $t = 0$ for all x

→ The computational algorithm consists of the following steps:

❶ Initialise the vector state variable ($T_i^1, i = 1, \dots, N_x$),

❷ Plot T^1 at every point in the domain,

❸ Apply equation (2) for all the internal points, $i = 2, \dots, N_x - 1$,

❹ Set the boundary values for $i = 1$ and $i = N_x$ (T_1^{n+1} and $T_{N_x}^{n+1}$),

❺ Plot T^2 at every point in the domain,

for $n = 1, \dots, N_t$

≺ Rince and repeat (steps ❸ ❹ ❺).

STEP 1: Logging onto the HPC cluster

- From a terminal/konsole:

```
ssh -X login@scp.chpc.ac.za
```

- Request one node with the alias command **qsubi1**

```
qsubi1
```

- Go your lustre directory and create a dedicated directory

```
cd lustre; mkdir diff; cd diff
```

- Copy a template of the computational algorithm and start **MATLAB**

```
cp /home/apps/chpc/earth/CROCCO_Workshop/
CROCO_TRAINING_Basic/3_Some_files/My_diffusion_1D.m .
```

STEP 2: Complete the Initialisation step

→ You have to complete this script at lines 29, 53 and 56 (see ➤➤➤ below). Here are the different parts of the algorithm:

➥ Line 14: K is the constant horizontal diffusion coefficient.

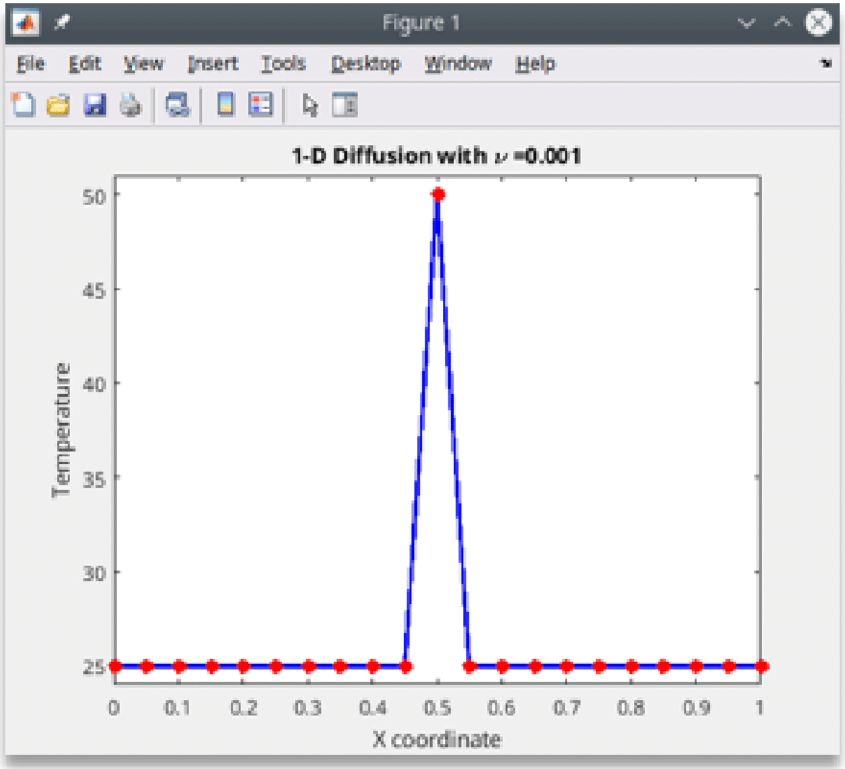
➥ Lines 16-18: the length of the swimming pool is descetized into 21 equaly-spaced points.

➥ Lines 20-21: We begin at $n = 1$, with $\Delta t = 0.1s$.

➥ Line 25: The swimming temperature pool is initialized at 25°C, everywhere. This corresponds to step ❶ of the computationsal algorithm (see #**3**).

➤➤➤ Complete line 29, to initialise (❶) the 11th spatial point at **50°C**.

➥ Lines 32-38: This is a plot (❷) of the temperature in the swimming pool.



STEP 3: Complete the Script

$$T_i^{n+1} = T_i^n + K_{Th}\frac{\Delta t}{\Delta x^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

➤➤➤ At line 53, start by applying equation (2) for the 11th spatial point (❸), such that: `temperature_new(11)= temperature(11)+...`

↳ You can define a coefficient `alpha`, such that $\alpha = K\Delta t/\Delta x^2$

➤➤➤ At line 53, apply equation (2) for all the internal points, $i = 2, \ldots, N_x - 1$ (❸) using a **for loop** (`for i=2,nx-1`)

➤➤➤ At line 56, apply the boundary conditions at $i = 1$ and $i = N_x$ (❹). Either the temperature at these points remains at 25°C, or you can copy the temperature of the closest internal point.

➤➤➤ When it is working, increase the number of time steps nt (at line 20)

➤➤➤ Decrease and the increase the time step dt (line 21). What do you observe?

STEP 4: Exiting

- Exit Matlab:

```
exit
```

- Give back the compute node:

```
exit
```

- Logoff the Lengau cluster:

```
exit
```