Peter Stanley Jørgensen

Simulation of physics

Today

Simulation of physics from the big picture to a code implementation

- Introduction to materials modeling
- The heat equation
- The finite difference method
- Overview of a Matlab solver for the heat equation

- Exercise – Explore and extend the Matlab solver for the heat equation
  – Learning how to do things in Matlab is a part of the exercise

- ~11:00 I will go through a few previous exam questions for the content in this module.
- ~11:45 Recap of exercise and solution

Learning objectives and exam

Course learning objectives for this module

- Simulate materials behavior related to their internal structure

Exam questions

- On the content of this presentation
- On the exercise
  – A solution will be uploaded after the module

- You will not be asked to do programming at the exam.

Introduction to materials simulation

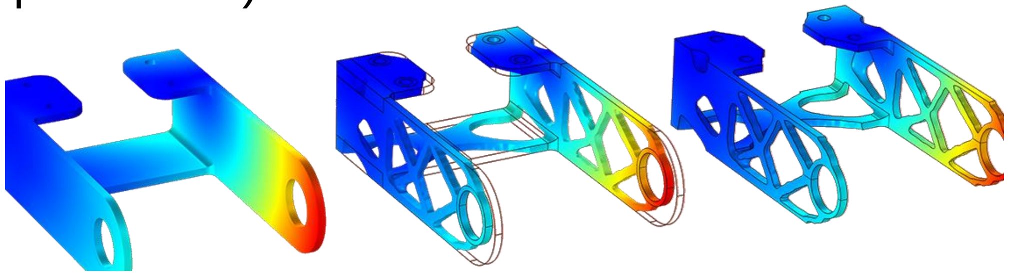
A crash test

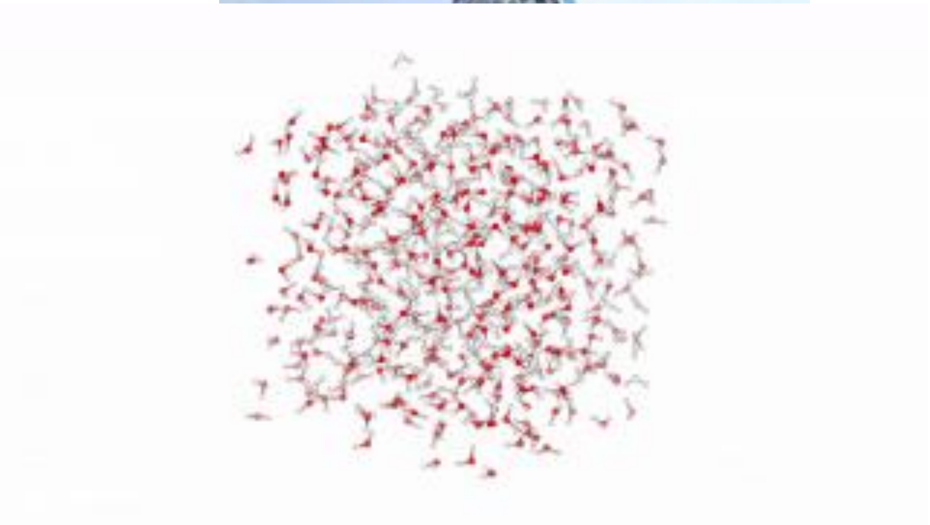
https://www.youtube.com/watch?v=A2Av9nygEoI

Why model and simulate?

- Avoid trial and error
  - Creating the object could be expensive.
  - Predict failure in rare events
    - Earthquake
    - Typhoone

- Optimize the design parameters
  - How little material can I use and still meet the safety standards for the building?
  - How should this material be distributed? (Topology optimization)



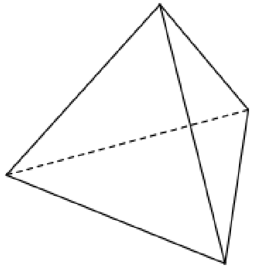


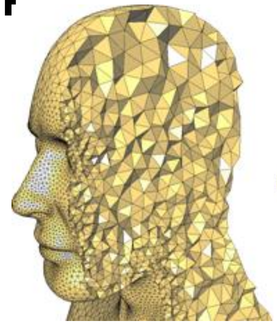
https://en.wikipedia.org/wiki/Molecular_dynamics

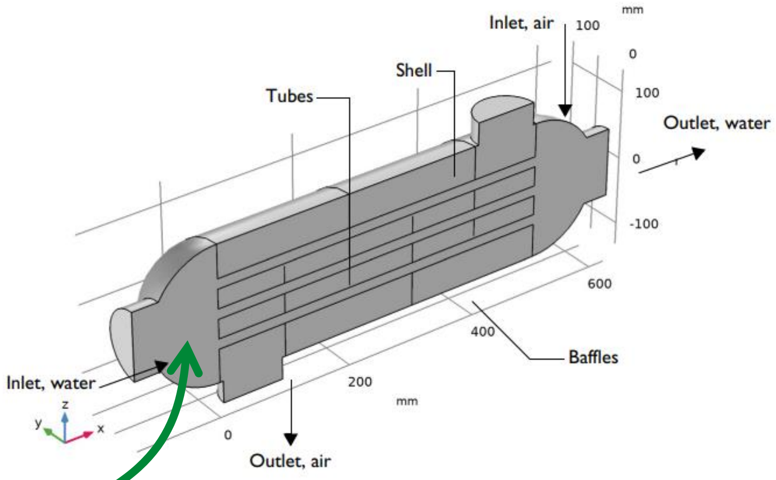
Molecular dynamics simulation of water molecules on a picosecond timescale

Components of a simulation?

- Geometry
  - A tetrahedral mesh
  - A grid
- Material properties
  - Thermal conductivity, density…
- Physics models
  - Mathematics, partial differential equations
- Boundary conditions
  - Limit ourselves to a certain domain.
  - What should happen at material interfaces?
- A solver
  - How to solve the equations on the model.

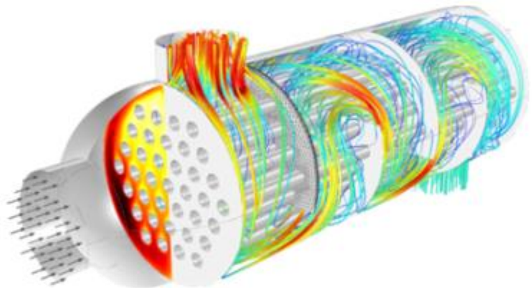






$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

$$\frac{\partial u}{\partial n} = f$$



The heat equation

The heat equation

- Descripes how the distribution of heat changes in a solid medium with time

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

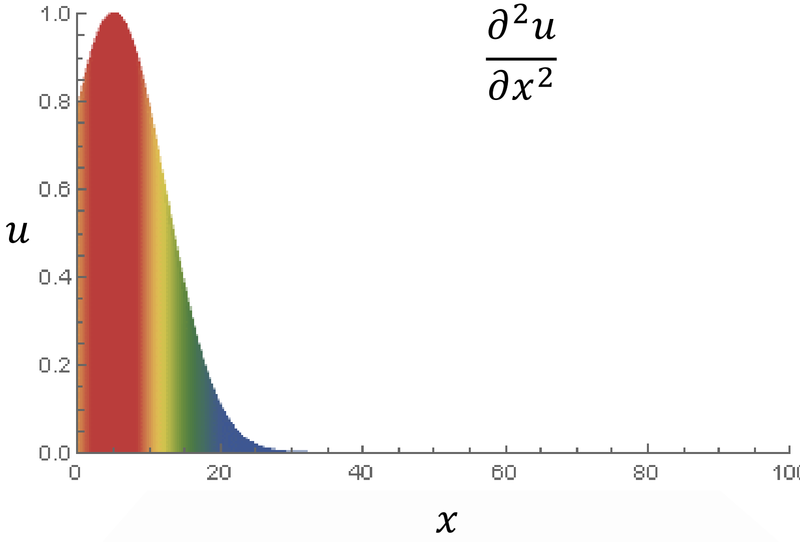
  – u temperature
  – t time
  – x, y, z spatial coordinates
- $\frac{\partial u}{\partial t}$ is the change in temperature with time.
- α is a material parameter, the thermal diffusivity, $\alpha = \frac{k}{c\rho}$
  – k thermal conductivity
  – c specific heat capacity
  – ρ density
- Very similar equations govern many other different types of physics. E.g. Electrical currents, diffusion, pressure etc.

The heat equation - intuition

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

- Whenever the second derivative is non-zero the temperature changes ($\frac{\partial u}{\partial t}$ is non-zero). i.e. where the curvature of the temperature with space is large.
- In a steady state solution (no change with time) the right hand side is zero everywhere.

- Intuitively:
  – If the average temperature in the surroundings is higher
    - Heat up
  – If the average temperature is lower
    - Cool down



1D example

The Laplacian

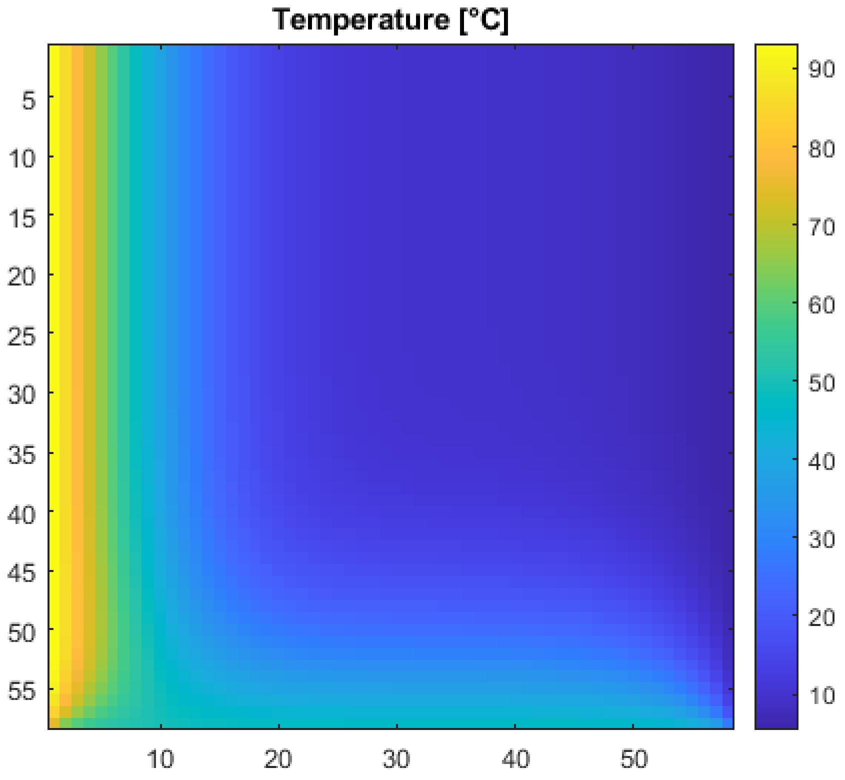
- $\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) = \nabla^2 u$
- ∇^2 is the Laplacian differential operator, convenient for compact notation.
- Can be seen as a generalization of the second derivative to n dimensions.

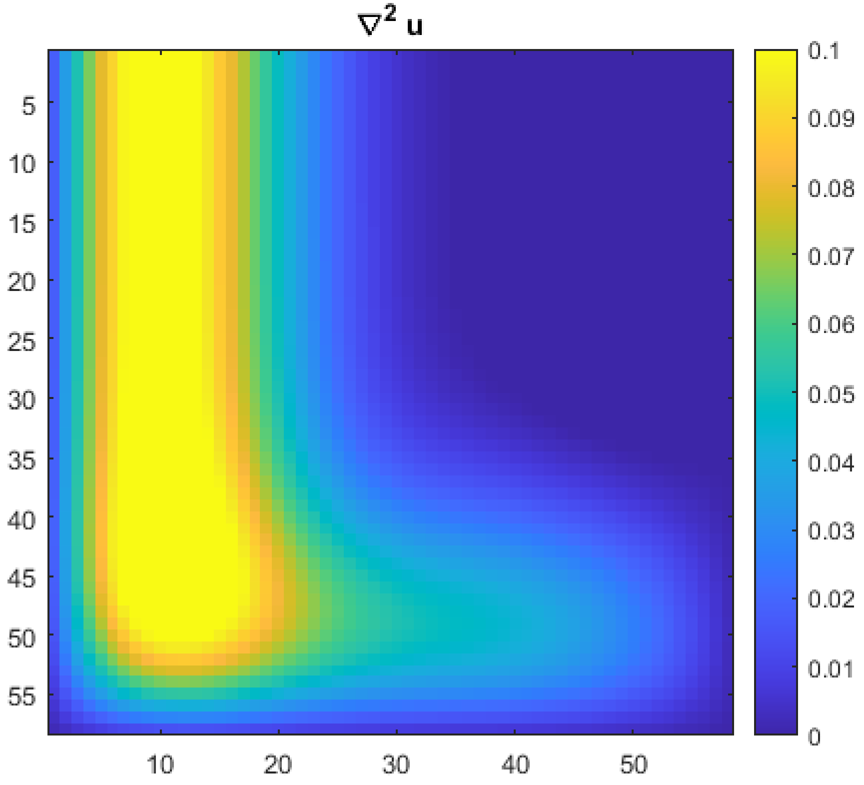
2D example

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \alpha\nabla^2 u$$

Is this temperature distribution a steady state solution to the heat equation?

NO! The laplacian is not zero everywhere. According to the heat equation the temperature will change



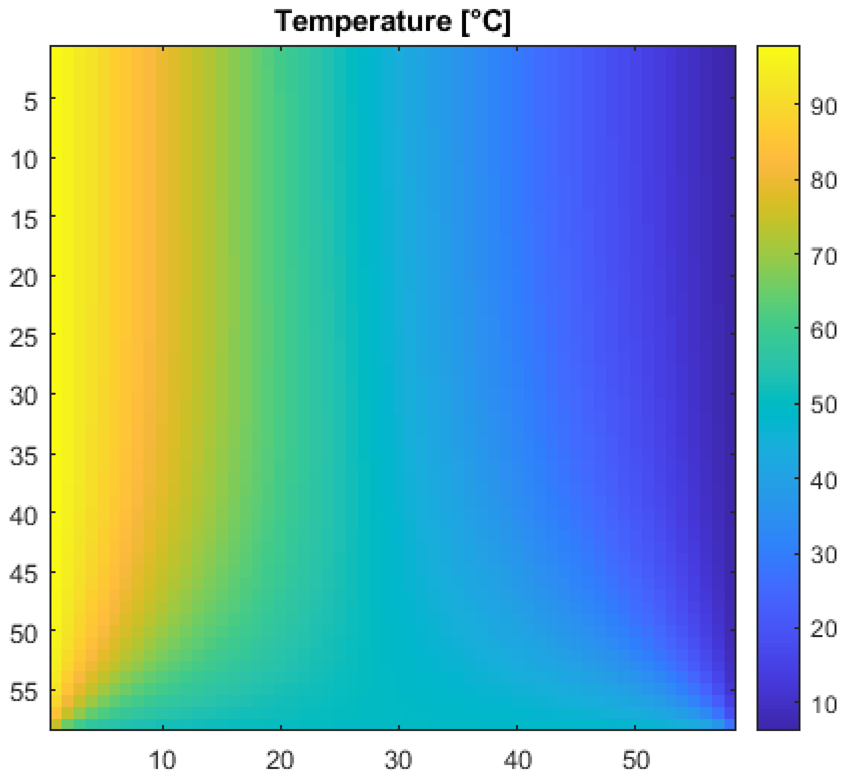


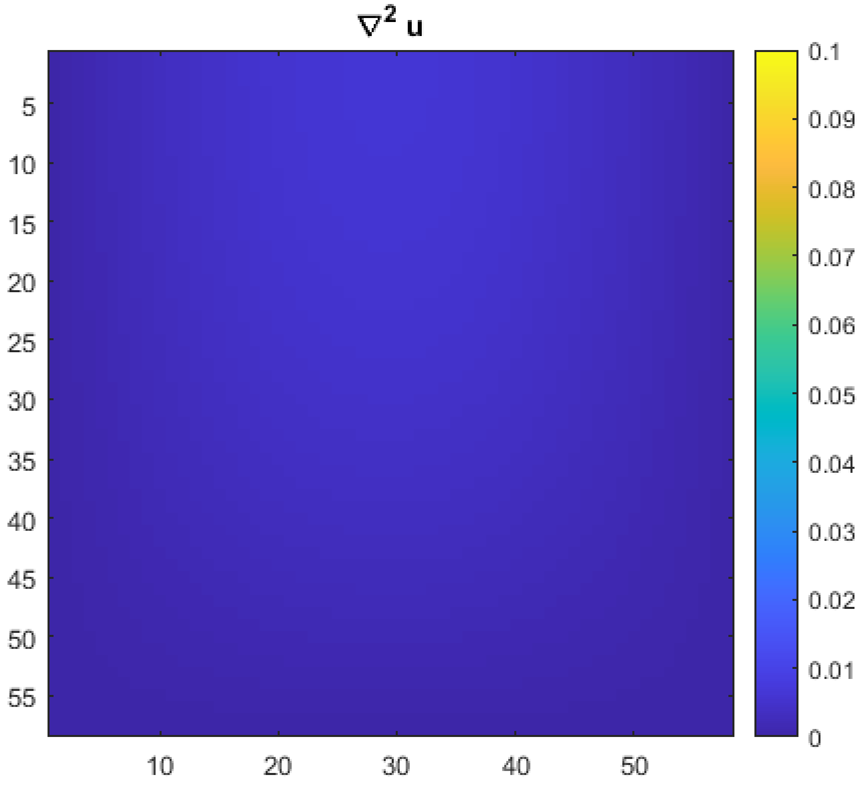
2D example

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \alpha\nabla^2 u$$

Is this temperature distribution a steady state solution to the heat equation?

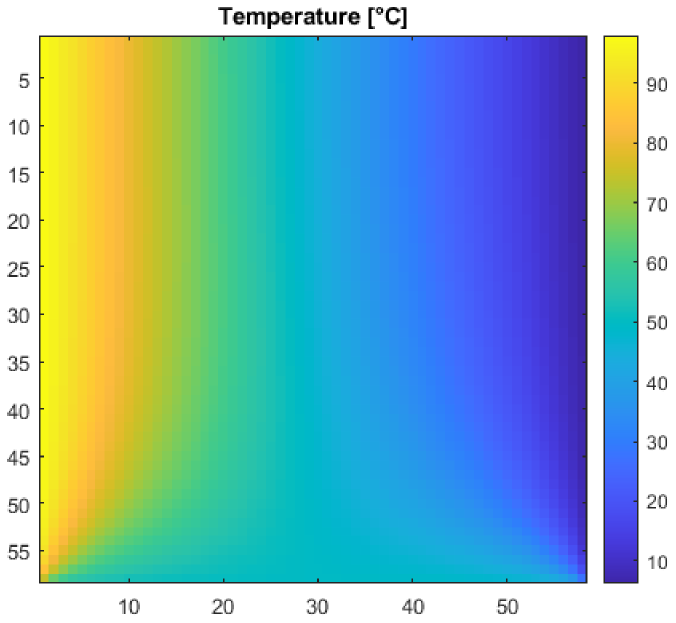
YES! The laplacian is (very close to) zero everywhere.





Which solution do we want?

- Infinitely many solutions, we need some constraints.
- What is the initial state of the system?
- Bondary conditions
  - What should happen at the edges of the simulation domain?
- Dirichlet boundary conditions, $u = f$
  - Set a boundary to having a constant value (temperature).
- Neumann boundary conditions, $\frac{\partial u}{\partial n} = f$
  - Set a boundary to having a specific (temperature) gradient.
  - The n means that it's the gradient normal to the boundary.

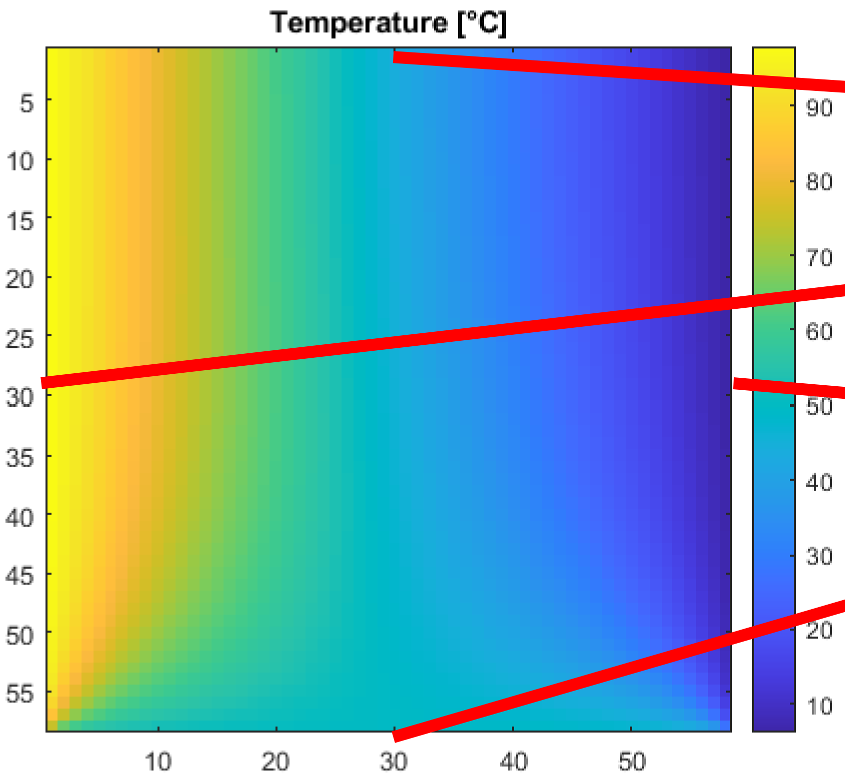


Modeling the real world

- The mathematical boundary conditions have direct real world interpretations.
- Dirichlet boundary conditions, $u = f$
  - Model a part of the boundary as having a set temperature that never changes.
  - The system is in contact with something that can be considered having a constant temperature.
- Neumann boundary conditions, $\frac{\partial u}{\partial n} = f$
  - $\frac{\partial u}{\partial n} = 0$
  - Ideal insulating surface, no heat can be transferred.
  - $\frac{\partial u}{\partial n} > 0$
  - Introduce heat into the system.
  - $\frac{\partial u}{\partial n} < 0$
  - Remove heat from the system

Guess a boundary condition…

- What boundary conditions do you think were used on each side of the system?



$$\frac{\partial u}{\partial n} = 0$$

$$u = 100\,°C$$

$$u = 5\,°C$$

$$u = 50\,°C$$

The initial temperature was set to 10 ° C everywhere. However, since this is the steady state solution the initial temperature is inconsequential.

Break!

Install Matlab if you haven’t already – instructions on DTU Learn

The finite difference method

How do we solve this?

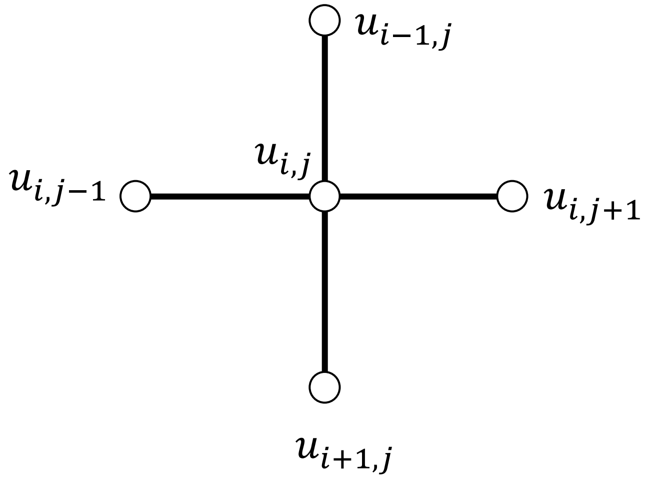
$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

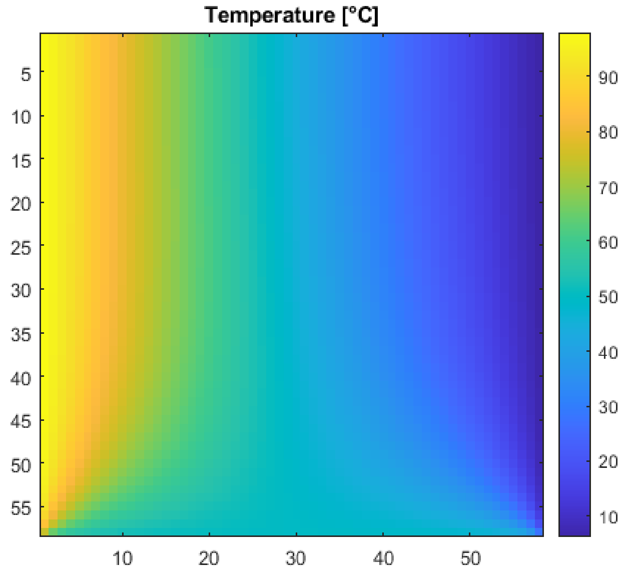
- Here we will use an iterative approach since it is the simplest (other methods exist).

- We want to find an update formula for what the temperature should be in our system at each iteration (at each step in time)
- This update formula should only contain expressions that a computer can evaluate. E.g. no continuous math (derivatives, integrals)
- Discretization…

Nomenclature for working on a grid with discrete time steps

- **Spatial discretization**
  - $u_{i,j}$ is the temperature at grid point (i,j)
  - Neigboring gridpoints are defined relative to the $u_{i,j}$ grid point
  - Think of i,j as the row and column number of a matrix.
- **Temporal discretization**
  - u^n is the temperature at the current time.
  - u^{n+1} is the temperature at the next time step.
- **Spatial and temporal discretization**
  - $u^n_{i,j}$ is the temperature at the current time at grid point i,j.





Finite differences

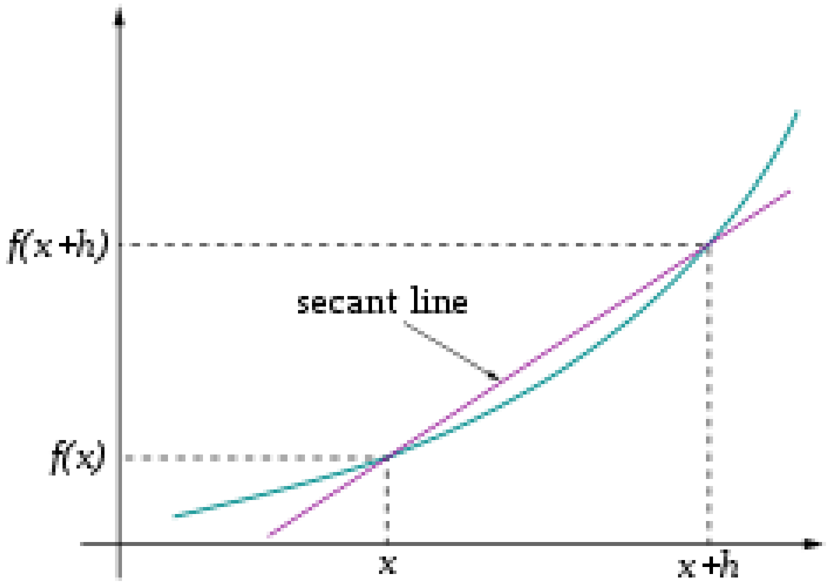
$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

- Forward differences in time:

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t}$$

- Central differences in space:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1,j}^{n} - u_{i,j}^{n}}{\Delta x}$$



$$\frac{\partial^2 u}{\partial x^2} \approx \frac{\frac{u_{i+1,j}^{n} - u_{i,j}^{n}}{\Delta x} - \frac{u_{i,j}^{n} - u_{i-1,j}^{n}}{\Delta x}}{\Delta x} = \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2}$$

It is just comparing the temperature to its surroundings

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}$$

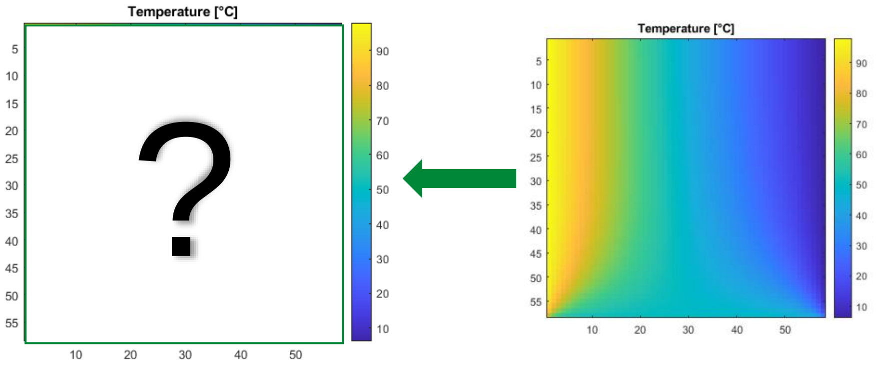
Putting it together

$$\frac{\partial u}{\partial t} = \alpha(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \alpha(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2})$$

$u_{i,j}^{n+1}$ is the only unknown.

$$\boxed{u_{i,j}^{n+1} = u_{i,j}^{n} + \alpha\Delta t(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2})}$$



You will use this in the exercise

A toy Matlab implementation

Introduction to the implementation

- You will work with a barebones implementation of a 2D heat equation solver in Matlab that uses the update formula arrived at above.

- $u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}\right)$

Matlab – the loop in pseudo code

```
Initialization - set the initial temperature
```
$u_{i,j}^{n}$

```
Loop
```

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}\right)$$

```
    Enforce boundary conditions
```

$$u_{i,j}^{n} = u_{i,j}^{n+1}$$

```
end
```

Matlab - initialization

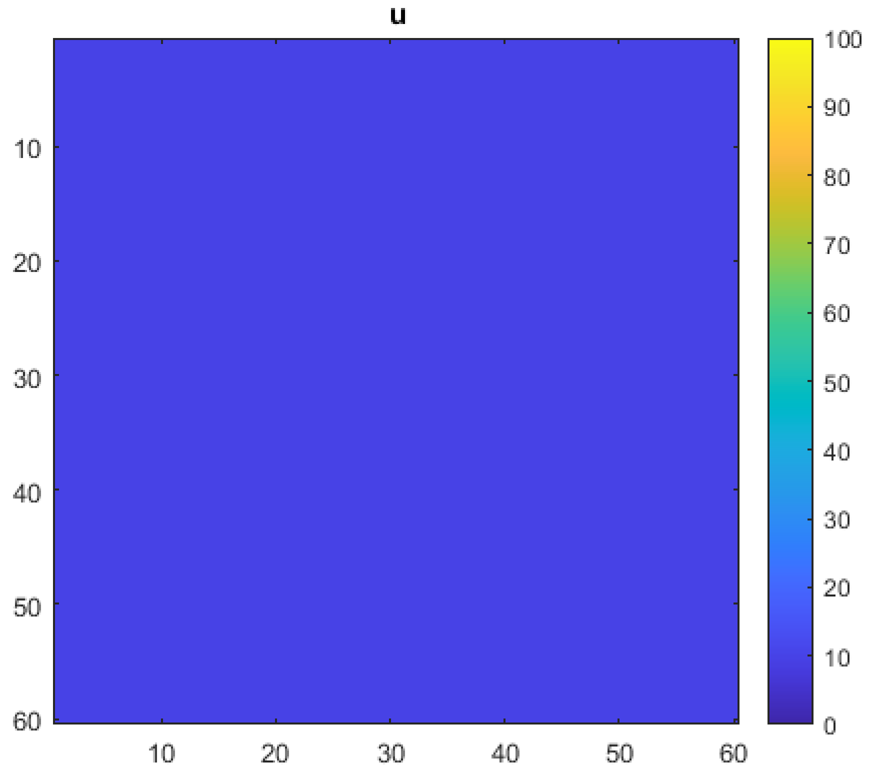
The temperature u is a matrix.

$u_{i,j}^{n}$ is quivalent to

```
u_n(i,j)
```

To initilialize the temperature to be 10 °C everywhere.

```
u_n = ones(60,60)*10;
```



Boundary conditions

- Dirichlet boundary conditions, $u = f$

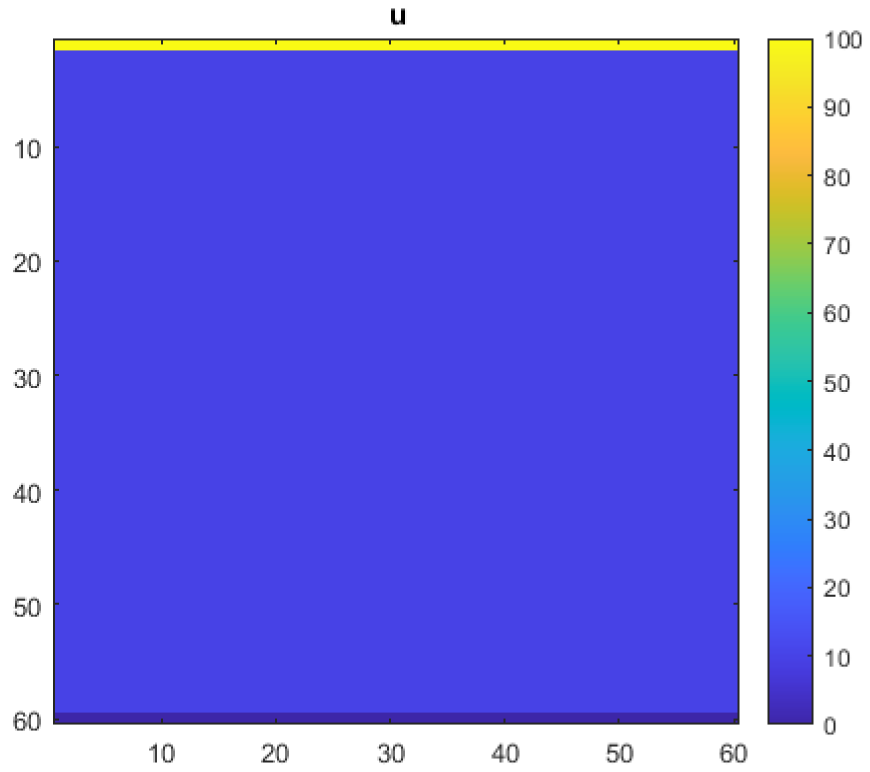
```
u_n(1,:)=100;
u_n(end,:)=0;
```

- Neumann boundary conditions, $\frac{\partial u}{\partial n} = f$

Here $\frac{\partial u}{\partial n} = 0$

```
u_n(:,1)=u_n(:,2);
u_n(:,end)=u_n(:,end-1);
```

- That's it! The update formula does the rest



Matlab – the loop in pseudo code

```
Initialization - set the initial temperature

Loop
```

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}\right)$$

```
        Enforce boundary conditions
```

$$u_{i,j}^{n} = u_{i,j}^{n+1}$$

```
end
```

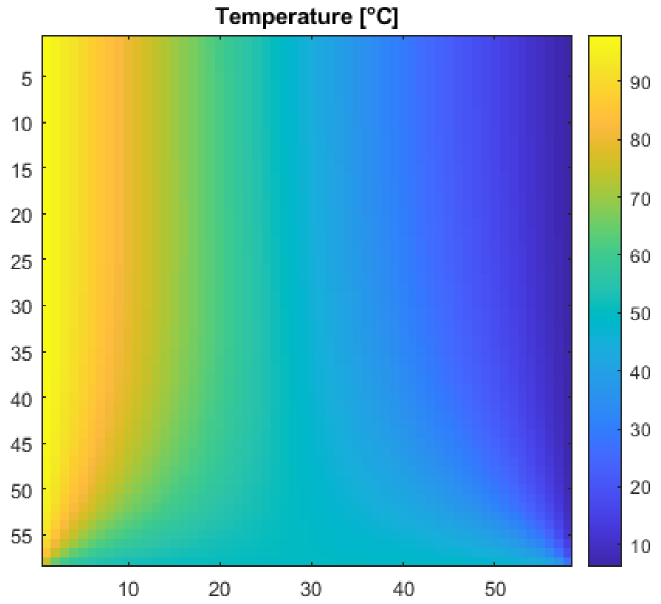
When do we stop the loop?

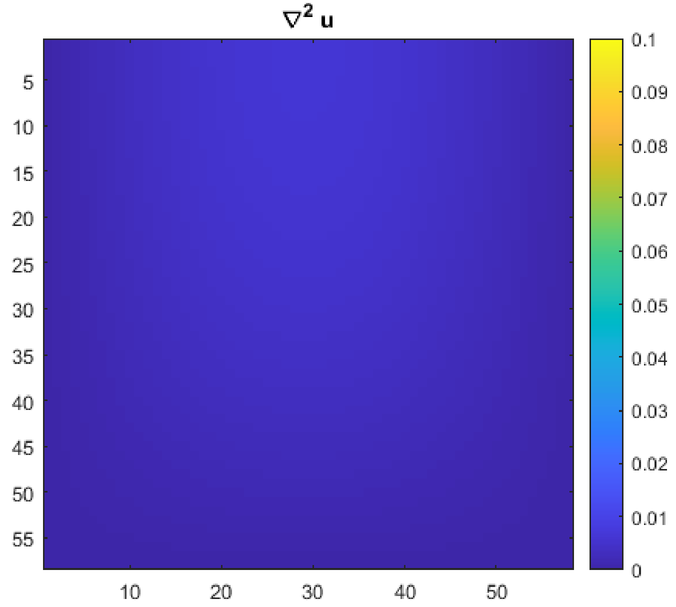
Implement the update formula and see it in action in the exercise.

Stopping criteria for a steady state solution

Strategies?

- Run plenty of iterations
- When the change in temperature is below a threshold at a specific location
- When the relative change in temperature is below a threshold at a specific location
- When the relative change in temperate is low everywhere





Back to reality

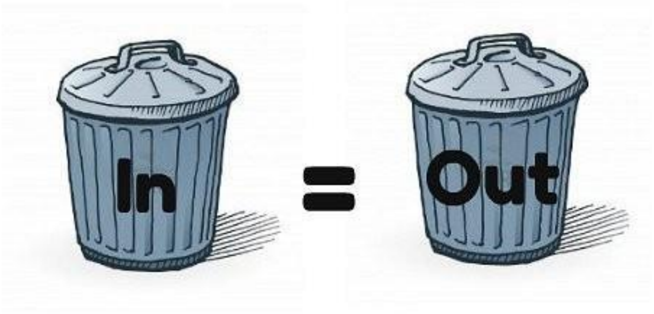
Non-toy simulations

- It takes a lot of resources to develop robust simulation packages.
  – The rabbit hole goes much deeper than we have explored here.

- Use a mature package instead of implementing your own methods
  – Unless you are specifically improving or developing a new simulation approach.

- Commercial software can often be used without knowing how the solver works.
  – However, Garbage in = Garbage out

COMSOL Ansys



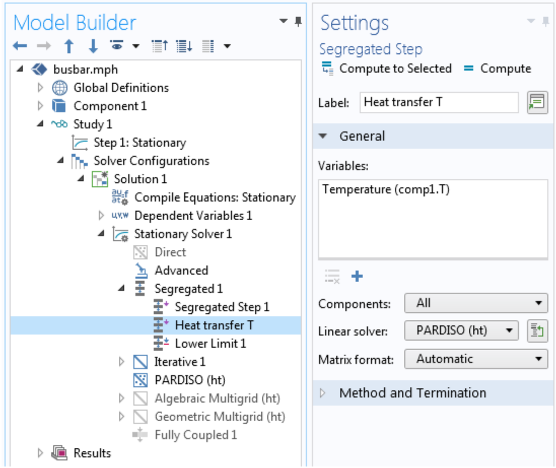
To be a succesful modeler

- Knowledge about the materials you are making simulations of.
  – Do you know their properties?
  – Which simplification or assumptions are OK to make?

- Knowledge of the equations that model the physics you are interested in.
  – Are the equations valid for what you are simulating? (e.g. length scales)

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

- Knowledge of the solver you are using to solve those equations.
  – Complex models can be unsolvable using standard solvers.
  – Which solver should you use and how should you tweak its settings?



Matlab quick tips

- Run a highlighted/selected chunk of code: F9 (in windows)
- Print a variable to the command window: double click it to highlight and press F9
- Run the entire script: F5
- Stop printing to command window: Ctrl-c (also interrupts a program that is running)
- See your variables: whos
- Get help on a function: help function_name

Exercise

- Open ”Exercise - Simulation.pdf” for instructions.

- Visual tutorial for setting up Matlab and running the script…

AI can be a great help in understanding code syntax and errors: [Copilot (microsoft.com)]

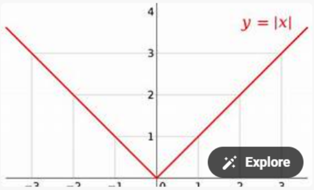
You

how do i find the largest absolute value of a matrix in Matlab?

Copilot

To find the largest absolute value of a matrix in **MATLAB**, you can follow these steps:

1. First, compute the absolute value of each element in the matrix using the `abs()` function. This will give you a matrix with the absolute values of the original elements.
2. Next, find the maximum value in this new matrix. You can achieve this using the `max()` function.



Here's an example of how to do it:

```
% Create a sample matrix (you can replace this with your own matrix)
A = [1.5 -3.2 4.7; -2.1 0.8 -5.6; 3.9 2.4 -1.0];

% Compute the absolute value of each element
abs_A = abs(A);

% Find the maximum absolute value
max_abs_value = max(abs_A(:));

disp(['The largest absolute value in the matrix is: ', num2str(max_abs_value)]);
```

In this example, `abs_A(:)` converts the matrix `abs_A` into a column vector, and `max()` finds the maximum value in that vector. The result will be the largest absolute value in the original matrix.

Remember to replace the sample matrix `A` with your actual matrix data. If you have any further questions or need additional assistance, feel free to ask! 😊

Learn more 1 mathworks.com 2 stackoverflow.com 3 mathworks.com +3 more

1 of 30 responses

As always, there is no guarantee that the AI answer is correct

11:00
Examples of exam questions

Boundary conditions 1

Consider a modeling the temperature distribution in a rod of metal as a 1D heat transfer problem using the heat equation. At one end of the metal rod the boundary condition $\frac{\partial u}{\partial n} = 0$ is used. What is the type and the physical interpretation of this boundary condition.

1. A Neumann boundary condition describing an insulating boundary.
2. A Neumann boundary condition describing a boundary with a set temperature of 0 °C.
3. A Dirichlet boundary condition describing an insulating boundary.
4. A Dirichlet boundary condition describing a boundary with a set temperature of 0 °C.
5. A Laplacian boundary condition describing an insulating boundary
6. A Laplacian boundary condition describing a boundary with a set temperature of 0 °C.

Q1 Boundary conditions

Consider simulating the temperature distribution in a rod of metal as a 1D heat transfer problem using the heat equation. The rod is 1m long. The two ends of the rod are defined at x=0 m (point A) and x=1 m (point B). In the steady state solution, the temperature in the center of the rod is 30 °C. Which combination of initial temperature (in the entire rod), boundary condition at point A and boundary condition at point B could **not** have produced this result?

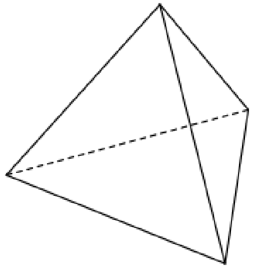
1. Initial temperature 20° C, point A: u=30° C, point B: $\frac{\partial u}{\partial n} = 0$
2. Initial temperature 20° C, point A: $\frac{\partial u}{\partial n} = 0$, point B: $\frac{\partial u}{\partial n} = 0$
3. Initial temperature 30° C, point A: u=20° C, point B: u=40° C
4. Initial temperature 20° C, point A: u=20° C, point B: u=40° C

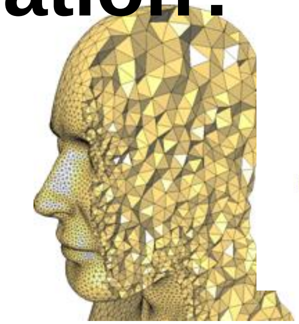
Strategy: Make a quick sketch of each of the setups

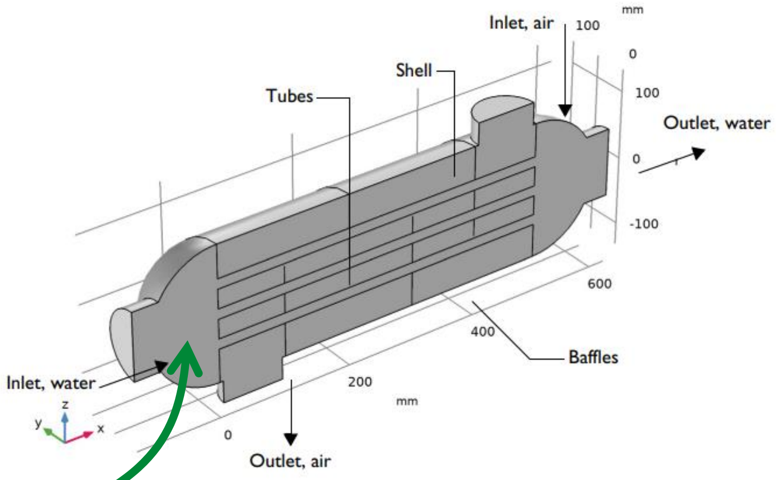
11:45
I will recap and go through solutions to exercises

A solution pdf and code files are available on Learn

What we did today!
The components of a simulation?







- Geometry
  - A tetrahedral mesh
  - A grid
- Material properties
  - Thermal conductivity, density…
- Physics models
  - Mathematics, partial differential equations
- Boundary conditions
  - Limit ourselves to a certain domain.
  - What should happen at material interfaces?
- A solver
  - How to solve the equations on the model
  - A simple working code implementation

$$\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$$

$$\frac{\partial u}{\partial n} = f$$