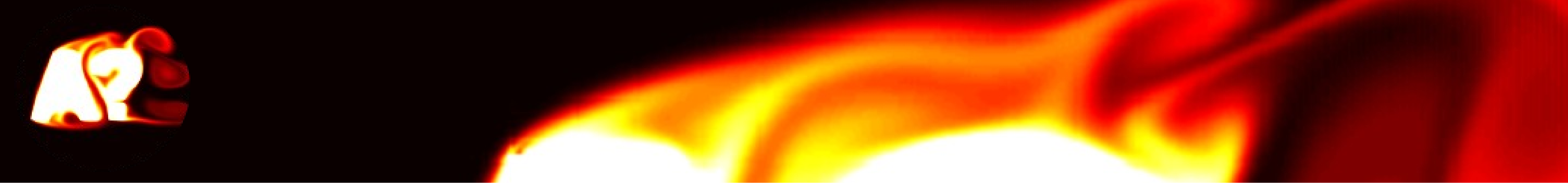
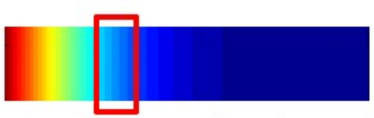


EEN-1020 Heat transfer

Week 1: Fourier’s law, heat equation, Newton’s law and numerical solution

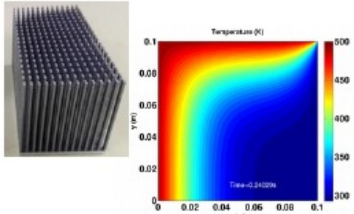
Prof. Ville Vuorinen
October 25th- 26th 2022
Aalto University, School of Engineering

Week 1: Energy conservation, heat equation, conduction Fourier/Newton

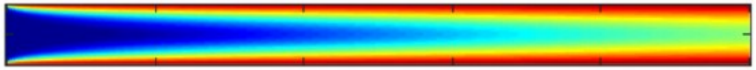


$$\frac{\partial T}{\partial t} \approx \frac{1}{\Delta x}\left(k\frac{\partial T(x+\Delta x/2,t)}{\partial x} - k\frac{\partial T(x-\Delta x/2,t)}{\partial x}\right)$$

Week 2: Fin theory, conduction, intro to convection

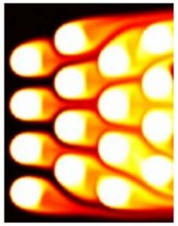


Week 3: convective heat transfer – internal flow (channel)

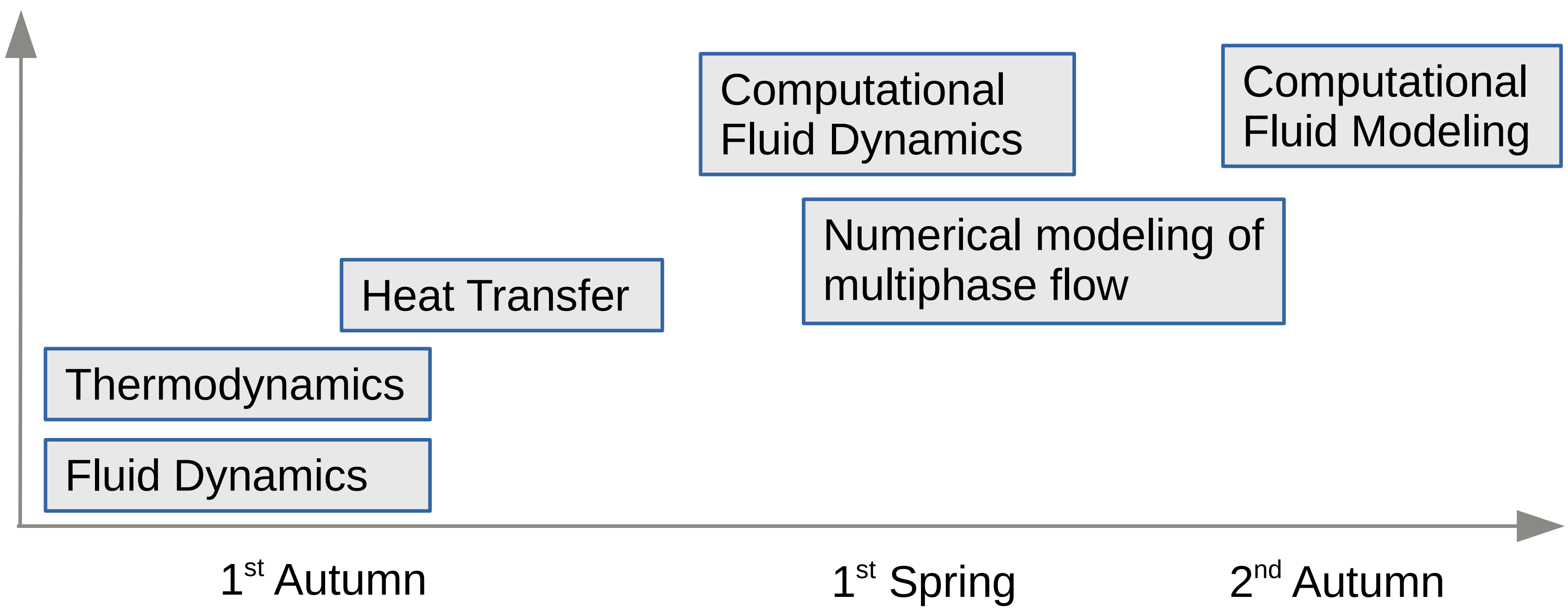


Week 4: convective heat transfer – external flow (fin systems)

Week 5: natural convection, boiling, correlations



Heat transfer in the AEE program e.g. computational methods study path in SEC major



On the heat transfer course, we have “5 friends” i.e. 5 main principles that are used to explain heat transfer phenomena

1) Energy conservation: “J/s thinking”
2) Fourier’s law
3) Newton’s cooling law
4) Energy transport equation – convection/diffusion equation
5) Momentum transport equation – Navier-Stokes equation

Lecture 1.1 Theory and analysis: Energy and mass conservation, Newton's cooling law, Fourier's law and conduction (1d heat equation)

ILO 1: Student can derive and explain physical origin of the heat equation, describe solution behavior by example solutions and boundary conditions, and solve the heat equation (1d) and Newton's cooling law (0d) numerically in Matlab.

Remarks on temperature, thermal energy and transport mechanisms

Temperature

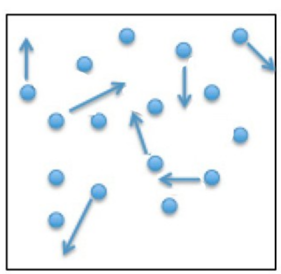
- **For gases or liquids:** temperature is actually closely related to the speed of the molecules on the molecular scales (molecules bouncing around). Molecular speeds are much higher (e.g. 2000-10000 m/s) than macroscopic fluid flow velocities (e.g. 0.1-10 m/s) in cooling/heating applications.
- **For solids:** temperature is related to the vibrational motion (velocity around an average position) of molecules/atoms in a lattice structure.

Energy

- Fluid=gas or liquid
- Fluids have kinetic energy and thermal energy. On the course we assume that

  kinetic energy does not change form and are typically only interested in thermal energy changes $dE=mc_p dT$
- **Main mechanisms of thermal energy transport:** convection, diffusion (conduction), radiation.

Propane Gas Tank



Molecules inside the gas tank

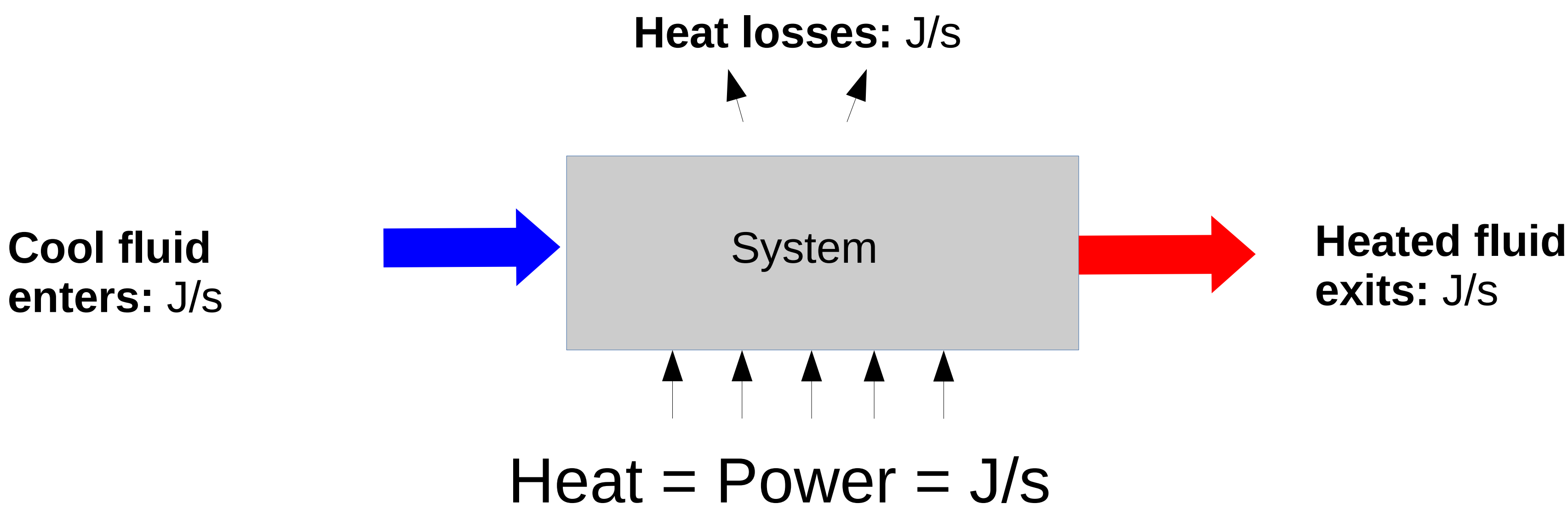
Mean squared molecule velocity relates to temperature.

$$\boxed{\frac{1}{2}m\overline{v^2} = \frac{3}{2}k_B T}$$

https://www.google.com/url?sa=i&url=https%3A%2F%2Fcommons.wikimedia.org%2Fwiki%2FFile%3AIdeal_Gas_Law.jpg&psig=AOvVaw05UbJ19Q3ydzSQ6WXX-1FJ&ust=1603788775676000&source=images&cd=vfe&ved=0CAkQjhxqFwoTCMia_vPw0ewCFQAAAAAdAAAAABAL

Energy conservation: “J/s” thinking

- Heat transfer course is largely involved with thermal energy balance considerations for a system.
- **[Energy] = J = kgm²/s²   [Power] = W = J/s**
- Typically we consider heating/cooling of fluid and/or solid
- Fluids = gas/liquid are assumed to be of constant density.



Example: Thermal energy and mass conservation for fluid flowing through heating system

(assume here: losses small)

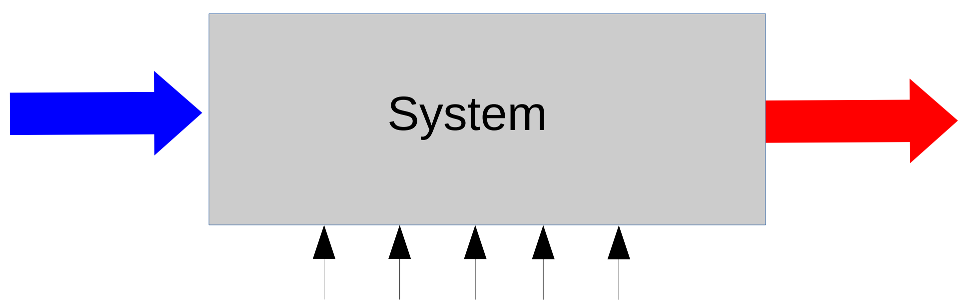
- **Mass conservation (kg/s):**

$$\boxed{\rho U_{\mathrm{in}} A_{\mathrm{in}} = \rho U_{out} A_{out} = \dot{m}}$$

- **Energy conservation (J/s):**

$$\boxed{c_p \rho U_{out} A_{out} T_{out} - c_p \rho U_{\mathrm{in}} A_{\mathrm{in}} T_{\mathrm{in}} = P_{heat}} \quad \rightarrow \quad \boxed{c_p \dot{m} \Delta T = P_{heat} = q}$$

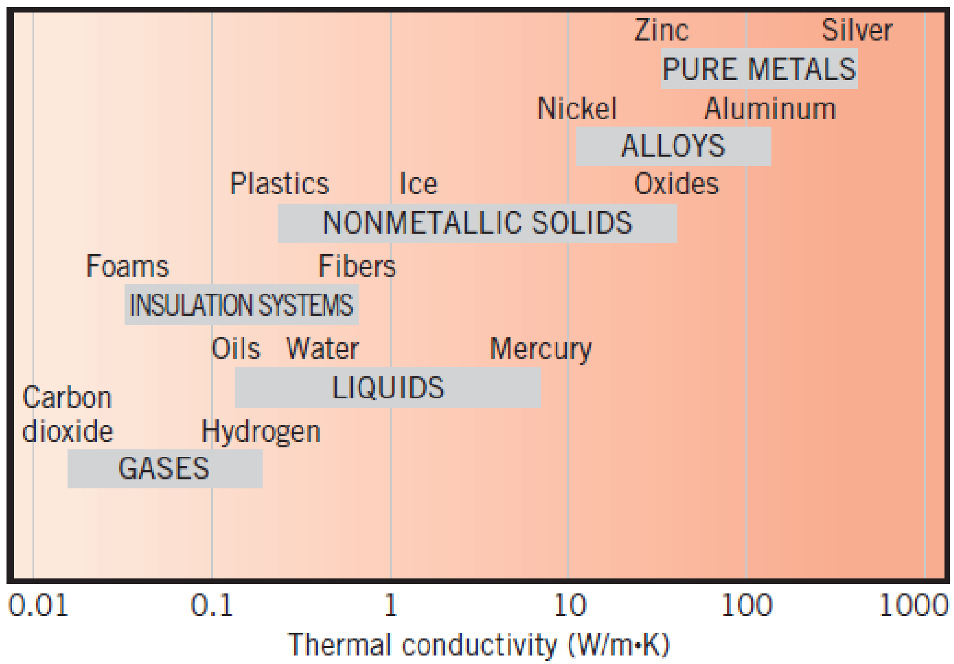
Cool fluid enters: J/s



Heated fluid exits: J/s

Heat = Power = J/s

Even convective heat transfer problems involve typically conduction: Thermal conductivity vs diffusivity



density

thermal conductivity → $\boxed{k=\alpha\,\rho\,c_p}$ ← specific heat

thermal diffusivity

$$\boxed{[c_p]=J/kg\cdot K\ \ [\rho]=kg/m^3\ \ [\alpha]=m^2/s}$$

Some thermal properties for air, water, aluminum and copper

Table: Some material property <u>estimates</u> close to NTP conditions (see: Inc.deWitt Appendix)

Substance	Density [kg/m³]	Specific heat [kJ/kgK]	Thermal conductivity [W/mK]	Thermal diffusivity [m²/s]
Air	1.2	1.007	0.026	$\sim 1.6 \cdot 10^{-5}$
Water	1000	4.217	0.569	$\sim 10^{-6}$
Aluminum	2700	0.900	237	$\sim 0.97 \cdot 10^{-4}$
Copper	8933	0.385	401	$\sim 1.2 \cdot 10^{-4}$
Iron	7870	0.447	80.2	$\sim 10^{-5}$

Water vs air as coolants

- By Fourier's law the heat flux depends on temperature gradient and thermal conductivity
- For a given temperature gradient, heat flux ratio and thermal capacitance ratios are:

$$\frac{k_{water}}{k_{air}} \approx 22 \qquad\qquad \frac{\rho_{water} c_{p,water}}{\rho_{air} c_{p,air}} \approx 3500$$

- These matters explain why water is much more efficient heat exchange fluid than air offering e.g. more compact heat exchanger (fin) design
- Air and water are by far the most common heat transfer fluids

Ordinary differential equations vs partial differential equations

- Example ODE:

$$\frac{dy}{dt} = -y(t)/\tau,\ \tau = \text{const.}$$

Initial condition:

$$y(t=0) = y_o$$

Here: solution to an ODE/PDE gives

ODE → the unknown function $y=y(t)$ which could represent at given time e.g. average radioactivity of an object, average temperature, average concentration, …

PDE → the unknown function $\varphi=\varphi(x,t)$ which could represent at given time and point *e.g. radioactivity, temperature, …*

- Example PDE:

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = \nu \frac{\partial^2 \phi}{\partial x^2},\ 0 < x < L$$

Initial condition

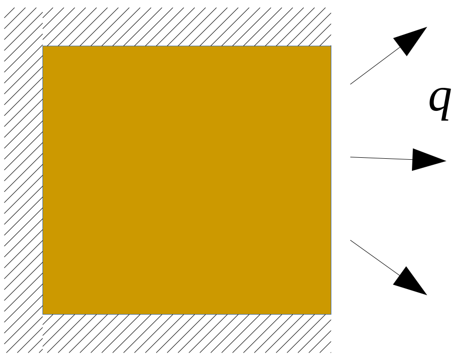
$$\phi(x, t=0) = \phi_o(x)$$

Boundary condition (here fixed values)

$$\phi(x=0, t) = \phi_{1,}\ \phi(x=L, t) = \phi_2$$

Example ordinary differential equation

- A storage box filled with air is insulated at initial temperature T_o and heat escapes via glass window at rate q ([q]=W). Average temperature $T=T(t)$=?



$$E(t=0)=c_p m T_o$$

$$\Delta E = c_p m \Delta T = q \Delta t$$

$$mc_p \frac{dT}{dt} = -q$$

$$dT = \frac{-q}{mc_p} dt$$

$$\int_{T_o}^{T} dT = -\int_0^t \frac{q}{mc_p} dt \quad \Rightarrow \quad T(t) = \frac{-q}{mc_p} t + T_o$$

Newton’s cooling law

- **Newton's cooling law:** Rate of change of heat (**W=J/s**) for an object is proportional to temperature difference between the object and its surroundings.

$$\boxed{mc_p\frac{dT}{dt}=-hA_s\left(T-T_\infty\right)}$$

- The temperature $T=T(t)$ could represent e.g. the average temperature of a beverage in the fridge.
- h = heat transfer coefficient (depends on air flow around the object)
- A_s = object surface temperature
- T_∞ = ambient temperature assumed constant here

$$\boxed{q=hA_s\left(T_s-T_\infty\right)}$$

$$\boxed{[q]=J/s,[m]=kg,[c_p]=J/kg\cdot K,[T]=K,[h]=W/m^2K,[A_s]=m^2}$$

Fourier's law

- **Fourier's law:** Heat flux results from a temperature gradient.

$$\boxed{q'' = -k \nabla T}$$

$$\boxed{[q''] = W/m^2, [T] = K, [k] = W/m\,K, [\nabla T] = K/m}$$

- **Fourier's law in 1d:**

$$\boxed{q'' = -k \frac{\partial T}{\partial x} = -k \frac{\Delta T}{\Delta x}}$$

- **Heat rate vs heat flux:**

$$\boxed{[q] = W, q = q'' A}$$

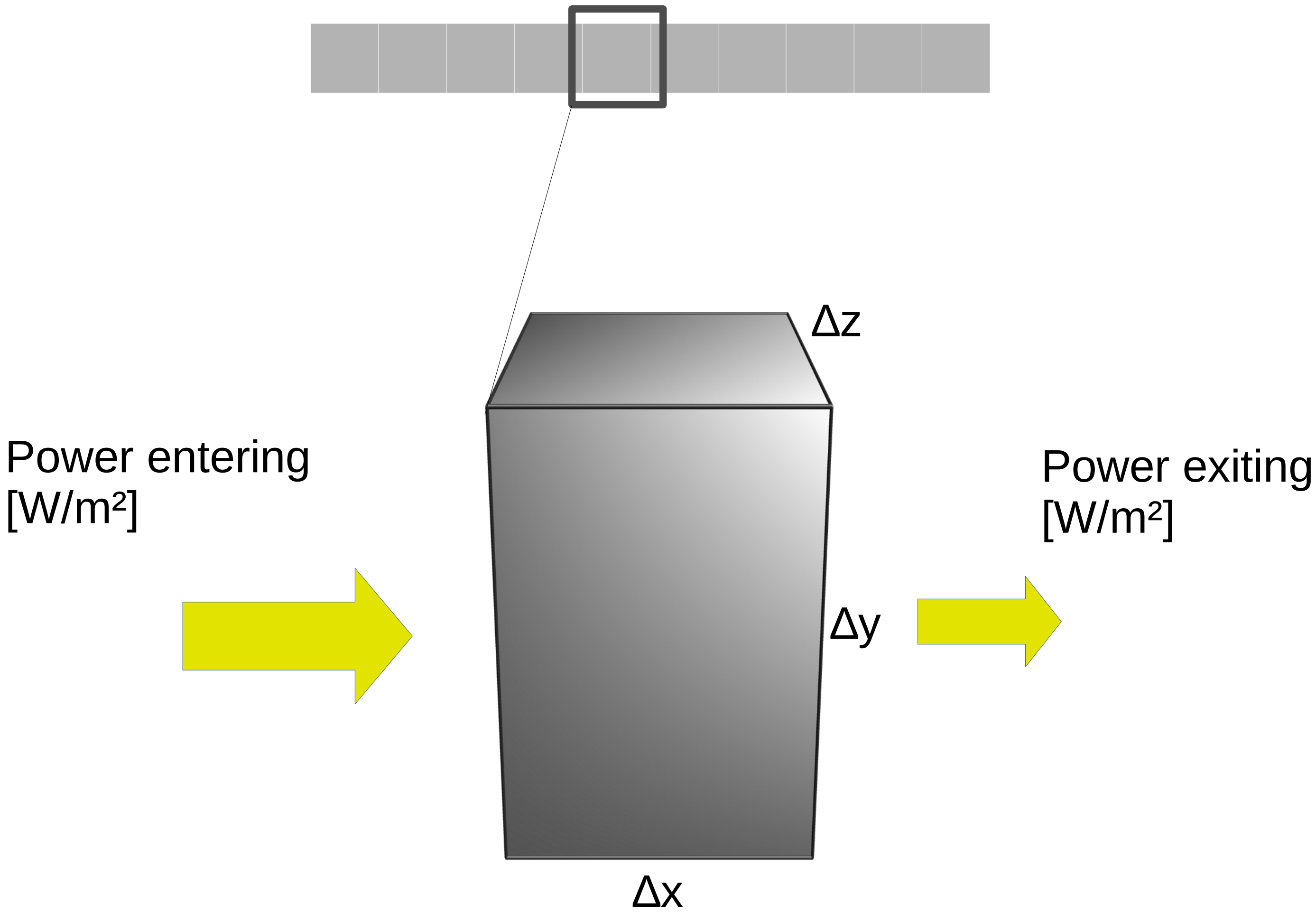
Temperature levels across a window

Remark 1: if we know surface temperatures T_1 and T_2 on both sides of a window it is easy to calculate escaping heat flux using Fourier's law. But, in practice we seldom know those surface temperature values.

Remark 2: T_1 and T_2 do not correspond to the indoor room temperature and outdoor temperature values. Also airflow on boths sides of the windows will affect the actual surface temperature values.

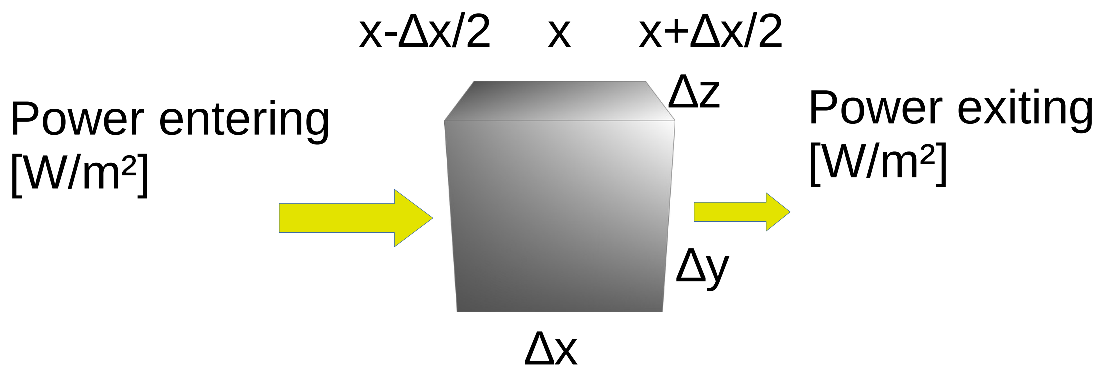
Remark 3: It is easy to measure T_{room} and T_{out}

Consider heat conduction in 1d e.g. in metal rod or through the window. Divide the 1d object into small elements and carry out energy balance analysis for 1 of those elements. Assume: no heat losses.



Derivation of heat equation:

Next we apply energy conservation law for a small infinitesimal volume assuming conduction only
(e.g.1d metal rod)



$$\boxed{\rho c_p \Delta T(x,t) \Delta x \Delta y \Delta z = [k \frac{\partial T(x+\Delta x/2,t)}{\partial x} - k \frac{\partial T(x-\Delta x/2,t)}{\partial x}] \Delta y \Delta z \Delta t}$$

Energy change in a short time [J]

Power flux exiting [W/m²] i.e. Fourier’s law

Power flux entering [W/m²] i.e. Fourier’s law

Then: Divide both sides by $\Delta x \Delta y \Delta z \Delta t$ and take the limit when all Δ-variables → 0 → We get the heat equation.

Heat Equation

- Heat equation is a partial differential equation describing heat diffusion
- Solution of heat equation offers temperature distribution in a solid or fluid (gas or liquid) as a function of space and time i.e. ***T=T(x,t)***

$$\boxed{\rho\, c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right)}$$

- ***Important:*** *Heat equation above is a general 1d energy conservation law when only heat conduction i.e. diffusion is taken into account.*
- ***Note***: *If second space derivative is positive/negative, then function T has a local minimum/maximum and temperature changes towards positive/negative i.e. heat flows from hot to cold.*
- To solve the heat equation, also **initial conditions** (IC's) and **boundary conditions** (BC's) are needed
- On the present course **we solve heat equation by computer.**

Example: Steady State Solution of the Heat Equation with Fixed Temperature boundary conditions at both ends

- In steady state time approaches infinity and we can write:

$$\boxed{\begin{gathered} 0 = \frac{\partial}{\partial x}\left(\alpha \frac{\partial T}{\partial x}\right), 0 \le x \le L \\ T(x=0) = T_1 \text{ and } T(x=L) = T_2 \end{gathered}}$$

- Integrate twice to obtain:

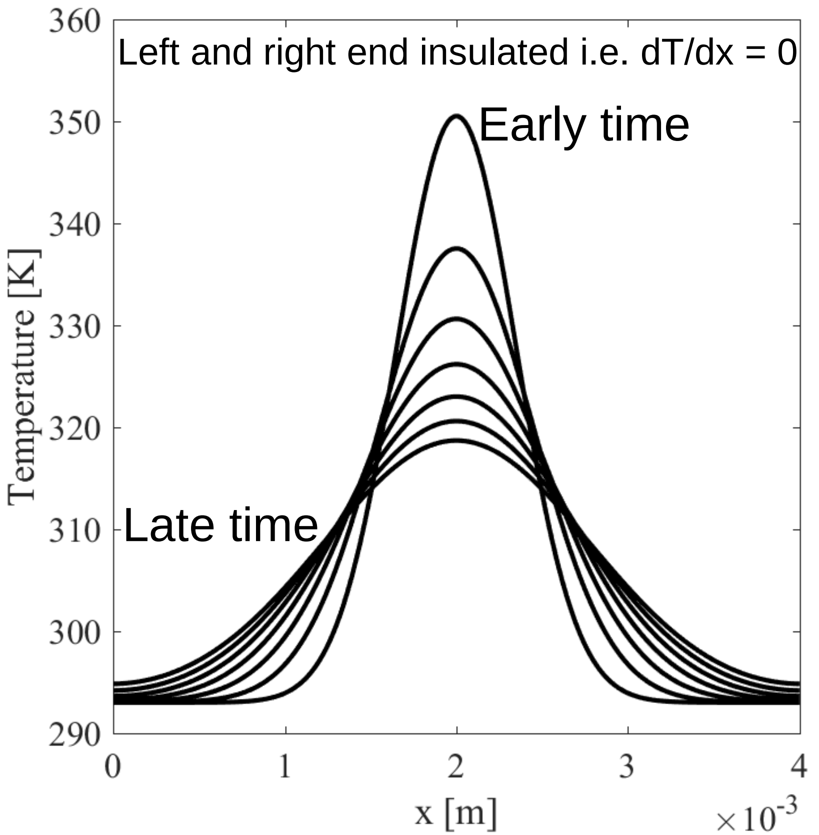
$$\boxed{T(x,t) = A + Bx}$$

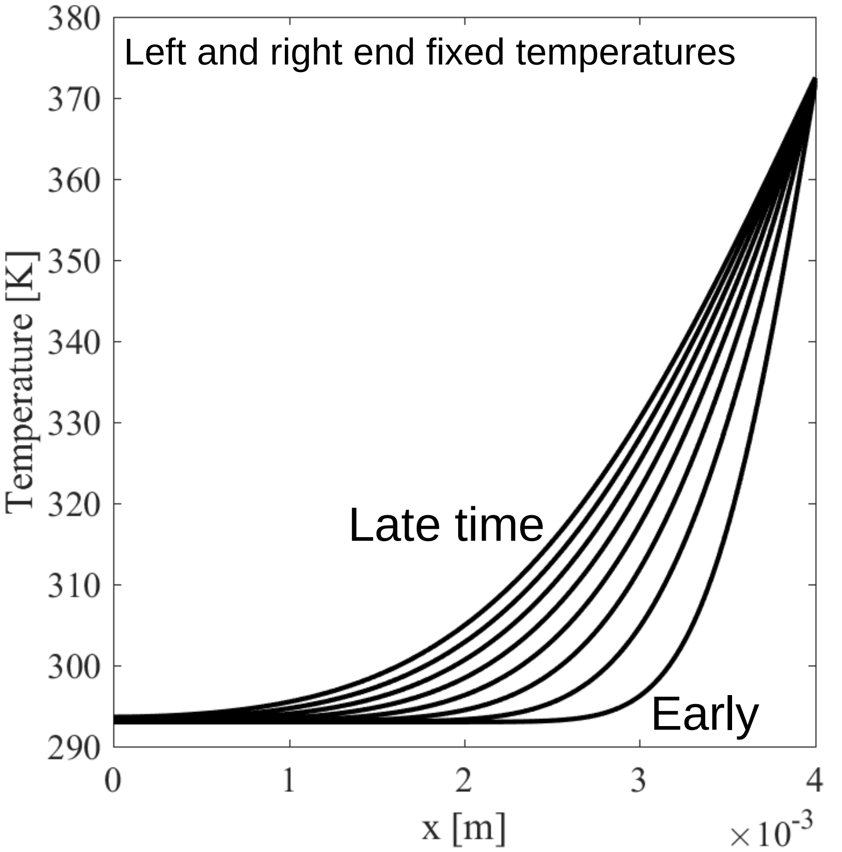
- The requirement to fulfill BC's gives:

$$\boxed{T(x,t) = T_1 + (T_2 - T_1)\, x/L}$$

Example solutions of heat equation with two different boundary conditions

- Diffusive processes are very slow in comparison to convective processes
- Below, two examples of heat diffusion in iron (profiles taken from different times)
- Simulation time is in the order of 0.03-0.1s





Example: Time-Dependent Analytic Solution of the Heat Equation in a Periodic (Infinite) Domain

- Assuming constant properties, it is convenient to write:

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

- In a periodic domain of length L (m) with trigonometric (sinusoidal) initial condition, $T(x,t=0)=T_o+T_1 sin(kx)$ the equation can be easily solved for unknown temperature
- It is noted that a general solution is of the form

$$T(x,t) = T_o + T_1 \sin(kx) \exp(-k^2 \alpha t)$$

where the **wavenumber** $\boldsymbol{k=2\pi/L}$.

- Exercise: show that the solution above fulfills the equation by inserting it to the heat eqn.

How Long Time Would it Take for the Heat to Diffuse Across Distance L?

- The earlier considered periodic solution in an infinite domain is:

$$\boxed{T(x,t)=T_o+T_1\sin(kx)\exp(-k^2\alpha t)}$$

- The exponential term has a timescale (think in form exp(-t/τ)):

$$\boxed{\tau=1/k^2\alpha=\frac{L^2}{4\pi^2\alpha}}$$

- The diffusion time is noted to be $\tau_{diff} \sim L^2/\alpha$
- This means essentially that if you double the distance (think doubling thickness of a wall of building) it will take four times longer time for heat to diffuse across that distance.

Lecture 1.2 Numerical approach: Newton's cooling law and 1d heat equation

ILO 1: Student can derive and explain physical origin of the heat equation, describe solution behavior by example solutions and boundary conditions, <u>and solve the heat equation (1d) and Newton's cooling law (0d) numerically in Matlab</u>.

HW1: Newton's cooling law applied for a soda-can example solved numerically in Matlab

Recall: Newton's cooling law

$$\frac{dT}{dt} = \frac{-hA_s}{c_p m}\left(T - T_\infty\right)$$

$$T\left(t=0\right) = T_o \text{ initial condition}$$

Analytical solution

$$T\left(t\right) = \left(T_o - T_\infty\right)\exp\left(\frac{-hA_s}{c_p m} t\right) + T_\infty$$

Analytical solution exists → good starting point for the computer learning: how to numerically solve temperature development in the above equation?

Recap: simple ODE's can be solved by separation of variables (relevance: HW1)

Simple ODE for unknown function $y=y(t)$

$$\frac{dy}{dt}=-a\,y$$

$$y(t=0)=y_o \text{ initial condition and a = const.}$$

Separate *y* and *t* containing parts to different sides and integrate:

$$\int\frac{dy}{y}=-\int a dt$$

$$\log(y)=-(at+D),\ \text{where a=integration const.}$$

By using the initial condition it is easy to see that:

$$y(t)=y_o\exp(-at)$$

Solving temperature numerically over a short time interval Δt (timestep)

$$\boxed{\frac{dT}{dt}=\frac{-hA_s}{c_p m}(T-T_\infty)} \rightarrow \boxed{dT=-dt\frac{hA_s}{c_p m}(T-T_\infty)}$$

$$\rightarrow \boxed{\Delta T=-\Delta t\frac{hA_s}{c_p m}(T-T_\infty)} \rightarrow \boxed{\Delta T_n=-\Delta t\frac{hA_s}{c_p m}(T_n-T_\infty)}$$

Find new temperature using the Euler method

$$\rightarrow \boxed{T_{n+1}=T_n+\Delta T_n}$$

Solution proceeds in discrete timesteps

$$\boxed{t_n=n\Delta t, n=0,1,2,\ldots}$$

Pseudo-code to solve Newton's cooling law by explicit Euler method

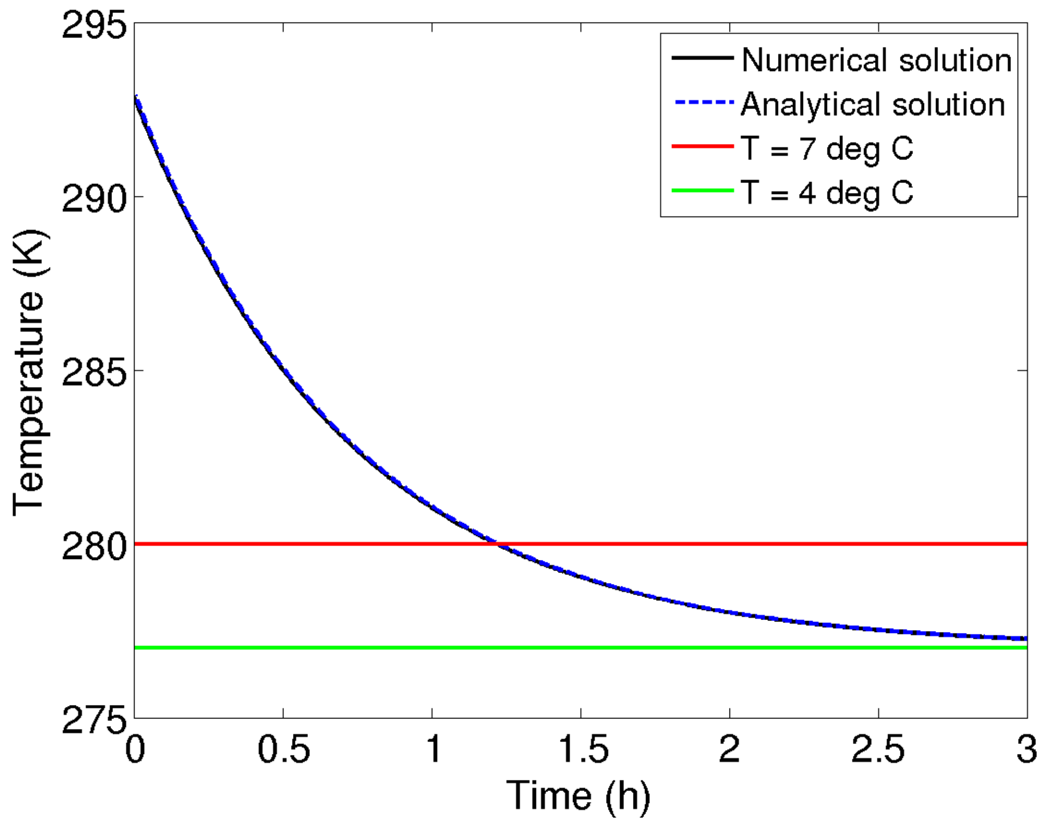
Step 0: T_o *known*

Step 1: $\Delta T_n = -\Delta t \dfrac{hA_s}{c_p m}\left(T_n - T_\infty\right)$

Step 2: $T_{n+1} = T_n + \Delta T_n$

Step 3: go to Step 1 until simulation time exceeded

Temperature of a cooling soda can computed by Newton's cooling law numerically and analytically



Matlab implementation

Program: /Example0d/cool0d.m
Execution: **`>> cool0d`**

What it does: Solves 0d Newton's cooling law for temperature of a “0d” drink can.

Snapshot of code that does the job:

```
cp = 4190;                  % specific heat J/kgK
dt = 20;                    % timestep in s
To = 273+20;                % initial temperature K
Tinf=273+4;                 % fridge temperature K
simutime   = 3*3600;        % simulation time s
simusteps = round(simutime/dt);
T   = To;                    % initial temperature

% h=heat transfer coefficient W/(m^2K)
H = ...;
% As=surface area
As = …;

for(k=1:simusteps)
    dT = -(h*As/(m*cp))*dt*(T-Tinf);
    T = T+dT;
    Tcol(k) = T; % collect temperatures to Tcol
end
```

HOW TO IMPLEMENT THIS IN PRACTICE?

→ open Matlab terminal
→ open text editor
→ create new file with some name e.g. cool0d.m
→ add the text from the left to file cool0d.m
→ run by typing text cool0d on terminal

Plotting the results

1) e.g. **`plot(x,y,'k-')`** where x and y are vectors

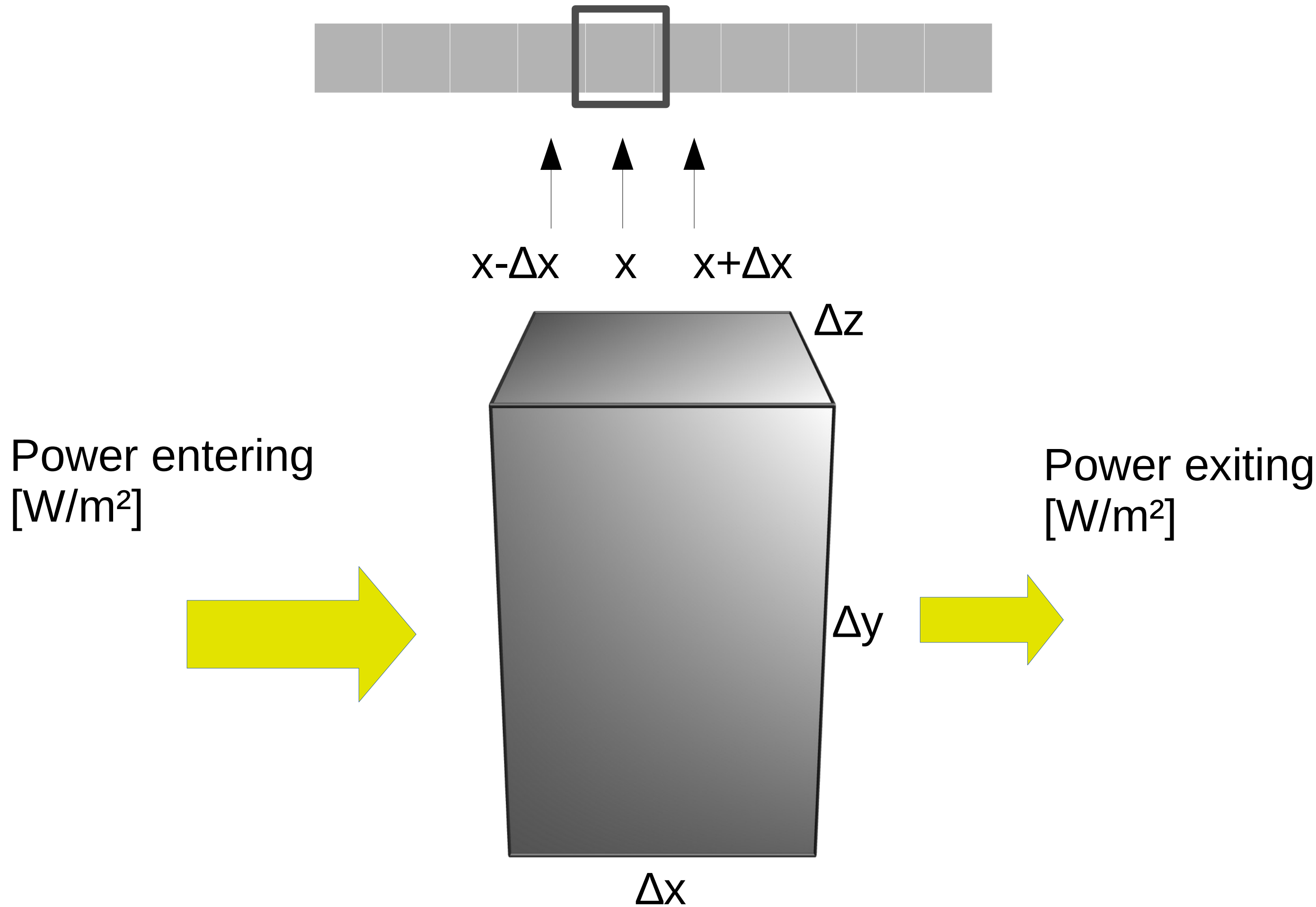
2) Note: length(x)=length(y)

3) **`>> help plot`**

```
figure(1), clf, box, hold on
alltime = linspace(0,simutime/3600, simusteps);
plot(alltime, Tcol, 'k-','Linewidth',2)
plot(alltime, (To-Tinf)*(exp(-h*As*3600*alltime/(m*cp)))
+ Tinf, 'b--','Linewidth',2)
plot(alltime, (273+7)*(ones(length(alltime),1)), 'r-',
'Linewidth',2)
plot(alltime, (Tinf)*(ones(length(alltime),1)), 'g-',
'Linewidth',2)

h=xlabel('Time (h)');
h=ylabel('Temperature (K)');
h=legend('Numerical solution',  'Analytical solution','T
= 7 deg C','T = 4 deg C'); set(h,'Fontsize', 16)
print -dpng TcoolingCan
```

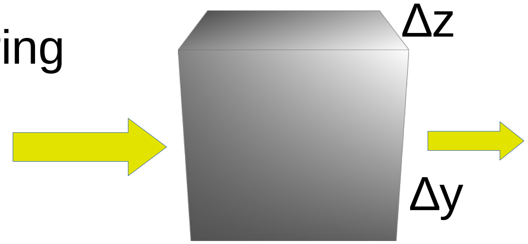
HW1: Heat equation solved in 1d by finite difference method in Matlab

Last week: Consider heat conduction in 1d e.g. in metal rod.
Divide the 1d object into small elements and carry out energy balance analysis for 1 of those elements. Assume: no heat losses.



Recall from last week → derivation of heat equation:

Next we apply energy conservation law for a small infinitesimal volume assuming conduction only (e.g.1d metal rod)



$$\boxed{\rho c_p \Delta T(x,t)\Delta x \Delta y \Delta z = [k \frac{\partial T(x+\Delta x/2,t)}{\partial x} - k \frac{\partial T(x-\Delta x/2,t)}{\partial x}]\Delta y \Delta z \Delta t}$$

Energy change in a short time [J]

Power flux exiting [W/m²] i.e. Fourier's law

Power flux entering [W/m²] i.e. Fourier's law

Then: Divide both sides by $\Delta x \Delta y \Delta z \Delta t$.

Heat Equation and the Finite Difference Method

Theory: Let Δ-variables → 0
→ heat equation

$$\boxed{\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}}$$

In simulations, Δt and Δx are of finite size
→"finite difference" approximation of heat equation

$$\boxed{\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{\left(T_{i+1}^n - T_i^n\right) - \left(T_i^n - T_{i-1}^n\right)}{\Delta x^2}}$$

$$\boxed{T_i^{n+1} = T_i^n + \Delta t \, \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}}$$

Courant-Friedrichs-Lewy number (CFL<0.5 for stability).

$$\boxed{CFL = \frac{\alpha \Delta t}{\Delta x^2}}$$

Now we have an explicit update scheme for T in each discrete grid point i. This is the explicit Euler scheme (most simple timestepping).

“Numerical solution of heat equation” is a “solution at discrete data points”

- Heat equation is already quite challenging equation to solve by pen/paper even in simple cases
- Typically, even if it would be possible to obtain an analytical solution, one would need a computer to evaluate/visualize the solution (e.g. sum of infinite Fourier series)
- **Discretization of solution points** means that in numerics e.g. temperature is evaluated in a finite value of evaluation points in space and time e.g. $T(x,t) \rightarrow T(x_i,t_i)$ where $x_i = i\Delta x$ and $t_n = n\Delta t$
- **Discretization of partial derivatives** means that the continuous partial derivatives are replaced by discrete finite difference estimations

Numerical approximation of partial derivatives

- Conduction and convection of temperature (energy conservation) can be generally described by the convection-diffusion equation.
- Finite difference formulas offer a way to approximate partial derivatives
- Once partial derivatives are known in space and time, then one obtains a way to solve temperature distributions
- The following convection-diffusion equation type appears commonly on this course (u = fluid velocity, α = thermal diffusivity).

$$\boxed{\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} = \alpha\frac{\partial^2 T}{\partial x^2}}$$

$$\boxed{\left(\frac{\partial T}{\partial t}\right)_i^n \approx \frac{T_i^{n+1} - T_i^n}{\Delta t}}$$

1st order Euler formula for time derivative at fixed space point.

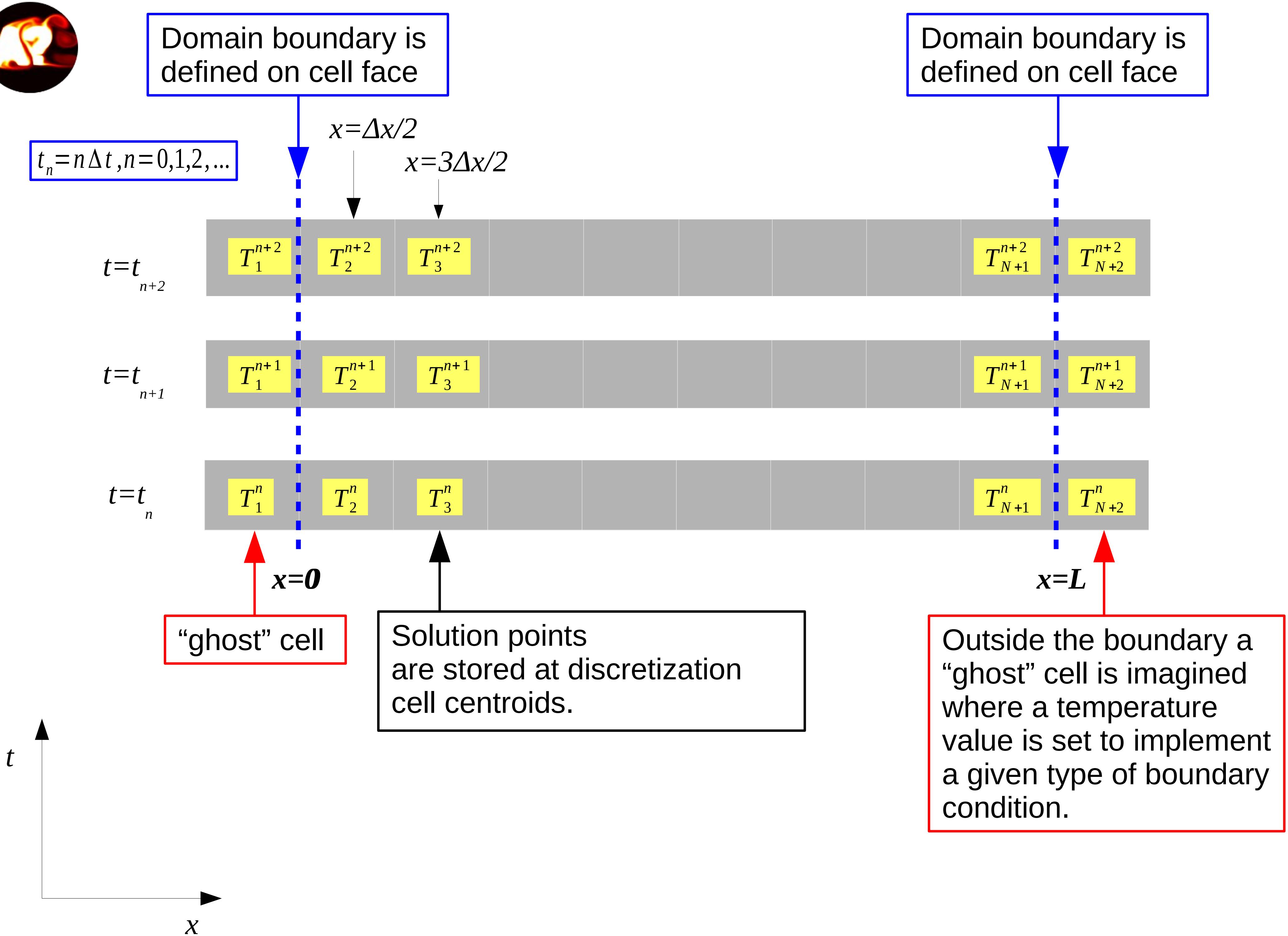
$$\boxed{\left(\frac{\partial T}{\partial x}\right)_i^n \approx \frac{T_{i+1}^n - T_{i-1}^n}{2\Delta x}}$$

2nd order central difference for 1st space derivative at a fixed time.

$$\boxed{\left(\frac{\partial^2 T}{\partial x^2}\right)_i^n \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}}$$

2nd order central difference for 2nd space derivative at a fixed time.

Observation: if the solution points from time level n are known in each point I the new solution values at time level n+1 can be solved for.


Domain boundary is defined on cell face
Domain boundary is defined on cell face
$t_n = n\Delta t, n = 0,1,2,...$
$x = \Delta x/2$
$x = 3\Delta x/2$
$t = t_{n+2}$
T_1^{n+2} T_2^{n+2} T_3^{n+2} T_{N+1}^{n+2} T_{N+2}^{n+2}
$t = t_{n+1}$
T_1^{n+1} T_2^{n+1} T_3^{n+1} T_{N+1}^{n+1} T_{N+2}^{n+1}
$t = t_n$
T_1^{n} T_2^{n} T_3^{n} T_{N+1}^{n} T_{N+2}^{n}
x=0
x=L
“ghost” cell
Solution points are stored at discretization cell centroids.
Outside the boundary a “ghost” cell is imagined where a temperature value is set to implement a given type of boundary condition.
t
x

Discretization of 1d Heat Equation by Finite Difference Method

Continuous PDE

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

Discretized PDE

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

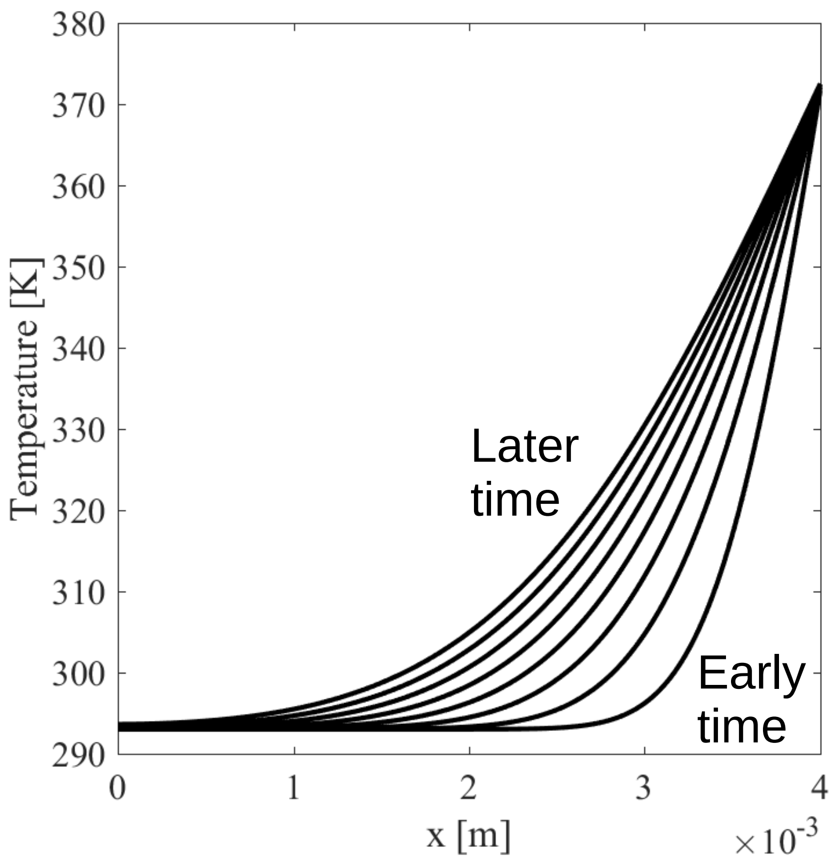
$$T_i^{n+1} = T_i^n + \Delta t \, \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

Courant-Friedrichs-Lewy number (CFL<0.5 for stability).

$$CFL = \frac{\alpha \Delta t}{\Delta x^2}$$

Now we have an explicit update scheme for T in each discrete grid point i. This is the explicit Euler scheme (most simple timestepping).

Numerical solution of heat eqn at different times and values stored in a table. Here: T_{right} = 373K and T_{left} = 293K



Here: 198 internal cells, 2 ghost cells

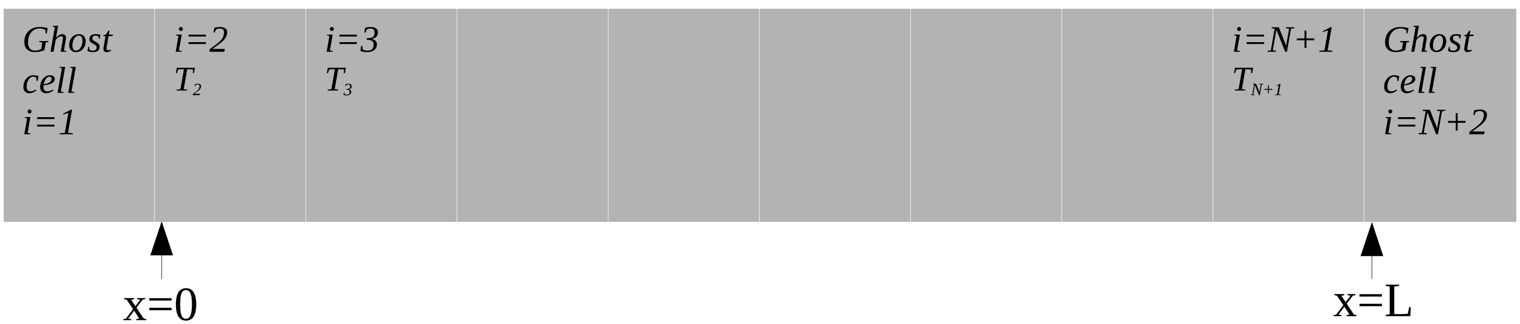
```
>> T

ans =

   373.7
   372.3
   372.2
   372.1
   372.0
   371.9
   371.7
   ...
```

"Ghost cell" value

First cell inside the domain at hot end

Boundary condition types

- **The problem:** some numerical value needs to be assigned to the **”ghost cells”**
- **Case 1:** Boundary temperature fixed → boundary heat flux follows
- **Case 2:** Boundary heat flux zero (insulated) → zero temperature gradient through boundary (Fourier’s law: heat flux = 0)
- **Case 3:** Boundary heat flux fixed → boundary temperature follows.



Case 1: $$\left(T_1^n + T_2^n\right)/2 = T_{min}$$

Case 2: $$T_1^n = T_2^n \qquad\qquad T_{N+1}^n = T_{N+2}^n$$

Case 3: $$-k\left(T_2^n - T_1^n\right)/\Delta x = q_L$$

→ In all the cases a “ghost cell” value is needed.
→ **Ghost cell:** we can imagine a virtual cell outside the domain where we enter a temperature value so that the desired BC becomes exactly fulfilled.

Update scheme for 1d heat equation

1) Set boundary conditions to ghost cells 1 and N+2 using T from step n.

2) Update new temperature at timestep n+1 in the internal cells 2...N+1

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

3) Update time according to t = t + dt

4) Go back to 1)

This update scheme is very easy to program in Matlab `for`-loop

Program: /Example1d/HeatDiffusion.m

Execution: `>> HeatDiffusion`

What it does: Solves 1d heat equation in equispaced grid, fixed T_{left} and T_{right}.

Main for-loop:

```
for(t=1:K)
    % set boundary conditions
    T(1)  = 2*Tleft - T(2); T(N+2) = 2*Tright - T(N+1);

    % update temperature in inner points
    T(in) = T(in) + (dt*kappa/dx^2)*(T(in+1)-2*T(in)+T(in-1));
end
```

Note: I use constantly the “trick” which makes Matlab-programs often very fast.

```
% define a table which refers to the 'inner points'
in = 2:(N+1);
```

Example for N+1 = 5