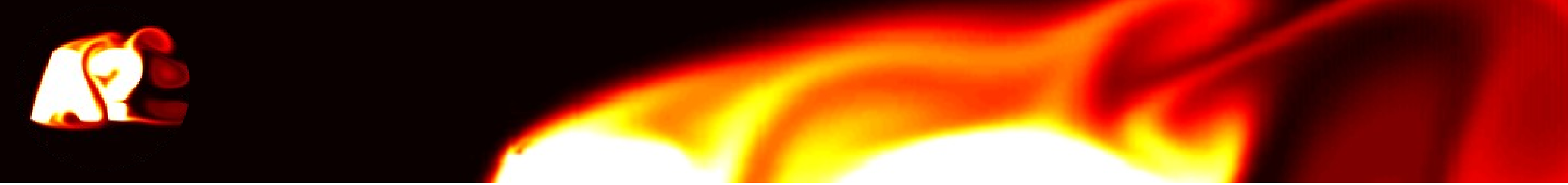


EEN-1020 Heat transfer

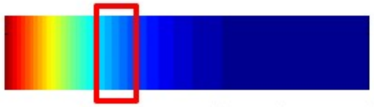
Week 1: Fourier's law, heat equation, Newton's law and numerical solution

Prof. Ville Vuorinen

October 27th - 28th 2020

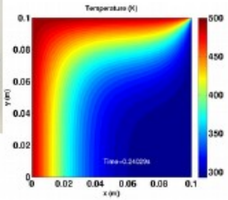
Aalto University, School of Engineering

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

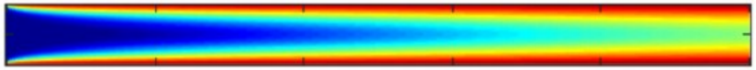


$$\frac{\partial T}{\partial t} \approx \frac{1}{\Delta x}\left(k\frac{\partial T(x+\Delta x/2,t)}{\partial x} - k\frac{\partial T(x-\Delta x/2,t)}{\partial x}\right)$$

Week 2: Fin theory, conduction, intro to convection

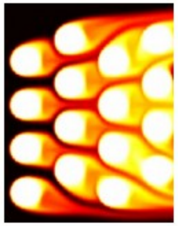


Week 3: convective heat transfer – internal flow (channel)



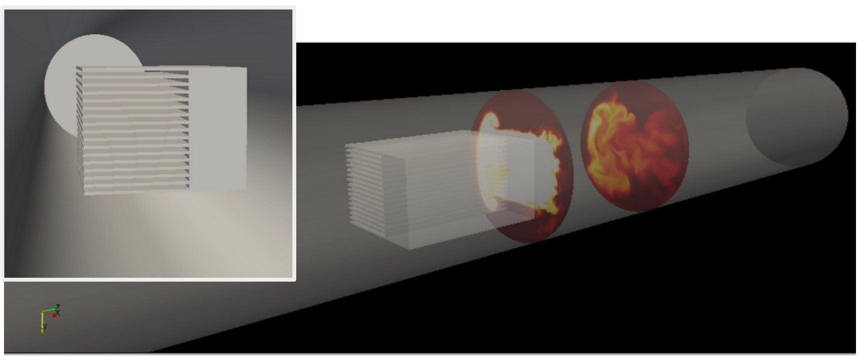
Week 4: convective heat transfer – external flow (fin systems)

Week 5: natural convection, boiling, correlations



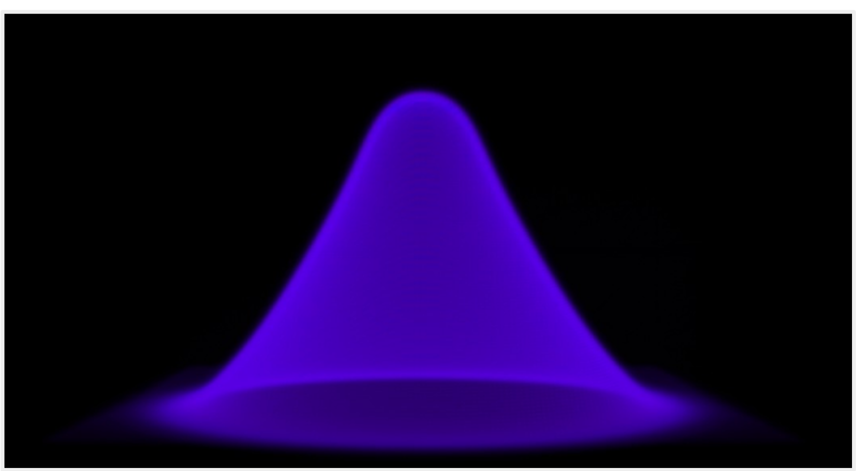
Understanding heat transfer and fluid dynamics is essential in design of energy efficient applications

Heating and cooling



Combustion

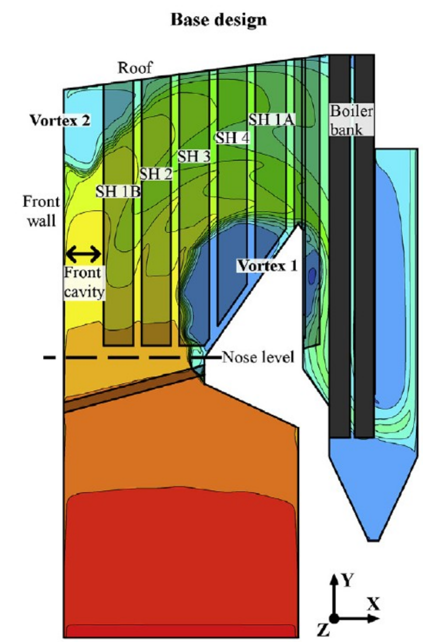
P.Peltonen



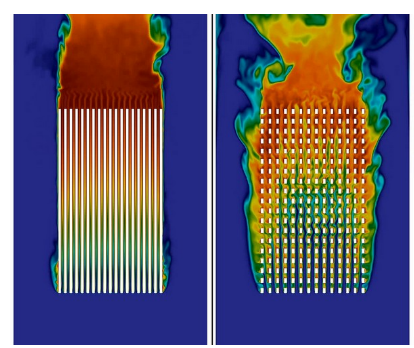
H.Kahila

Small and large scale energy production

Peltonen,Vuorinen (2019)

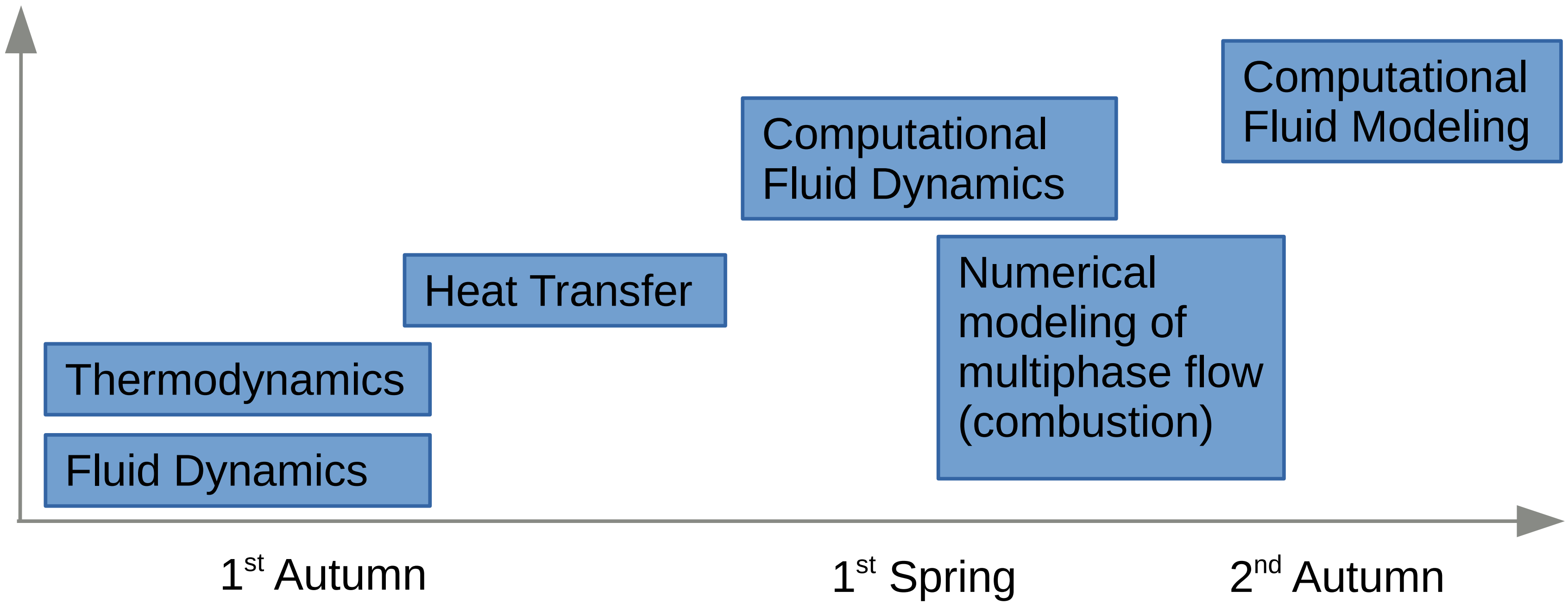


Maakala, Järvinen, Vuorinen, Energy, 160, 361-377, (2018)



Heat exchangers

Positioning heat transfer in AEE program e.g. computational study path in SEC major



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

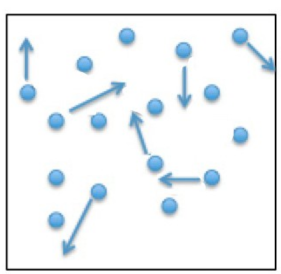
- **For gases or liquids:** temperature is actually closely related to the speed of the molecules on the molecular scales (molecules bouncing around). Molecular speeds are much higher (e.g. 2000-10000 m/s) than macroscopic fluid flow velocities (e.g. 0.1-10 m/s) in cooling/heating applications.
- **For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

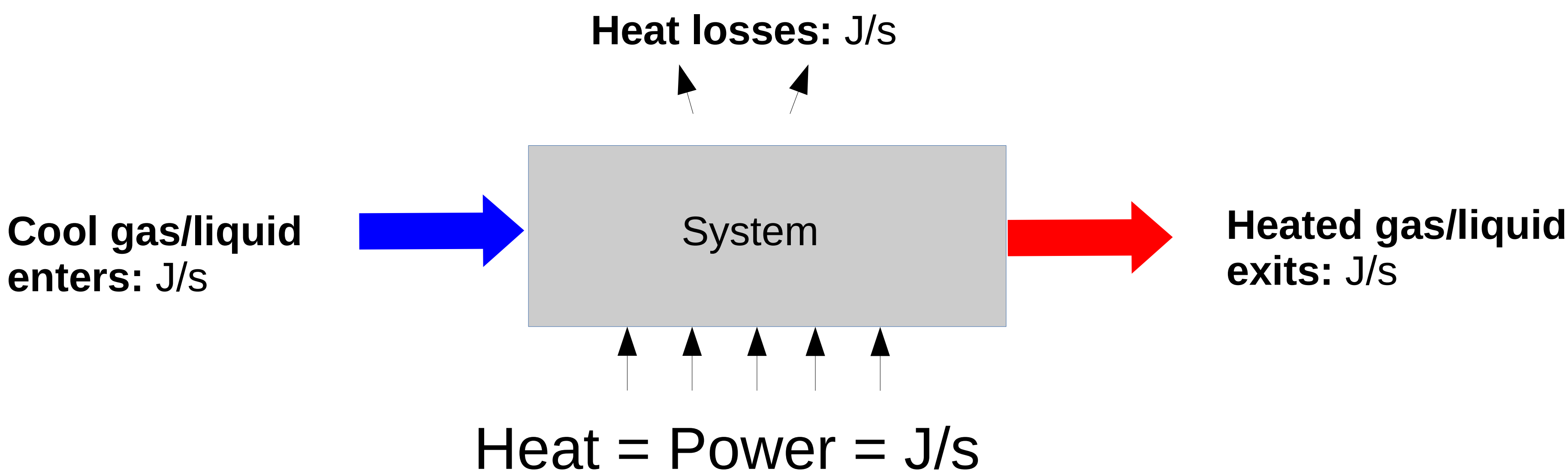
Mean squared molecule velocity relates to temperature.

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

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 (the “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.



Thermal energy and mass conservation

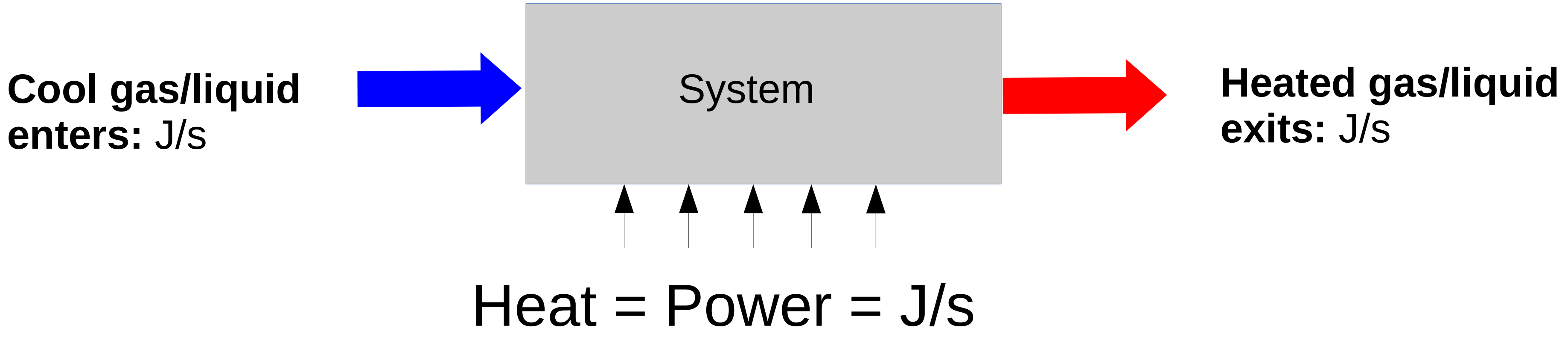
(assume here: losses small)

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

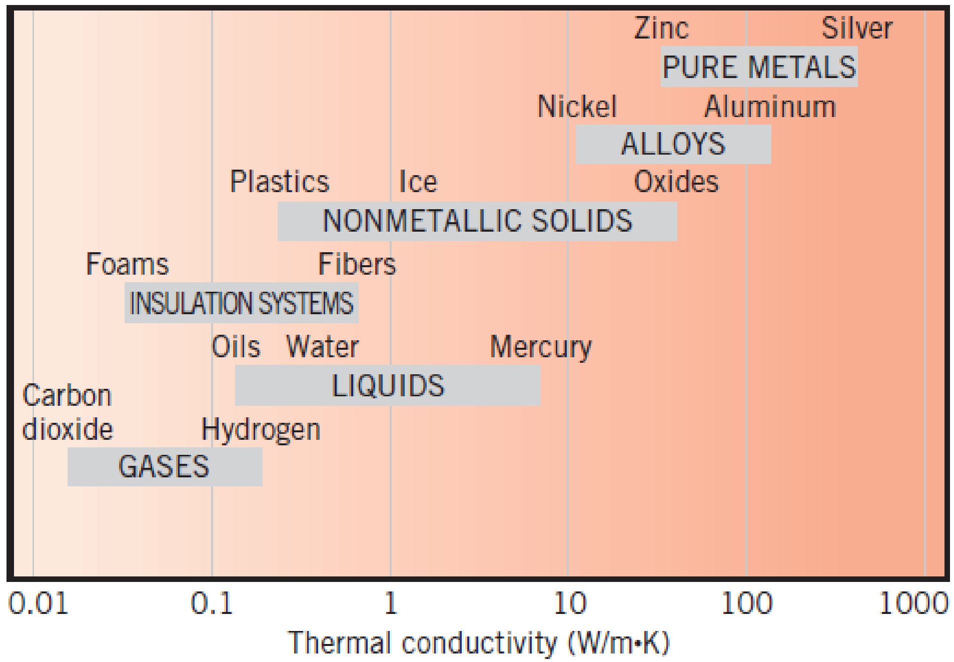
$$\rho U_{\text{in}} A_{\text{in}} = \rho U_{out} A_{out} = \dot{m}$$

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

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



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



density

thermal conductivity → $k = \alpha \rho c_p$ ← specific heat

thermal diffusivity

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

Some thermal properties for air, water, aluminum and copper

Table: Some material property estimates 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:

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

- 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

Dimensional analysis – "light version" - a useful tool to reason dependencies and eliminate redundant parameters

- **Example**: [Velocity] = m/s
  - → velocity depends on [x]=m and [t]=s
  - → We make an Ansatz: $u = x^a t^b$
  - → $m/s = m^a s^b$
  - → For the units on lhs and rhs to match, the only option is: $a=1$, $b = -1$

- **Example**: [Thermal energy flux of flowing fluid] = [φ] = J/m^2s = W/m^2
  - → energy flux depends on
  - [ρ]=kg/m^3
  - [U]=m/s
  - [c_p] = J/kgK
  - [T] = K
  - → We make an Ansatz: $\varphi = \rho^a U^b c_p^{\,c} T^d$
  - → For the units on lhs and rhs to match, the only option is: $a=b=c=d=1$ and $\varphi = \rho U c_p T$

Ordinary differential equations vs partial differential equations on the course

- Example ODE:

$$\frac{dy}{dt} = -y(t)$$

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, temperatur , …*

- 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

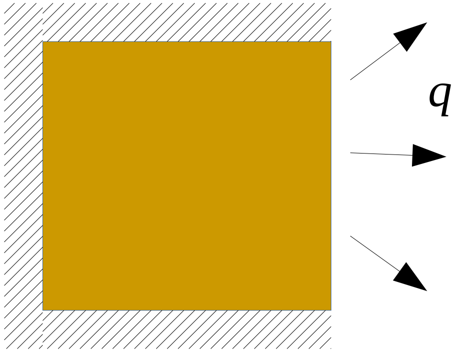
$$\phi(x, t=0) = \phi_o(x)$$

Boundary condition (here fixed values)

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

Example ordinary differential equation

- A storage box is insulated at initial temperature T_o and heat escapes only through one side at rate q ([q]=W). Find the average box temperature $T=T(t)$.



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

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

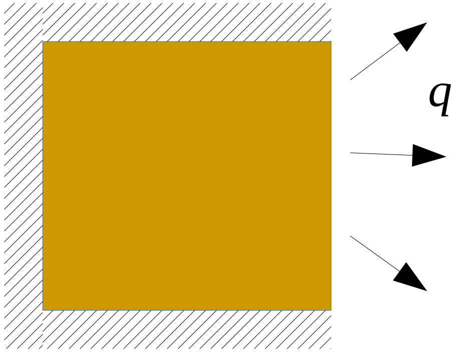
$$mc_p \frac{dT}{dt} = -q$$

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

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

Example ordinary differential equation

- A storage box is insulated at initial temperature T_o and heat escapes only through one side at rate q ([q]=W). Find the average box temperature $T=T(t)$.



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

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

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

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

Note: this is formally correct for $T>0$.

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

Newton's cooling law – an engineering tool

- **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.

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

$$T_\infty = const.$$

- The temperature $T=T(t)$ could represent e.g. the average temperature of a beverage in the fridge.
- More generally, $T=T_s$ i.e. the object surface temperature (either average/representative or even local but then we talk about PDE's as in HW3-4) and heat transfer (either average or local) typically is attempted to be cast in the form below

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

- Here h is the convective heat transfer coefficient

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

Fourier's law – a physical law/principle

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

$$q'' = -k \nabla T$$

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

- **Fourier in 1d:**

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

- **Heat rate vs heat flux:**

$$[q] = W, q = q''A$$

Heat Equation

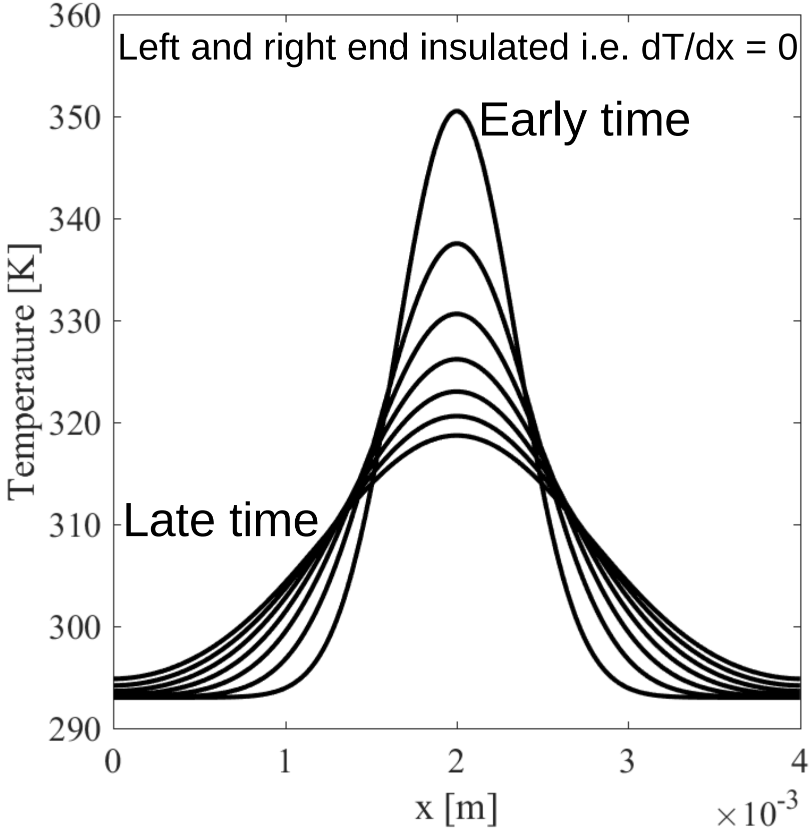
- Heat equation is a partial differential equation describing heat diffusion
- Solution of heat equation offers temperature distribution in a solid or fluid (gas or liquid) as a function of space and time i.e. $T=T(x,t)$

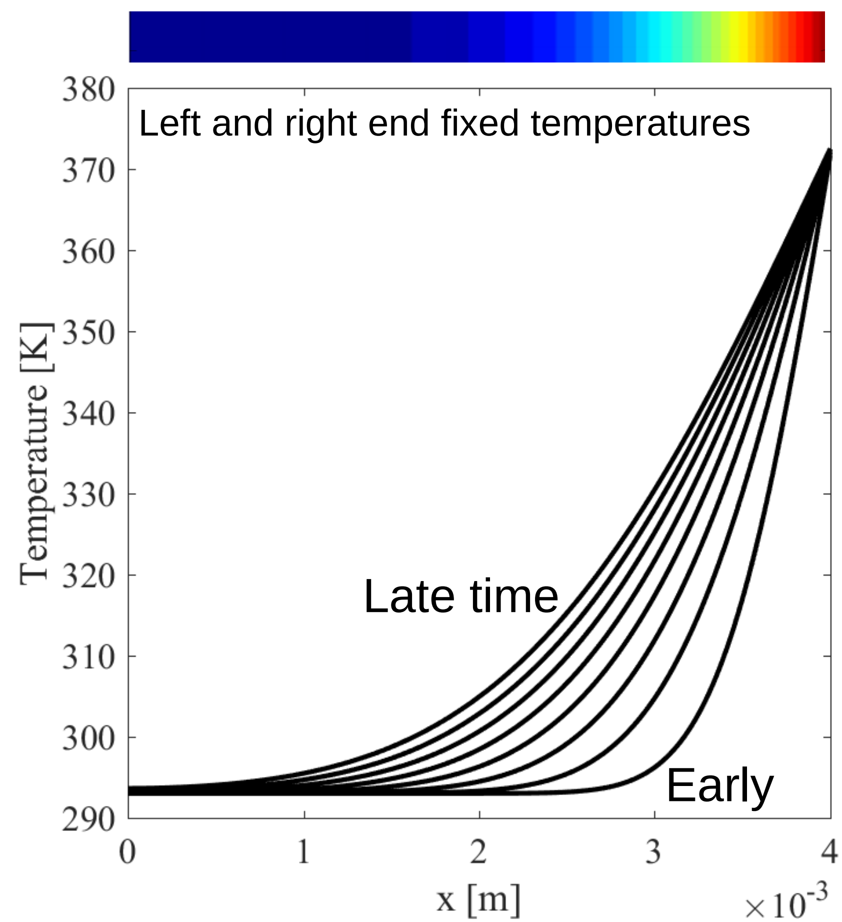
$$\rho c_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right)$$

- ***Important:*** *Heat equation represents simply 1d energy conservation and transport via diffusion.*
- ***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

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





Derivation of the Heat Equation from Energy Conservation Principle

- Heat equation can be derived easily by considering energy balance for an infinitesimal element of fluid/solid and flow of energy through the sides of the element by conduction via Fourier's law

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

Energy change in a short time [J]

Power exiting/entering [W/m²]

Power entering/exiting [W/m²]

- Dividing by $\Delta x \Delta y \Delta z \Delta t$ and taking the limit when all Δ-variables → 0 gives the heat eqn.

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 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

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

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

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

Example: Steady State Solution of the Heat Equation with Fixed Temperature BC's

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

$$0 = \frac{\partial}{\partial x}\left(\alpha \frac{\partial T}{\partial x}\right), 0 \leq x \leq L$$

$$T(x=0) = T_1 \text{ and } T(x=L) = T_2$$

- Integrate twice to obtain:

$$T(x,t) = A + Bx$$

- The requirement to fulfill BC's gives:

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

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

Insight by dimensional analysis:

- The transient heat eqn tells that essentially the only parameter affecting temperature diffusion is α having units m^2/s
- t is unknown but obviously t increases as L having unit meter increases
- Dimensionally, we try to find an expression $t=t(L, \alpha)$
- We make an Ansatz $t_{diff} \sim L^a \alpha^b$
- Match dimensions: $s = m^a m^{2b} s^{-b}$. Thus: $a+2b=0$ and $b=-1 \rightarrow a=2$
- Thereby $\boldsymbol{t_{diff} \sim L^2/\alpha}$
- For an iron plate of thickness L=1mm the diffusion time estimate would be $t_{diff} \sim 1e\text{-}6/1e\text{-}5\ s \sim 0.1\ s$
- **Exercise:** Estimate diffusion time for iron plate of thickness $L=4mm$.

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

Insight by Fourier series:

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

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

- The exponential damping timescale is seen to be:

$$\tau_{damping} = 1/k^2 \alpha = \frac{L^2}{4\pi^2 \alpha}$$

- **Summary**: two different scalings $t_{diff} \sim L^2/\alpha$ and $t_{damping} = L^2/4\pi^2\alpha$ were obtained with similar looking characteristic timescale (with a constant pre-factor difference).
- The scaling $t_{diff} \sim L^2/\alpha$ is a famous relationship and a very useful concept in heat transfer analysis.

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, and solve the heat equation (1d) and Newton's cooling law (0d) numerically in Matlab.

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?

Solving temperature over a short time interval (timestep)

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

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

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

Discretization by explicit Euler method

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

Find new temperature

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

Solution proceeds in discrete timesteps

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

Pseudo-code

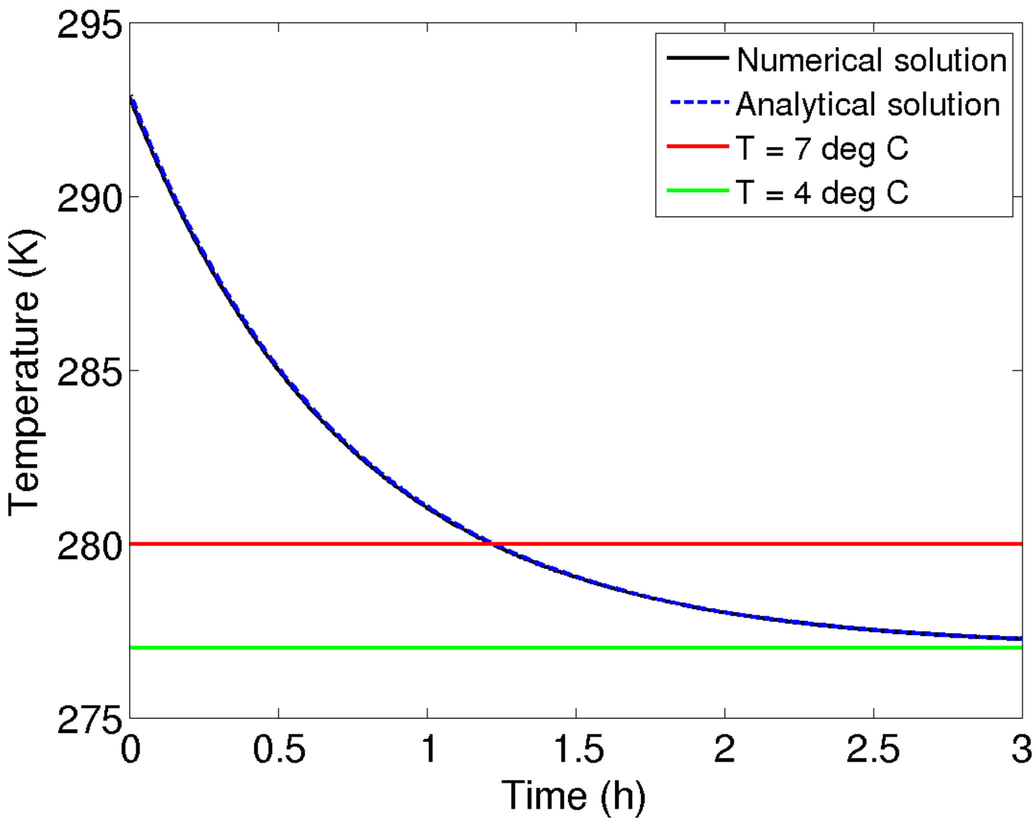
Step 0: T_o *known*

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

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

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`

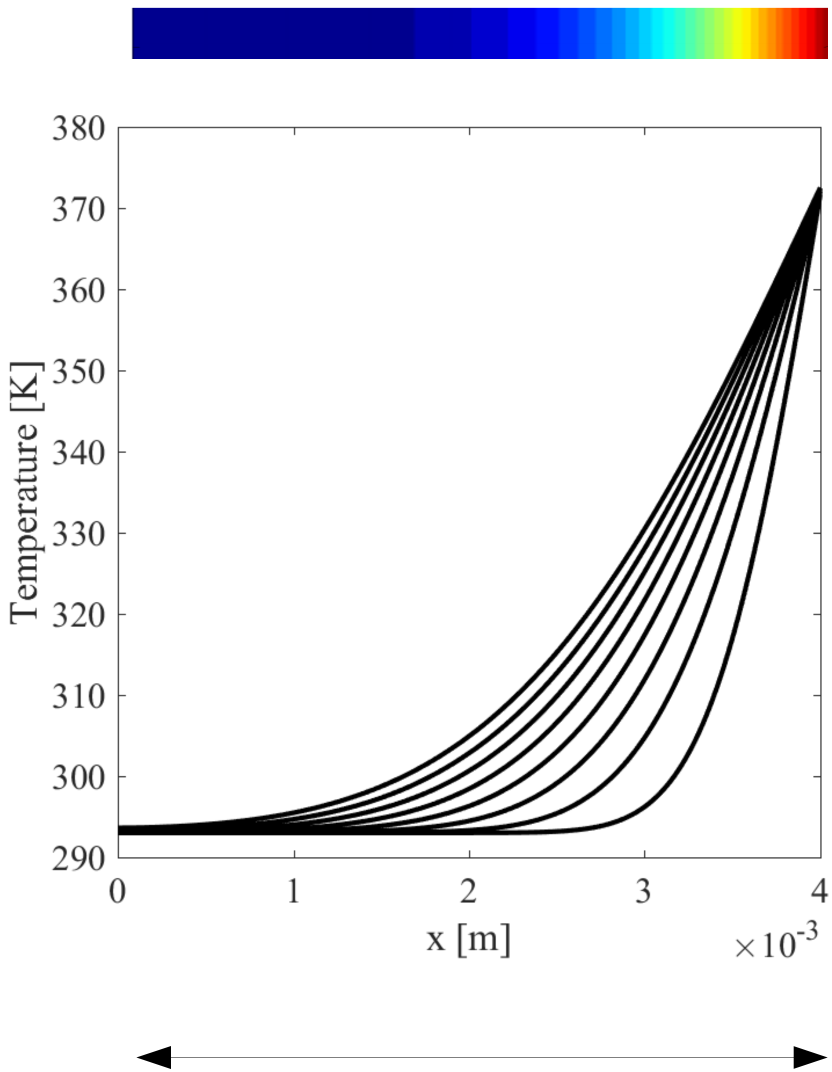
```
figure(1), clf, box, hold on
alltime = linspace(0,simutime/3600, simusteps);
plot(alltime, Tcol, 'k-','Linewidth',2)
plot(alltime, (To-Tinf)*(exp(-h*As*3600*alltime/(m*cp)))
+ Tinf, 'b--','Linewidth',2)
plot(alltime, (273+7)*(ones(length(alltime),1)), 'r-',
'Linewidth',2)
plot(alltime, (Tinf)*(ones(length(alltime),1)),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

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

"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 the discrete counterparts obtained by Taylor series expansions

Numerical solution and values stored in a table

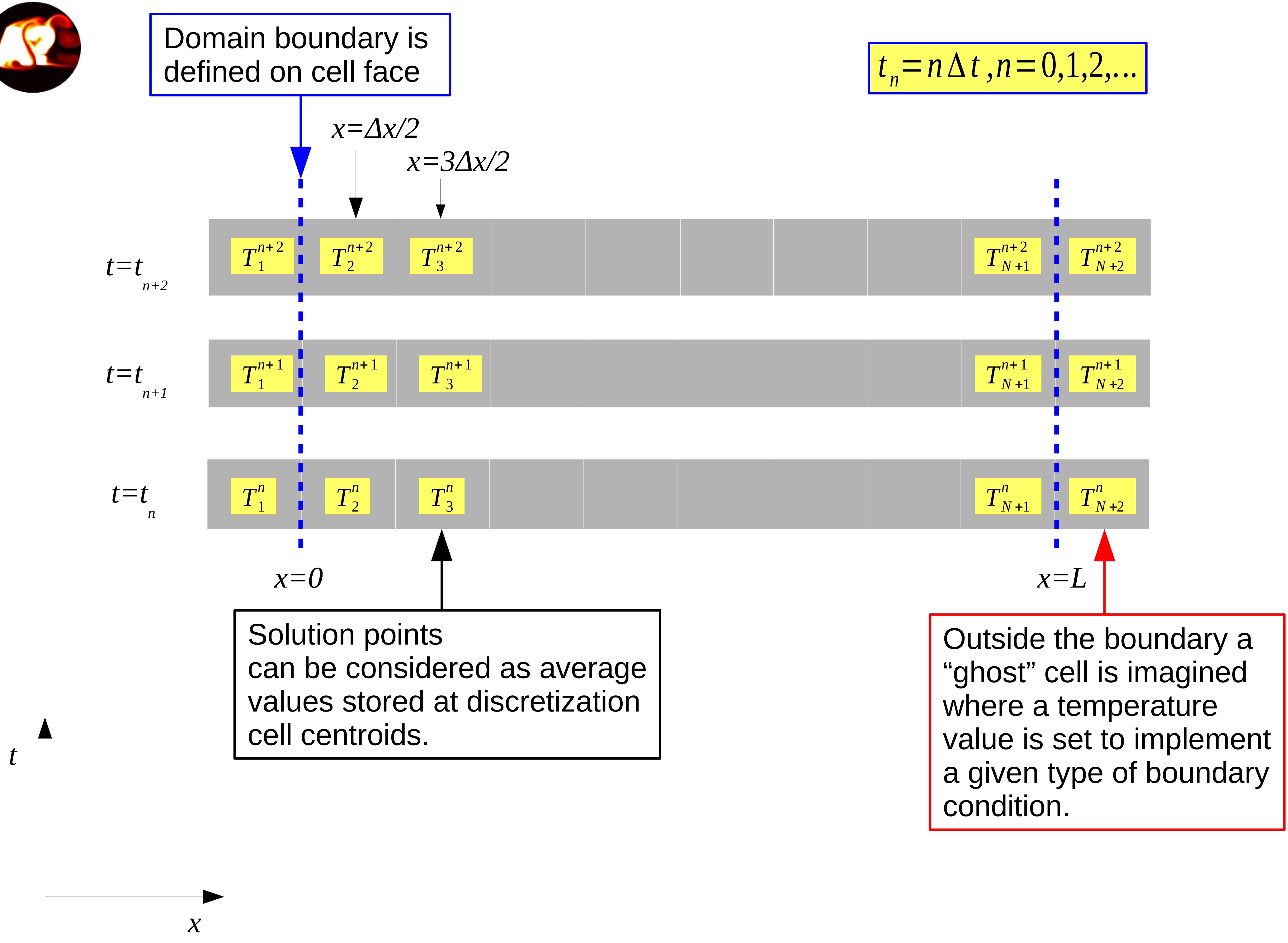


Here: 198 internal cells, 2 ghost cells

```
>> T

ans =

   293.7484
   293.7484
   293.7498
   293.7525
   293.7565
   293.7619
   293.7687
   293.7769
   293.7864
   293.7973
   293.8096
   293.8233
   293.8385
   293.8551
   293.8731
   293.8927
   293.9137
   293.9362
   293.9603
   293.9859
   294.0131
   294.0419
```

"Ghost cell" value (first value: 293.7484)

First cell inside the domain (second value: 293.7484)


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}
$x=0$
$x=L$
Solution points can be considered as average values 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

Numerical approximation of partial derivatives

- Practically all heat transfer phenomena involve PDE's
- 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.

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

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

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

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

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

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

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.

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

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

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

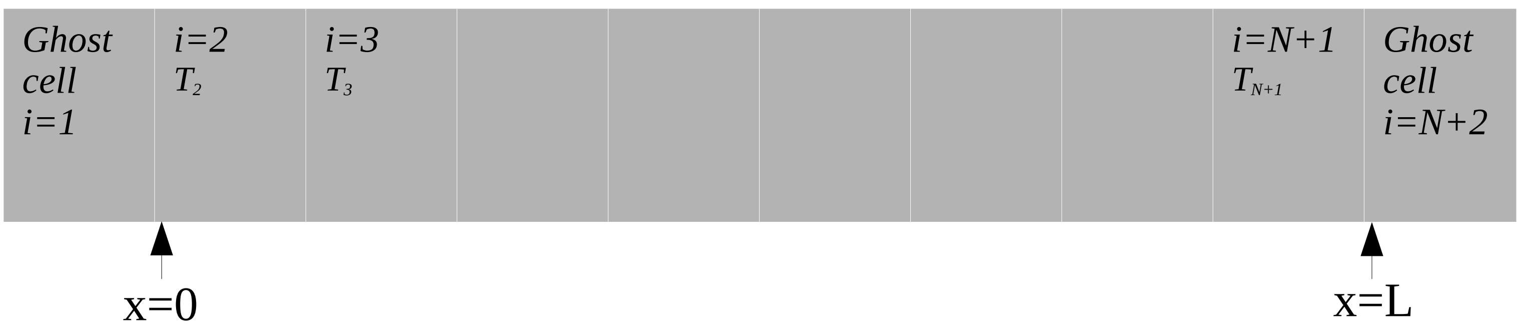
Courant-Friedrichs-Lewy number (CFL<0.5 for stability).

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

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

Boundary condition types

- **The problem:** some numerical value needs to be assigned to the **"ghost cells"**
- **Otherwise:** we can not calculate second derivative of T in cells i=2 and i=N+1
- **Case 1:** Boundary temperature fixed → boundary heat flux follows
- **Case 2:** Boundary heat flux zero (insulated) → zero temperature gradient through boundary
- **Case 3:** Boundary heat flux fixed → boundary temperature follows.



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

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

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

→ 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 cells 1 and N+2 using T from step n.

$$T_i^n$$

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

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

$$\Rightarrow \quad T_i^{n+1}$$

3) Update time according to t = t + dt

$$t_{n+1} = t_n + \Delta t$$

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}.

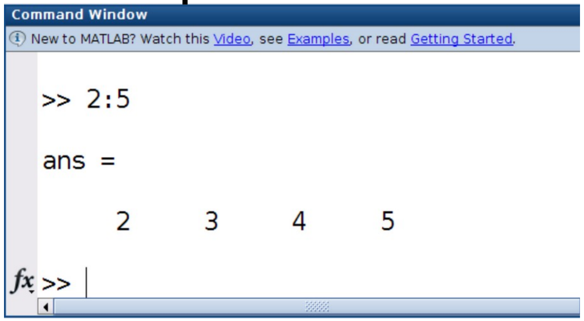
Main for-loop:

```
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.

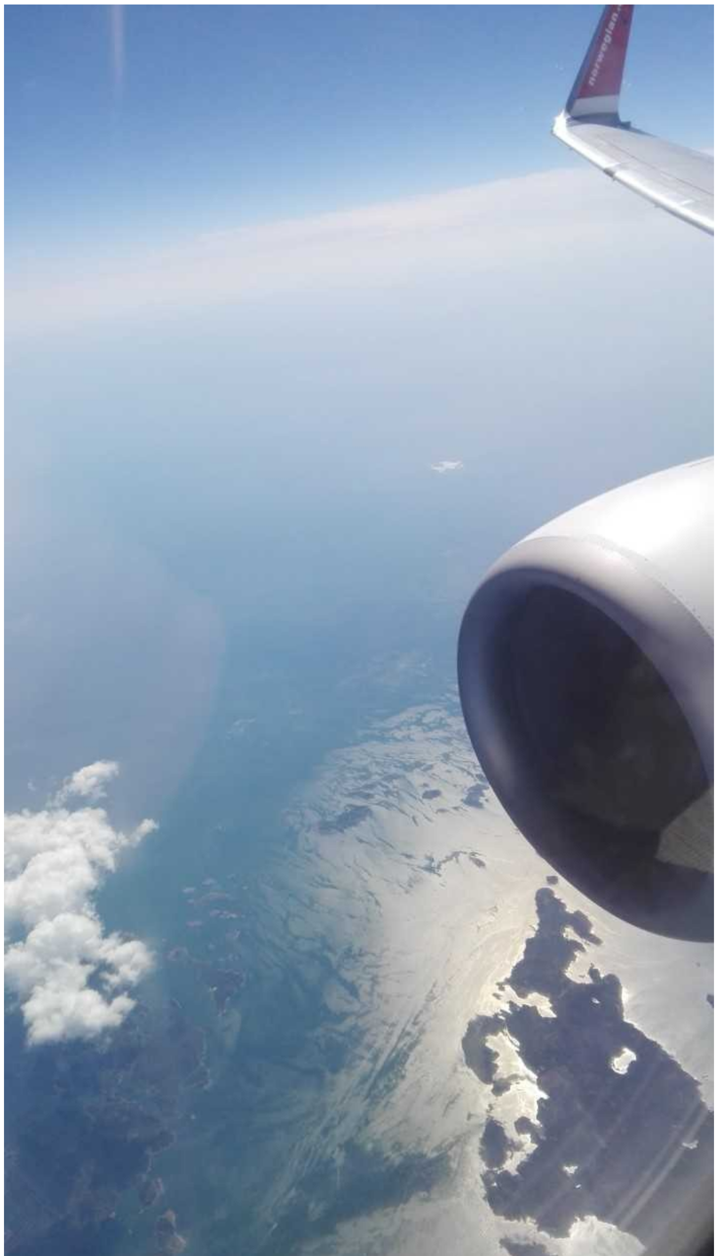
```
% define a table which refers to the 'inner points'
in = 2:(N+1);
```

Example for N+1 = 5



Supplementary material: Cooling demo from a class experiment in 2018

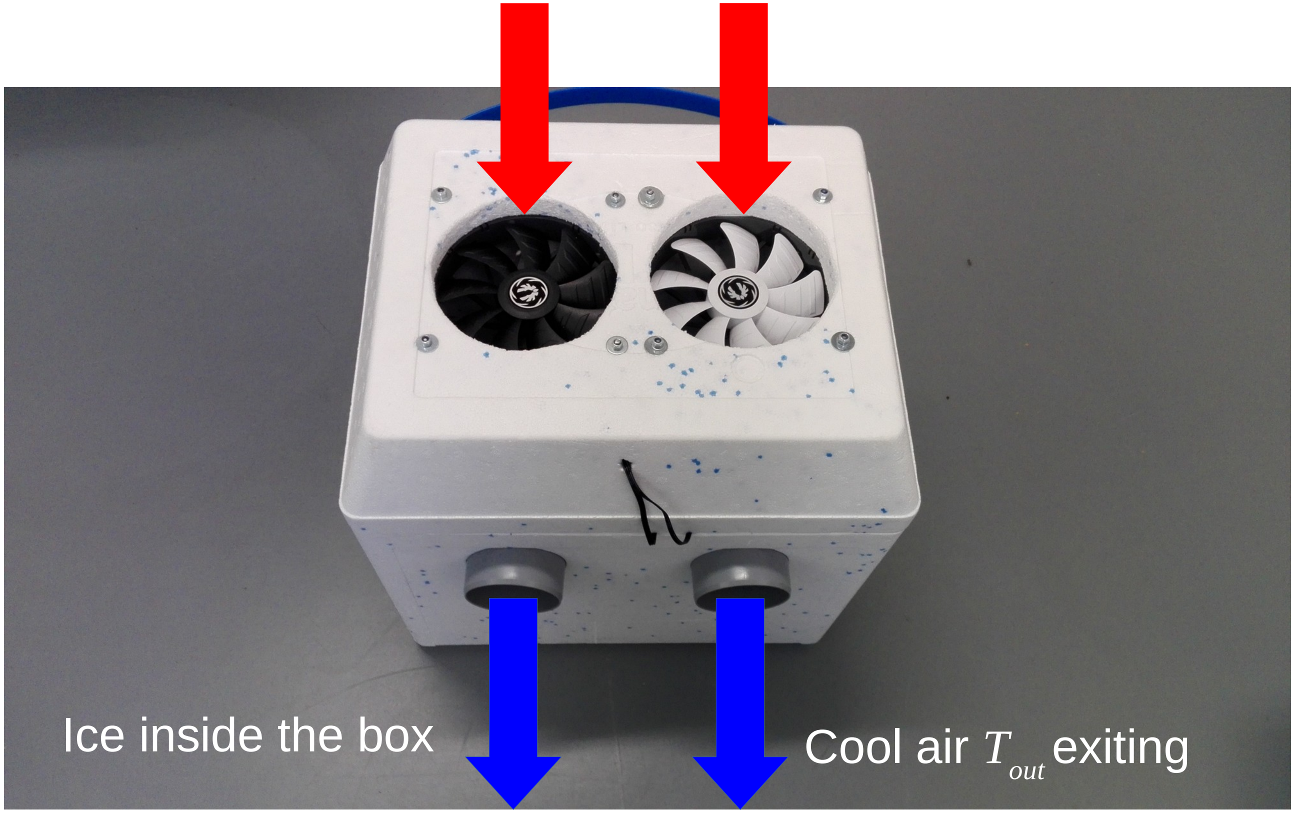
The next experimental demo (15 min) is strongly influenced by a hot hotel room



Group A: Experimenting on energy conservation analysis of homemade air-conditioner (AC)

Acknowledgement:
Mika Ahlgren and Kari Saari

Warm air T_{in} entering



Tasks:
1) Measure velocities of inflow and outflow. Is mass conservation fulfilled?
2) Measure temperature of inflow and outflow?
3) Measure the velocity also at different positions. What do you observe?

Group B: Energy conservation analysis of self-made air-conditioner

- **It is quite clear that in watts the average energy absorption rate of the ice (in Watts):**

$dE/dt = P_{in} - P_{out} = \rho U_{in} A_{in} c_p T_{in} - \rho U_{in} A_{in} c_p T_{out}$

- For incompressible fluid (air is incompressible to very high precision at low speeds < 100m/s) and thus mass conservation would imply $\rho U_{in} A_{in} = \rho U_{out} A_{out} = const.$

- However, it is not so clear where T_{out} comes from

- **Class room thinking tasks:**

1) How does dE/dt depend on:
- ice mass
- other aspects ?

2) How does T_{out} and dE/dt depend on
- inflow velocity
- inflow temperature
- other factors
- dE/dt being a constant

3) What kind of 3d features would this problem have?