TAMPERE UNIVERSITY OF TECHNOLOGY

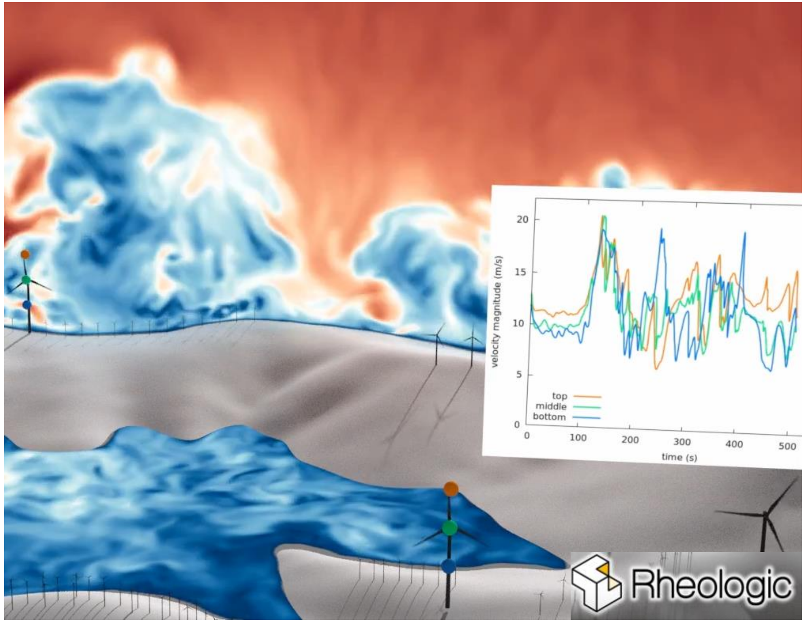
KEB-45250 Numerical Techniques for Process Modeling

Spring 2018

Antti Mikkonen
Kaj Lampio
Niko Niemelä

What is the course about?

- Industrial applications
  - Heat transfer
  - Fluid flow
  - Reacting systems
- Numerical modeling
  - Flexible
  - Custom codes
  - Software packages

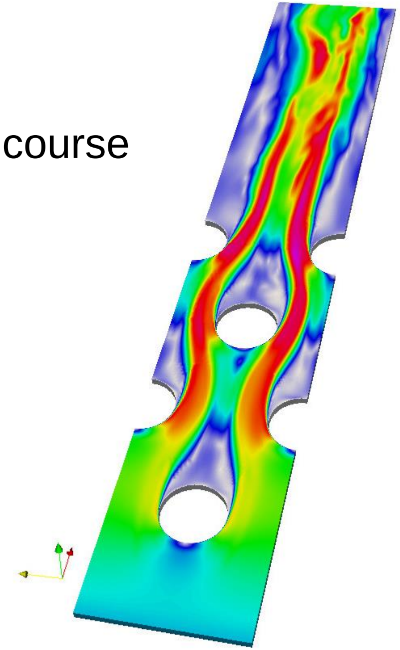


General information

- First implementation, all plans are tentative
- Lectures
  - Wednesday 13-15 K1241
- Exercises
  - Thursday 10-12 SB202 – Computer lab
- Intensive course on ANSYS Fluent
  - Tuesday 30.1. 9-16 SB202
  - Wednesday 31.1. 9-15 RG100C

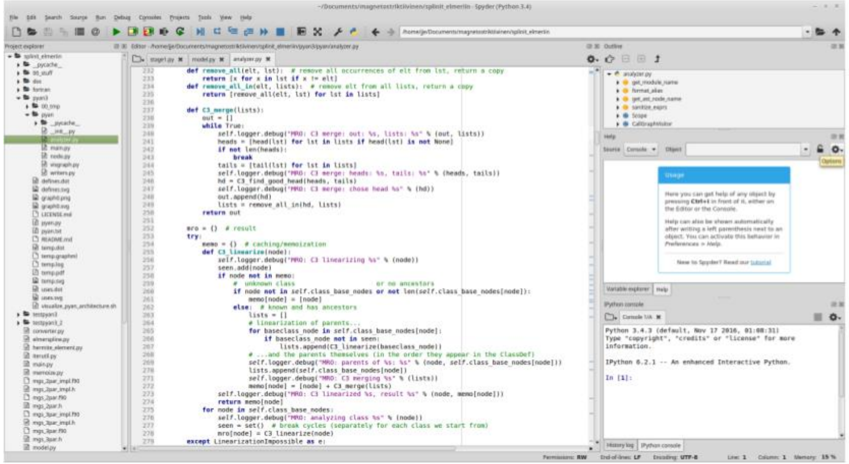
CFD software

- ANSYS Fluent
  - Computational Fluid Dynamics (CFD) software on this course
  - Easy to learn
  - Commercial and expensive
  - Intensive course
    - Tuesday 30.1 9-16 SB202
    - Wednesday 31.1. 9-15 RG100C
- OpenFOAM
  - More popular at TUT
  - Slow to learn
  - Free and open source



Programming language

- Python 3.6
  - Most familiar with
  - One of the most popular languages in the world
  - Extensive liberties for engineer
- If you want to use something else, for example Matlab, just ask



https://github.com/spyder-ide

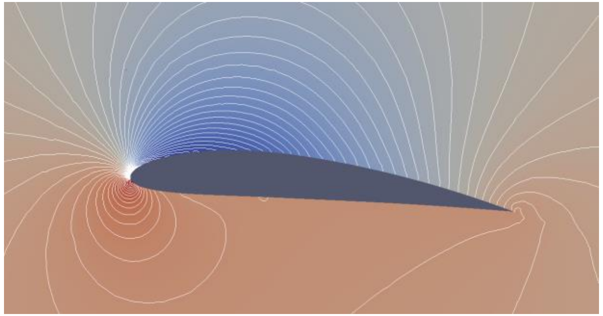
Mandatory steps to pass

- Exam
  - 60% of total points
  - Must be passed
- 2 assignments with reports
  - 40% of total points
  - Must be passed

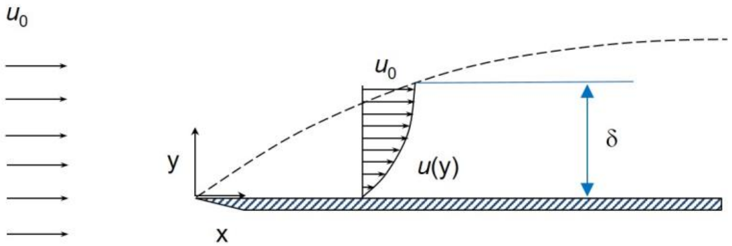
Exam, 60%

- 11.05.2018, time: 17-20
- Preliminary plan
  - 5 questions
  - 2.5 about Computational Fluid Dynamics
  - 1.5 about numerical modeling in general
  - 1 about reacting systems

Assignment 1, 25%

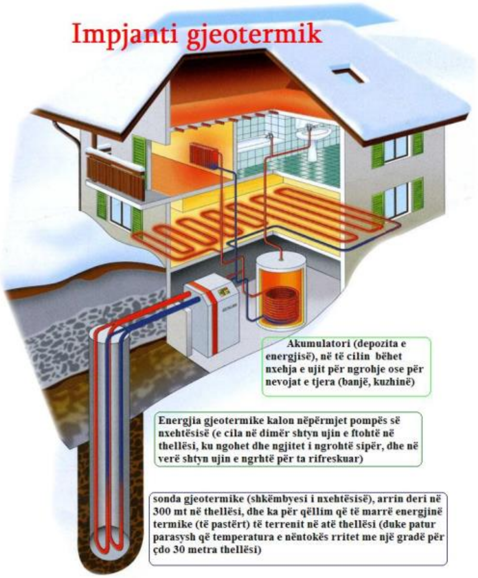


- Two options
  - Calculate a 2D case with CFD software
    - Probably wing
    - Ansys Fluent/OpenFOAM
    - Industry oriented
  - Own code
    - Probably flat plate
    - Python/other language
    - Deeper understanding
  - You can also do both
    - 5-10% extra



Assignment 2, 15%

- Custom code (Python)
  - Custom code is better for some applications
- Maybe geothermal energy



https://commons.wikimedia.org/wiki/File:Impjanti_Gjeotermik.jpg

Course material

- CFD
  - There will be lecture notes
  - Additionally:
    - H. Versteeg & W. Malalasekera:

      "An introduction to computational fluid dynamics: The finite-volume method", 2nd ed.
- Lecture slides for the rest

Tentative plan

			Lectures		Exercises
Month	Week	Day		Day	
1	1	3		4	
	2	10	Introduction	11	Python basics and libraries
	3	17	Basics. Matrix, NS,…	18	Lecture material
	4	24	CFD Basics	25	Lecture material
2	5	30	ANSYS intensive course	31	ANSYS intensive course
	6	7	Heat convection, FVM	8	Lecture material with Python
	7	14	Advection	15	Lecture material with Python
	8	21	Navier-Stokes	22	Navier-Stokes with ANSYS
	9	28	Mesh	1	Mesh with ANSYS
3	10	7	Turbulence	8	Turbulence with ANSYS
	11	14	Linear systems	15	Lecture material
	12	21	Linear systems	22	Lecture material
	13	28	Easter Holiday	29	Easter Holiday
4	14	4	Non-linear systems	5	Lecture material
	15	11	Non-linear systems	12	Lecture material
	16	18	Reacting systems	19	Lecture material
	17	25	Reacting systems	26	Lecture material

Example Cases

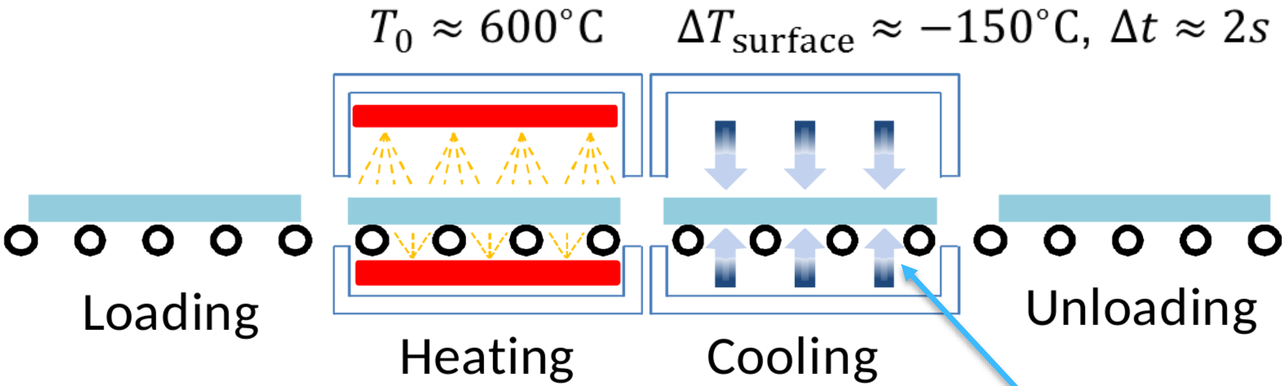
- Glass tempering
  - Antti Mikkonen
- Fin optimization (separate slides)
  - Kaj Lampio
- Combustion modelling (separate slides)
  - Niko Niemelä

Tempered Glass

Safety Glass Door
by Wei Min Chan,
https://www.youtube.com/watch?v=aQ902DfWILs

John Hancock Panorama, Chicago
by RhythmicQuietude, CC BY-SA 3.0
https://commons.wikimedia.org/w/index.php?curid=10589310

Production



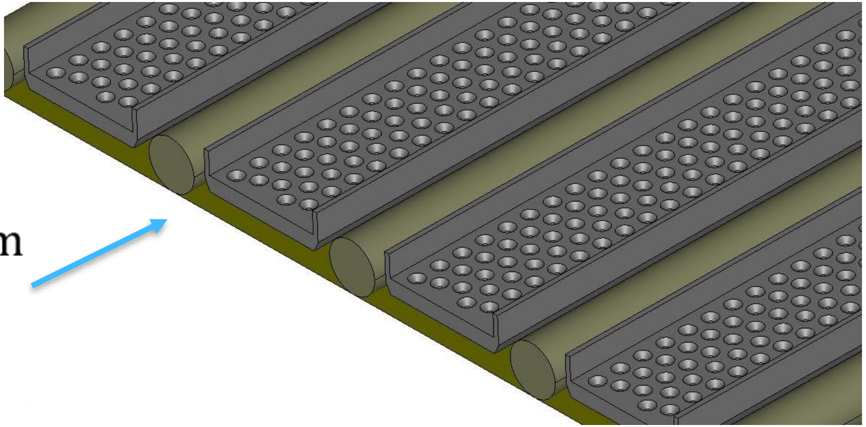
Aronen 2012 [1]

Cooling Jets

$\text{Ma} \approx 0.85$

$d_{\text{nozzle}} \approx 1 - 3\ \text{mm}$

$\bar{h} \approx 1000\ \text{W/m}^2\text{K}$

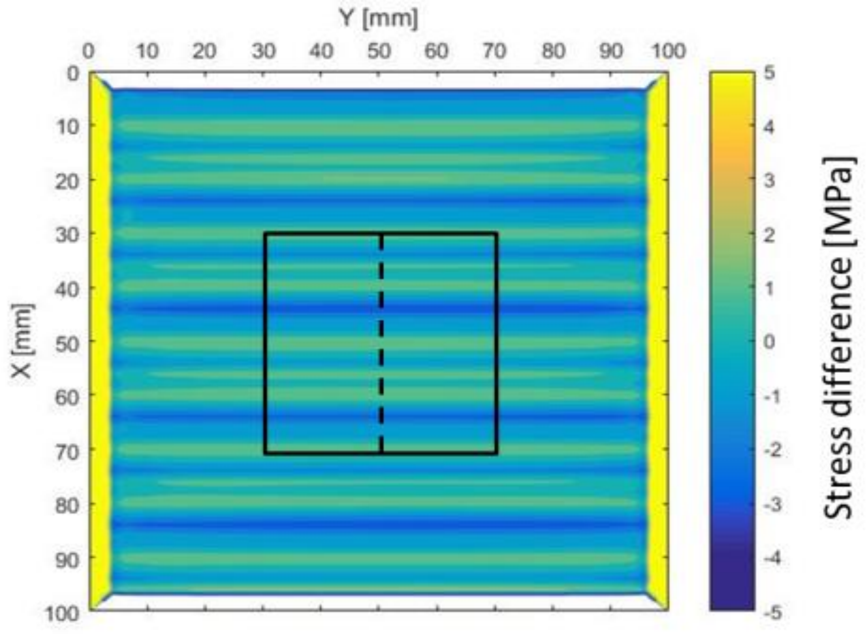


Visual issues

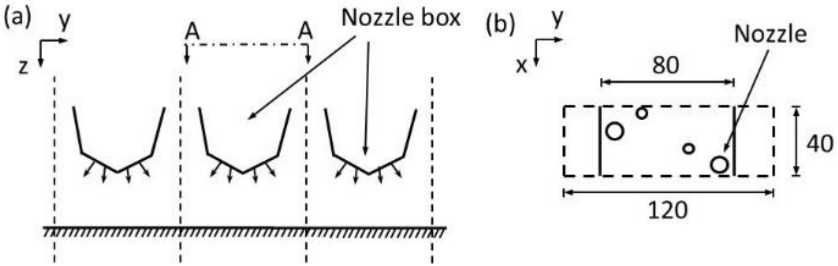
Figure 1.3. Roller waves (Henriksen & Leosson 2009).

Figure 1.4. Longitudinal patterns (Henriksen & Leosson 2009).

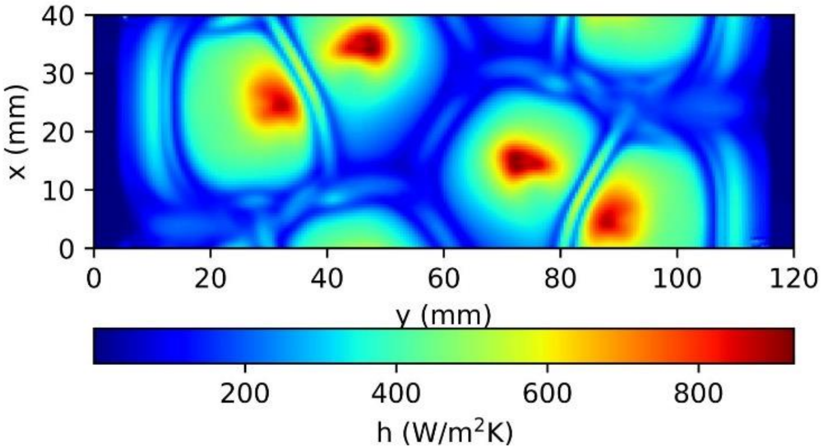
Residual stress



Heat transfer modeling

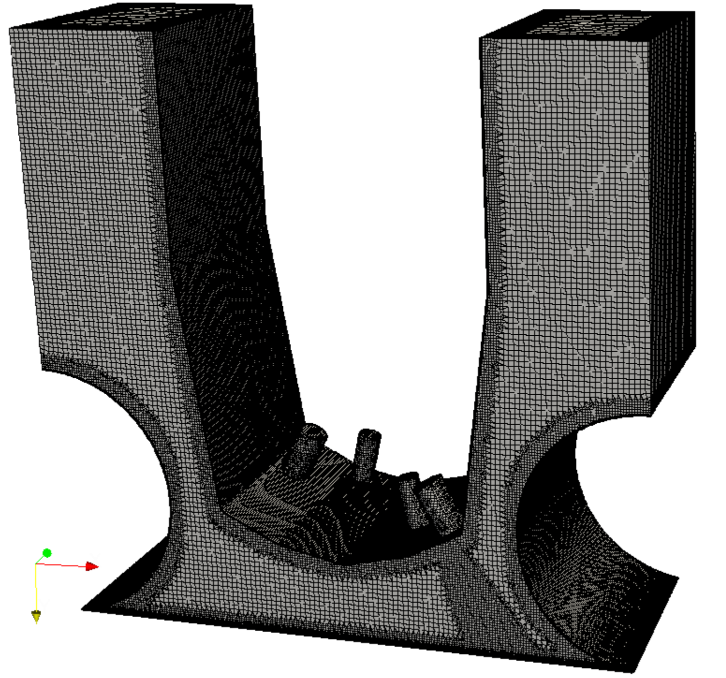


Schematic of the nozzles (a) and locations in nozzle plate (b).

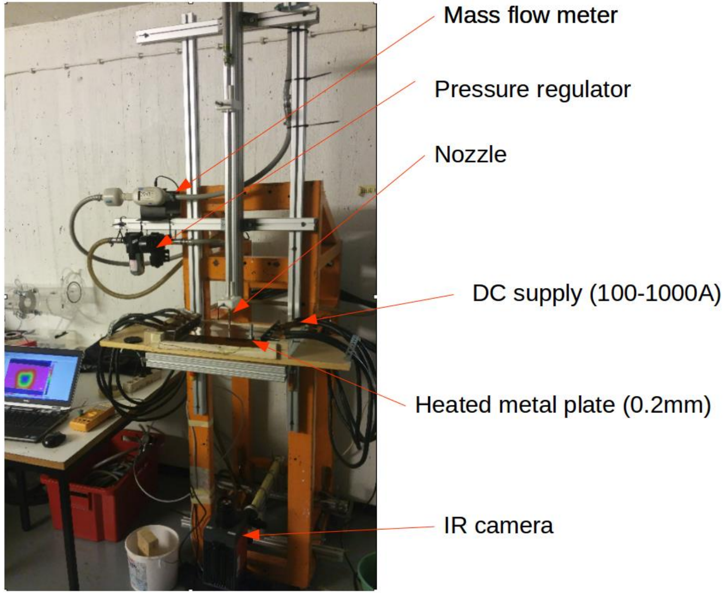


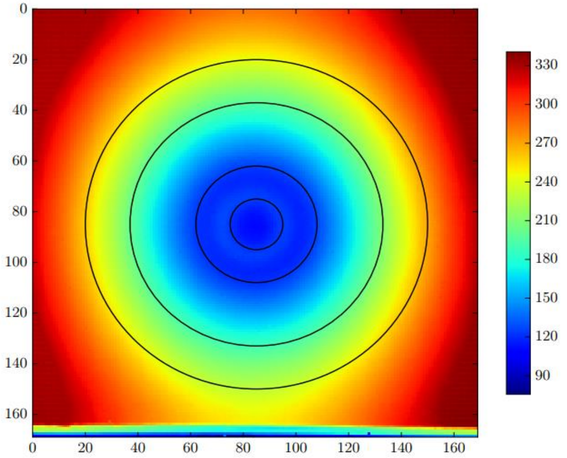
Distribution of heat transfer coefficient

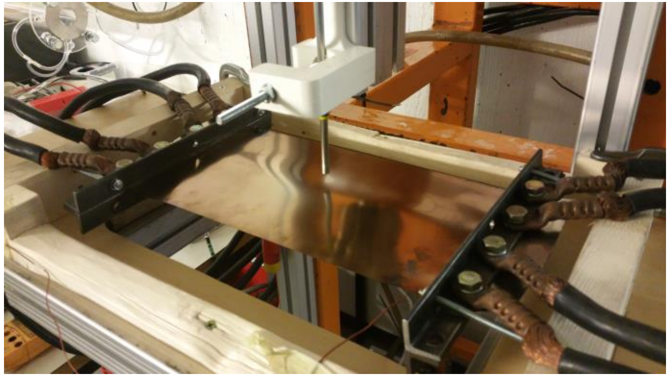
Mesh



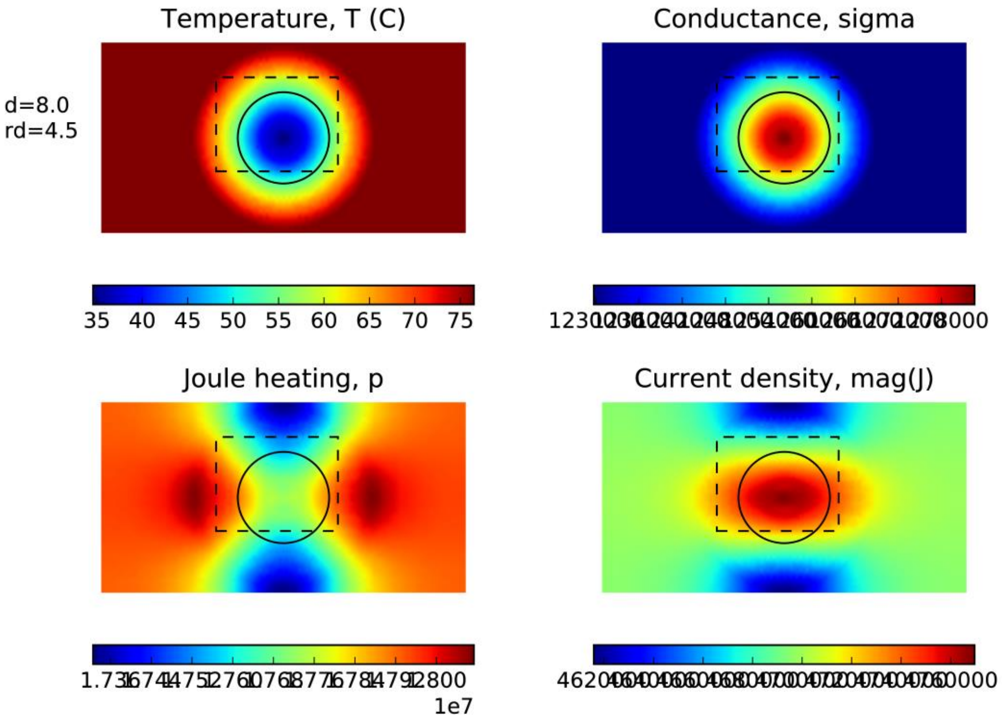
Measurements



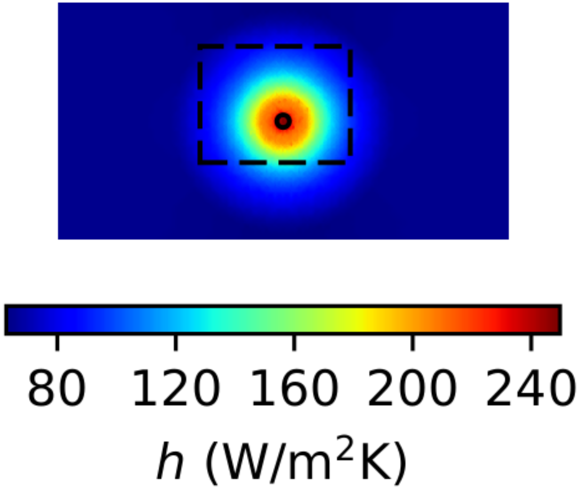


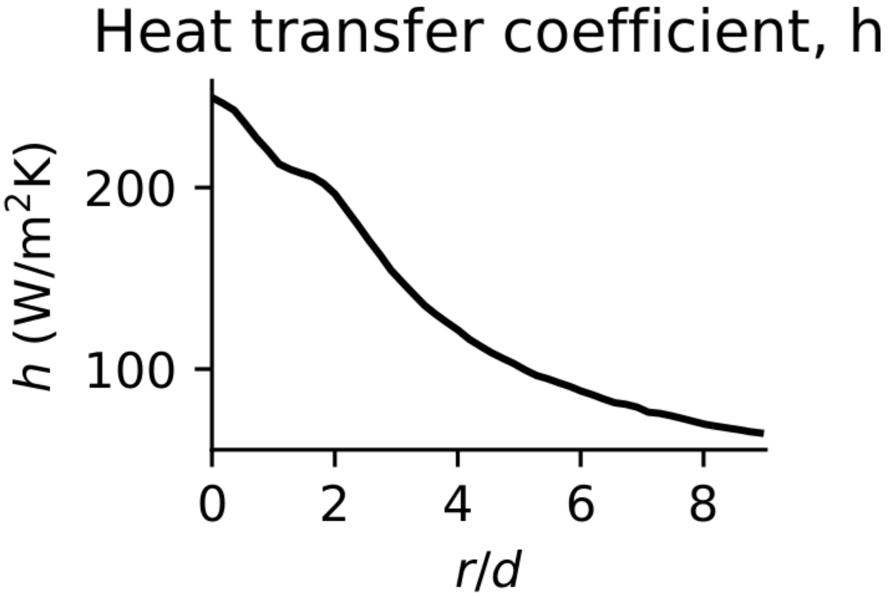


Joule heating



Heat transfer





OpenFOAM, Open Source

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial\left(\rho U_j k\right)}{\partial x_j} = \tilde{P} - \beta^* \rho k \omega + \frac{\partial}{\partial x_i}\left[(\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_i}\right]$$

```
    // Turbulent kinetic energy equation
    tmp<fvScalarMatrix> kEqn
    (
        fvm::ddt(alpha, rho, k_)
      + fvm::div(alphaRhoPhi, k_)
      - fvm::laplacian(alpha*rho*DkEff(F1), k_)
     ==
        min(alpha*rho*G, (c1_*betaStar_)*alpha*rho*k_*omega_)
      - fvm::SuSp((2.0/3.0)*alpha*rho*divU, k_)
      - fvm::Sp(alpha*rho*betaStar_*omega_, k_)
      + kSource()
      + fvOptions(alpha, rho, k_)
    );
```

Full address:
https://github.com/OpenFOAM/OpenFOAM-dev/blob/master/src/TurbulenceModels/turbulenceModels/RAS/kOmegaSST/kOmegaSST.C

Electric heating

$$\nabla \cdot \sigma \nabla \phi = 0$$

$$\boldsymbol{E} = -\nabla \phi$$

$$\boldsymbol{J} = \sigma \boldsymbol{E}$$

$$p = \frac{dP}{dV} = \boldsymbol{J} \cdot \boldsymbol{E} = \boldsymbol{J} \cdot \boldsymbol{J} / \boldsymbol{\sigma} = \frac{|\boldsymbol{J}|^2}{\boldsymbol{\sigma}}$$

```
# Define variational problem
    u = TrialFunction(V)
    v = TestFunction(V)
    F = sigma*dot(grad(u), grad(v))*dx
    a, L = lhs(F), rhs(F)
    u = Function(V)
    solve(a == L, u, [bcL, bcR])

    J = -sigma*grad(u)
    p = project(dot(J,J)/sigma, V)
```