Finite Element Modelling Of Thermal Processes With Phase Transitions

Abdelhaq Abouhafç

Delft University of Technology
TNO Science and Industry

June 14, 2007

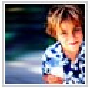
TNO: Knowledge for business

- a research organization
- different locations
- many experts work together in teams

TNO: Knowledge for business

Defence Industry

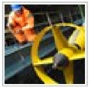
Environment, Health and Safety



Food and Nutrition

High-end Equipment

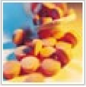
ICT



Maritime

Medical

Oil and Gas



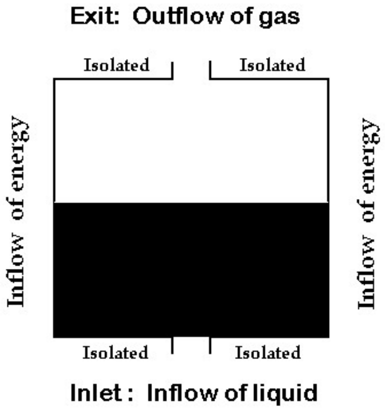
Pharma

Prevention and Care

Process Industry

Security and Safety

Introduction: Boiler System



Aim and Difficulties

Aim:

Finite Element modelling of thermal processes with phase transitions using density-enthalpy phase diagrams

Difficulties:

- Nonlinearity
- Time dependent problem
- Coupled equations
- High convection: Numerical oscillations
- Accuracy and stability

Outline

Classical methods: Disadvantages

- For each phase: a set of equations
- A lot of coefficients and parameters
- Discontinuity across the interfaces
- A lot of assumptions: Less accurate

Density Enthalpy method: Advantages

- One set of equations
- Less input parameters
- Less assumptions
- Accuracy
- Stability

Density-Enthalpy phase diagrams: $T = T(\rho, h), P = P(\rho, h)$.

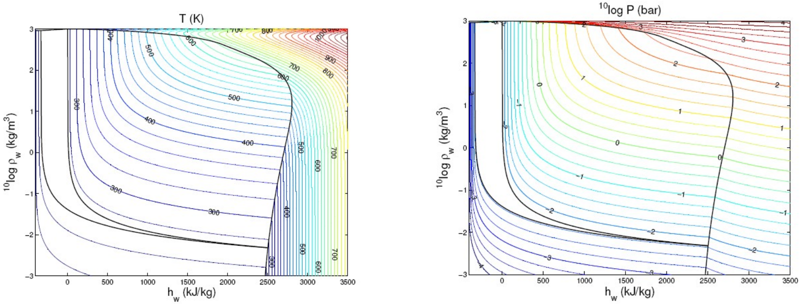


Figure: Temperature and Pressure as functions of density and enthalpy for pure water

Density-Enthalpy phase diagram: $X^G = X^G(\rho, h)$.

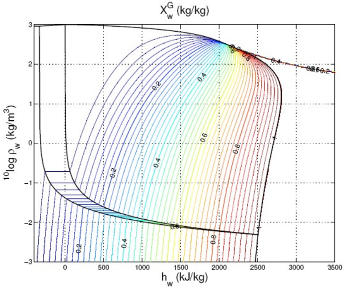


Figure: Gas Mass fraction as function of density and enthalpy for pure water

Mass and Energy balances

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= -\vec{\nabla}.(\rho \vec{v}) \\ \frac{\partial (\rho h)}{\partial t} &= -\vec{\nabla}.(\rho h \vec{v}) + \vec{\nabla}.\left(\lambda \vec{\nabla} T\right) + q \end{aligned}$$

ρ	density $[Kg/m^3]$
h	enthalpy $[Kg/m^3]$
T	temperature $[K]$
$\vec{v}$	velocity $[m/s]$
λ	heat conduction $[W/m/K]$
q	heat source $[W/m^3]$

Darcy velocity

$$\vec{v} = -\frac{K}{\mu}\vec{\nabla}P$$

$\vec{v}$	velocity $[m/s]$
K	permeability $[m^2]$
μ	dynamic viscosity $[Pa.s]$
P	pressure $[Pa]$

Boundary Conditions

The external mass transfer:

$$\rho\vec{v}.\vec{n} = k_m\left(\rho - \rho_a\right)$$

k_m mass transfer coefficient $[m/s]$
ρ_a ambient density $[Kg/m^3]$

Boundary Conditions

External Energy transfer by convection:

- if $\rho - \rho_a > 0$

$$(\rho h)\,\vec{v}.\vec{n} = h|_\Gamma k_m\,(\rho - \rho_a)$$

- if $\rho - \rho_a < 0$

$$(\rho h)\,\vec{v}.\vec{n} = h_a k_m\,(\rho - \rho_a)$$

External energy transfer by conduction:

$$\lambda\vec{\nabla}T.\vec{n} = k_h\,(T - T_a)$$

k_h heat transfer coefficient $[W/m^2/K]$
h_a ambient enthalpy $[J/Kg]$

Initial Conditions

$$
\begin{aligned}
\rho(t_0, x) &= \rho_0, \quad x \in \Omega, \\
h(t_0, x) &= h_0, \quad x \in \Omega.
\end{aligned}
$$

Ω domain
t_0 starting time $[s]$

Numerical Approach: transform PDEs to ODEs

- Weak formulation:

$$\int_\Omega \frac{\partial \rho}{\partial t}\eta dV = -\int_\Omega \vec{\nabla}.\left(\rho\vec{v}\right)\eta dV$$

- Integration by parts:

$$\int_\Omega \frac{\partial \rho}{\partial t}\eta dV = -\int_\Gamma \rho\eta\vec{v}.\vec{n}dS + \int_\Omega \rho\vec{v}.\vec{\nabla}\eta dV$$

Ω	domain
Γ	boundary
$\vec{n}$	outward unit normal to $\Gamma = \partial\Omega$
η	test function

Numerical Approach: transform PDEs to ODEs

- Use the boundary conditions:

$$\int_{\Omega} \frac{\partial \rho}{\partial t} \eta dV = \int_{\Gamma} \eta k_m (\rho_a - \rho) \, dS - \int_{\Omega} \rho \frac{K}{\mu} \vec{\nabla} P . \vec{\nabla} \eta dV$$

- The solution $\rho(\vec{x}, t)$ is approximated by:

$$\rho(\vec{x}, t) = \sum_{j=0}^{N} \rho_j(t) \varphi_j(\vec{x})$$

ρ_j	unknown density
φ_j	basis function
$N + 1$	number of mesh nodes for the unknowns

Numerical Approach: transform PDEs to ODEs

- Galerkin approximation:

$$\sum_{j=0}^{N} \frac{d\rho_j(t)}{dt} \left(\int_{\Omega} \varphi_i \varphi_j dV \right) =$$

$$\int_{\Gamma} \varphi_i k_m \left(\rho_a - \sum_{j=0}^{N} \rho_j(t) \varphi_j \right) dS -$$

$$\sum_{j=0}^{N} \left(\int_{\Omega} \frac{K}{\mu} \vec{\nabla} P . \vec{\nabla} \varphi_i \varphi_j dV \right) \rho_j(t)$$

Numerical Approach: transform PDEs to ODEs

- System of nonlinear ODEs:

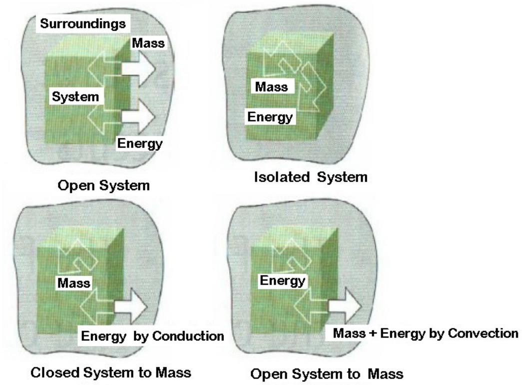
$$M\frac{d\vec{\rho}}{dt} = S(\vec{\rho}, \vec{h})\vec{\rho} + \vec{F}$$

$\vec{\rho}$	unknown density
M	mass Matrix
S	stiffness Matrix
$\vec{F}$	vector

Numerical Tools

- Use SUPG method to reduce the effect of high convection,
- Use Implicit backward Euler scheme to guarantee stability.

Types of simulated systems



Open System to Mass, $k_m = 2, k_h = 0, dt = 0.001$

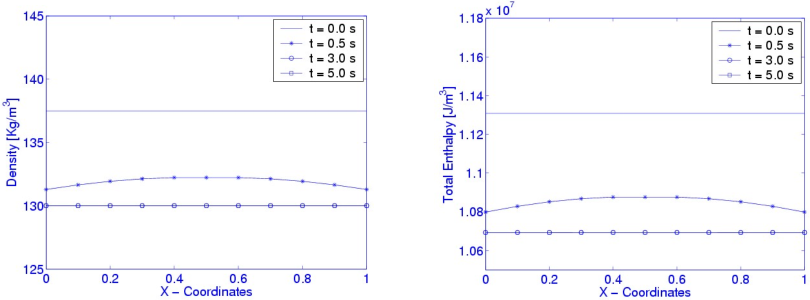
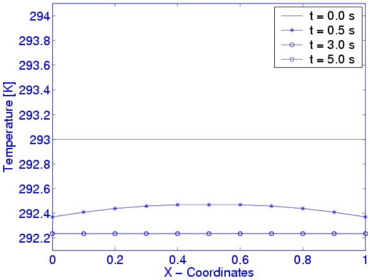


Figure: Density and Total Enthalpy after 5000 time steps

Open System to Mass, $k_m = 2, k_h = 0, dt = 0.001$



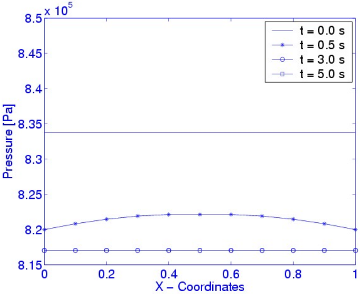


Figure: Temperature and Pressure after 5000 time steps

Open System to Mass, $k_m = 2, k_h = 0, dt = 0.001$

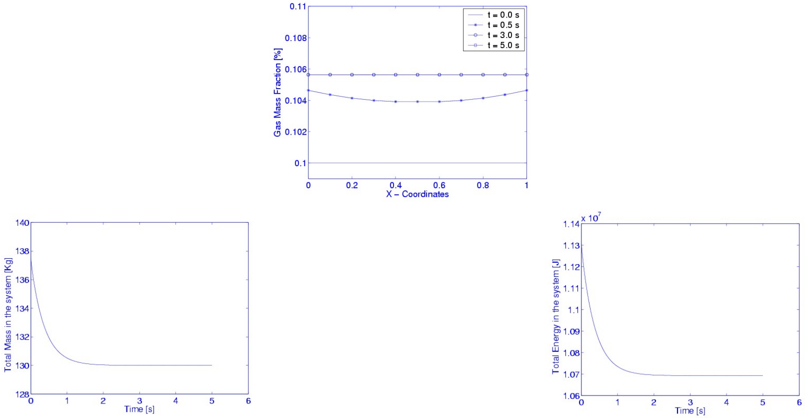
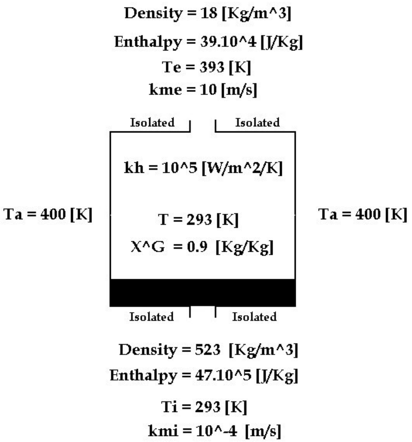


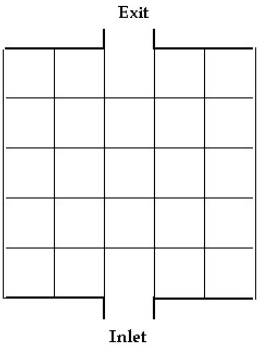
Figure: Gas Mass Fraction, Total Mass and Energy after 5000 time steps

Thermodynamical results

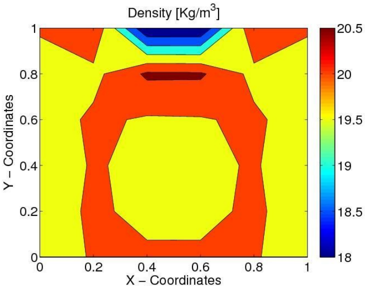
- Density inside an open system to mass, tends to the ambient density when the system reaches its steady state,
- The external transfer process of mass and energy stops as soon as the system reaches its steady state at thermodynamical equilibrium.

Geometry and Operating Conditions





2D Boiler: Mass is flowing out and Energy is flowing in



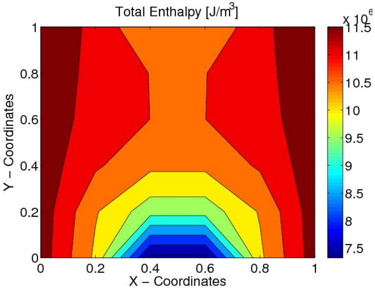


Figure: Density and Total Enthalpy with respect to x and y after 200000 time steps, dt $= 0.001$

2D Boiler: Mass is flowing out and Energy is flowing in

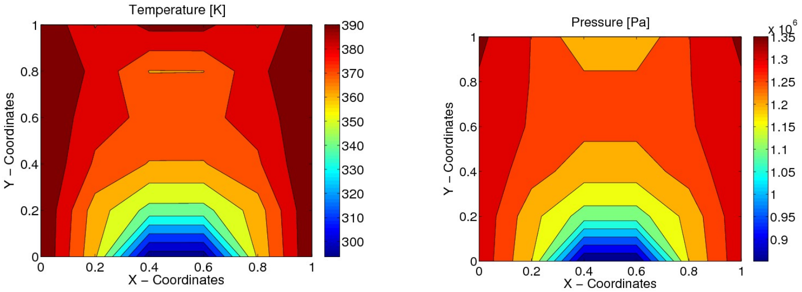


Figure: Temperature and pressure with respect to x and y after 200000 time steps, dt $= 0.001$

2D Boiler: Mass is flowing out and Energy is flowing in

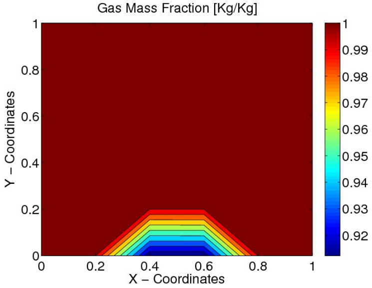


Figure: Gas Mass Fraction with respect to x and y after 200000 time steps, dt $= 0.001$

Conclusions

1. Set up a stable and accurate FEM model using Density-Enthalpy diagrams (0D, 1D and 2D with Matlab),
2. Numerical results are logical and have obtained for problems where classical methods may not work.

Recommendations

1. Transfer model to SEPRAN,
2. Study more aspects of the model (Compare implicit with explicit),
3. Exploit the sparsity of the matrix: essential for saving memory and computing time,
4. Apply iterative solvers to speed up calculations,
5. Validation of the numerical model: Compare with experimental results,
6. Add the gravity force to the equations.

Questions