COMSOL Implementation of a Porous Media Model for Simulating Pressure Development in Heated Concrete

B. Weber¹, D. Dauti², and S. Dal Pont²

¹Empa, Swiss Federal Laboratories for Material Science and Technology, Dübendorf
²Laboratoire 3SR, University Grenoble Alpes, Grenoble, France
Motivation: concrete spalling

- When exposed to high temperatures (fire), concrete can spall.
- Spalling: violent detachment of flakes
- Severe damage in tunnels after fire
- Presumable causes:
  - Pore pressure
  - Thermal stresses

→ Simulate pore pressure development
Thermo-hydro model

- Porous medium
- Gas (air+vapor) and liquid phase, partially saturated
- Heat transfer
- Moisture transfer

- Implemented with weak form interface (no physics interfaces)
  - Comparison with other implementations
  - Explore different formulations
Conservation equations

- General form of conservation equation

\[
\frac{\partial}{\partial t} \text{density} + \nabla \cdot \text{flux} = \text{source}
\]

- Air

\[
\frac{\partial}{\partial t} (\phi(1 - S) \rho_a) + \nabla \cdot (\rho_a \mathbf{v}_a) = 0
\]

- Water species (vapor+liquid water, evaporation does not appear)

\[
\frac{\partial}{\partial t} \phi \left( S \rho_i + (1 - S) \rho_v \right) + \nabla \cdot \left( \rho_v \mathbf{v}_v + \rho_i \mathbf{v}_i \right) = \dot{m}_{\text{dehyd}}
\]

- Energy

\[
\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot \left( -k_{\text{eff}} \nabla T \right) + \left( \rho_v c_p \mathbf{v}_v + \rho_a c_p \mathbf{v}_a \right) \cdot \nabla T = -\dot{m}_{\text{dehyd}} \Delta h_{\text{dehyd}} - \dot{m}_{\text{evap}} \Delta h_{\text{evap}}
\]
Variables

- Temperature
- Mass densities (air, vapor, liquid)
- Mass flux (air, vapor, liquid)
- Saturation (liquid, gas)

- Three equations → 3 dependent variables
  - Temperature
  - Gas pressure
  - Variable for water content
    - Capillary pressure (used here)
    - Vapor pressure (used as alternative)
    - Saturation

- Need constitutive equations
Constitutive equations: mass density and flux

- Ideal gas law
  \[ \rho_a = \frac{p_a M_a}{RT} \]
  \[ \rho_v = \frac{p_v M_v}{RT} \]

- Dalton’s law
  \[ p_g = p_a + p_v \]

- Darcy’s law
  \[ \mathbf{v}_g = -\frac{\kappa_{rg} K}{\mu_g} \nabla p_g \]
  \[ \mathbf{v}_l = -\frac{\kappa_{rl} K}{\mu_l} \nabla p_l \]

- Fick’s law

- Incompressible water
  \[ \rho_l(T) \]
Capillary effects: vapor-liquid equilibrium

- Surface tension in interface (meniscus)
  \[ h = \frac{p_c}{\rho_l g} \]
  \[ p_c = \frac{2\sigma}{r} \]

- Surface tension gives rise to pressure difference between gas and liquid water = capillary pressure
  \[ p_c = p_g - p_l \]

- Vapor-liquid equilibrium (Kelvin equation)
  \[ p_v = p_{\text{sat}}(T) \cdot \exp \left( -\frac{p_c M_w}{\rho_l RT} \right) \]
Capillary effects: Saturation

- Saturation $S(p_c)$ where $p_c = p_g - p_l$ (global)
- Relative height in each tube depends only on radius
- If global water pressure (reservoir level) is increased
  - Global $p_c$ decreases
  - Saturation increases
Properties related to capillary pressure

- Saturation (van Genuchten)
  - Vapor pressure (Kelvin)
Limitations of model

- Density of liquid water

- Density is used in Kelvin’s equation

\[ p_v = p_{sat}(T) \cdot \exp \left( \frac{-p_c M_w}{\rho_l RT} \right) \]

- Liquid water density undefined above critical temperature (374°C)

- As a workaround, we kept the water density constant beyond the critical density.
Example: weak form for air conservation

\[
\frac{\partial}{\partial t} \left( \phi(1 - S) \rho_a \right) + \nabla \cdot \left( \rho_a \mathbf{v}_a \right) = 0
\]

PDE for air

\[
0 = -\int_{\Omega} \tilde{p}_g \frac{\partial}{\partial t} \left( \phi(1 - S) \rho_a \right) d\Omega
\]

 Weak form

\[
+ \int_{\Omega} \nabla \tilde{p}_g \cdot \left( \rho_a \mathbf{v}_a \right) d\Omega
\]

\[
+ \int_{\Sigma} \tilde{p}_g \left( \rho_a \mathbf{v}_a \right) \cdot \mathbf{n} d\Sigma
\]

Boundary condition

\[
\text{test}(pg)*d(phi*(1-S)*rho_a,t)
\]

\[
\text{test}(pgx)*\text{flux}_ax+\text{test}(pgy)*\text{flux}_ay (2D)
\]
Automatic recursive variable substitution

\[ \frac{\partial}{\partial t} \left( \phi(1 - S) \rho_a \right) \]

\[ d(\phi*(1-S)*\rho_a,t) \]

\[ S(pc,T) \quad \rho_a = \frac{Ma}{(R_{\text{const}}*T)}*pa \quad \text{Ideal gas} \]

\[ \phi = \phi_0 + A\phi*(T-T_{\text{amb}}) \]

\[ \rho_l = c_1 + c_2*T + c_3*T^2 + ... \]

\[ \text{Derivative with resp. to } t \]

\[ \text{pa} = pg - pv \quad \text{Dalton in this form!} \]

\[ pv = psat(T)*\exp(-pc/\rho_l*Mw/R_{\text{const}}/T) \quad \text{Kelvin} \]

- Dependent variables \( T, pg, pc \) on the right-hand side
- Avoid circular variable definitions
Experiment form literature

- 12-cm-thick concrete slab 30 x 30 cm²
- Pressure and temperature sensors
- Heated with radiator from top during several hours
- Not all material parameters are provided in the paper: others from literature or by calibration.

Comparison with experiment

Temperature

Gas pressure
Material properties and calibration

- Temperature (fitting)
  - Radiator temperature
  - Heat transfer coefficients
- Permeability (main tuning parameter)
  - Nominal value (tuning)
  - Evolution with temperature (cracking)
- Saturation law (main uncertainty)
  - Standard curve at room temperature (van Genuchten)
  - Temperature dependence: no data >100°C
Comparison with other implementations

- Similar computing time with fixed time step.
- COMSOL with variable time step 6-7 times faster, but curves not as smooth.
- COMSOL with pv instead of pc does not significantly change computing time.
Summary and conclusions

- Used Weak Form interface to implement a model for heat and mass transfer in heated concrete.
- Automatic variable substitution makes COMSOL very powerful for implementing complex problems.
- Model was also used to verify the implementation with Cast3M.
- Model was able to reproduce data from experiments.
- Calibration is needed for missing material properties (and model deficiencies).