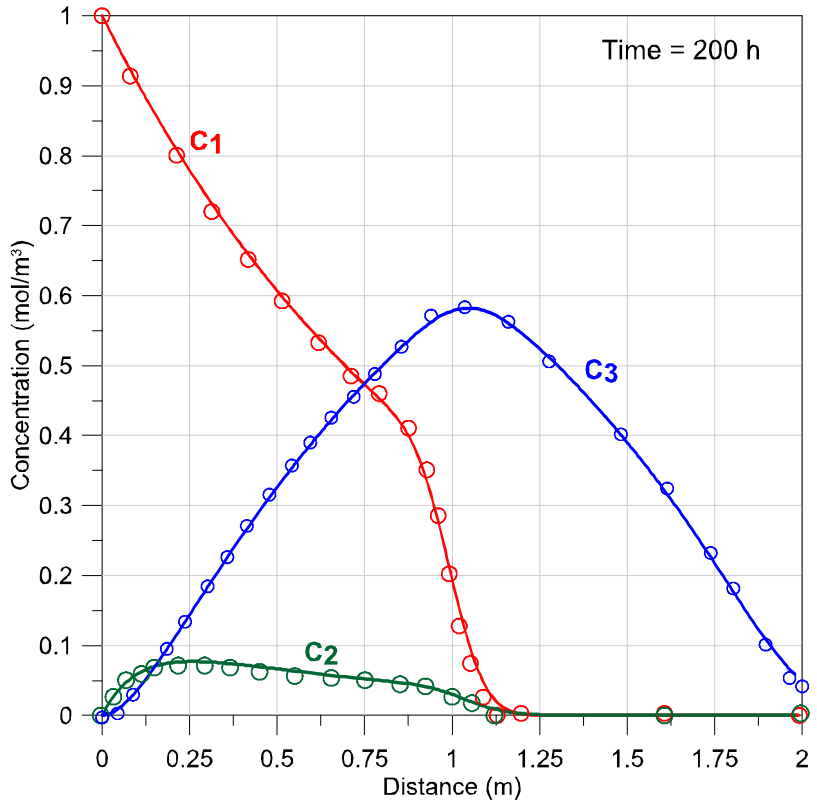
GeMA – Multispecies reactive transport example

Tecgraf
PUC-RIO

16/07/2018 – Version 1.0

The example

- This example presents an analysis of reactive transport on a rectangular plate subjected to Dirichlet (prescribed concentration) boundary conditions and multiple species with linear reaction rates.



Results agree well with the numerical and analytical solutions mentioned in VAN GENUCHTEN, M. TH. (1985), “Convective-dispersive transport of solutes involved in sequential first-order decay reactions”, Comput. Geosci. 11, 129–147.

The Problem

- The problem solve simultaneous convective-dispersive transport of three species with sorption and decay. The convective part is governed by a flow field in the x direction.
- The model equations involved in this specific case are:

$$k_1 \frac{\partial C_1}{\partial t} = \nabla \cdot (D \nabla C_1) + \mathbf{v} \nabla C_1 - k_1 \lambda_1 C_1$$

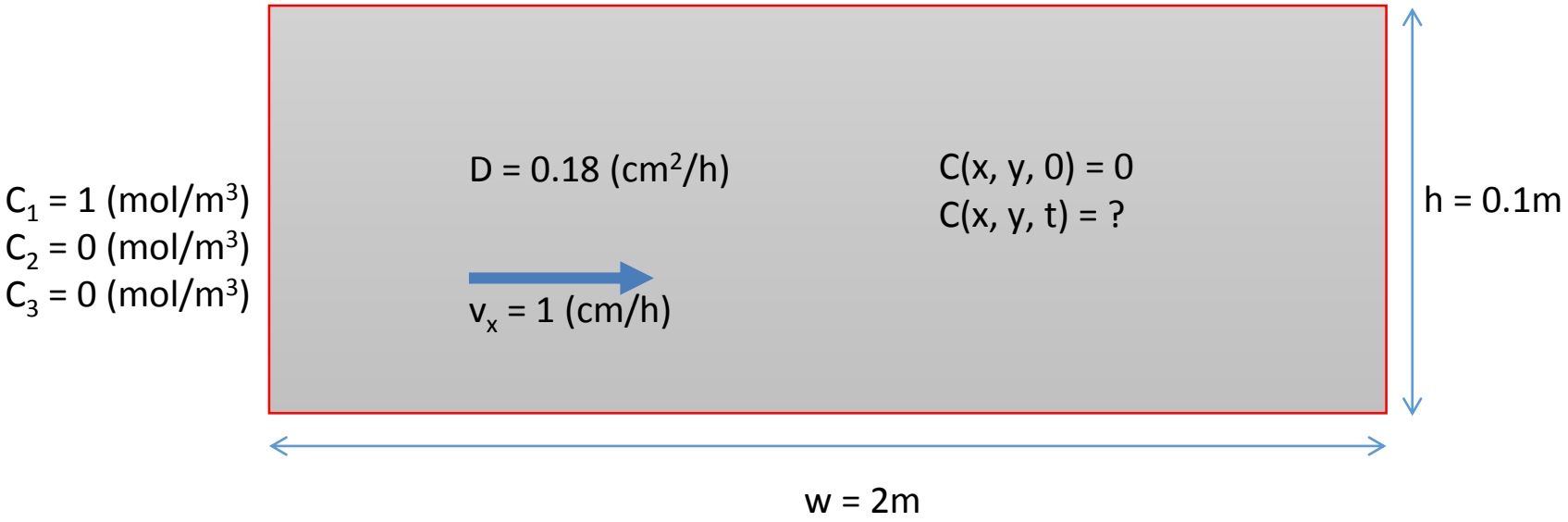
$$k_2 \frac{\partial C_2}{\partial t} = \nabla \cdot (D \nabla C_2) + \mathbf{v} \nabla C_2 + k_1 \lambda_1 C_1 - k_2 \lambda_2 C_2$$

$$k_3 \frac{\partial C_3}{\partial t} = \nabla \cdot (D \nabla C_3) + \mathbf{v} \nabla C_3 + k_2 \lambda_2 C_2 - k_3 \lambda_3 C_3$$

where C_1, C_2, and C_3 are species concentrations, D is the diffusion coefficient, *v* is the fluid velocity, λ_1, λ_2, and λ_3 are the species reaction rates and finally k_1, k_2, and k_3 are the retardation coefficients for each species.

The Problem

- This problem has two distinct parts. The first is the convective and diffusive transport on a rectangular plate subjected to prescribed concentrations on its left border:



- The second is the reactive part, whose parameters are: λ_1=0.005 (1/h), λ_2=0.1 (1/h), and λ_3=0.0 (1/h) for the species reaction rates and k_1=2, k_2=1, and k_3=1 for the retardation coefficients.

Model file: Model parameters

For a Multispecies convective diffusive simulation, the state variable should be a vector value named C (concentration), with dimension equal to the number of species.

The number of species

```
StateVar{id = 'C', description = 'concentration - 3 species', dim = 3,
        format='8.4g', groupName = 'chemical'}
```

The group name is used by the fem solver to group types of state variables and to define tolerances for each one

Model file: Material properties

The main material properties are the Diffusion coefficient, the material porosity, the velocity field and the retardation factor for each species. Reaction rates are given as physics attributes and not as a material property.

```
PropertySet
{
  id          = 'MatProp',
  typeName    = 'GemaPropertySet',
  description = 'Material parameters',
  properties  = {
        {id = 'D',   description = 'diffusion', unit = 'cm^2/h'},
        {id = 'phi', description = 'porosity', unit = ''},
        {id = 'v',   description = 'velocity', unit = 'cm/h', dim = 2},
        {id = 'k',   description = 'retardation', dim = 3},
  },
  values = {
        {D = 0.18, phi = 1.0, v = {1.0, 0}, k = {2.0, 1.0, 1.0}}
  }
}
```

Diffusion

Porosity

Velocity field (x, y).

Retardation factor for each species

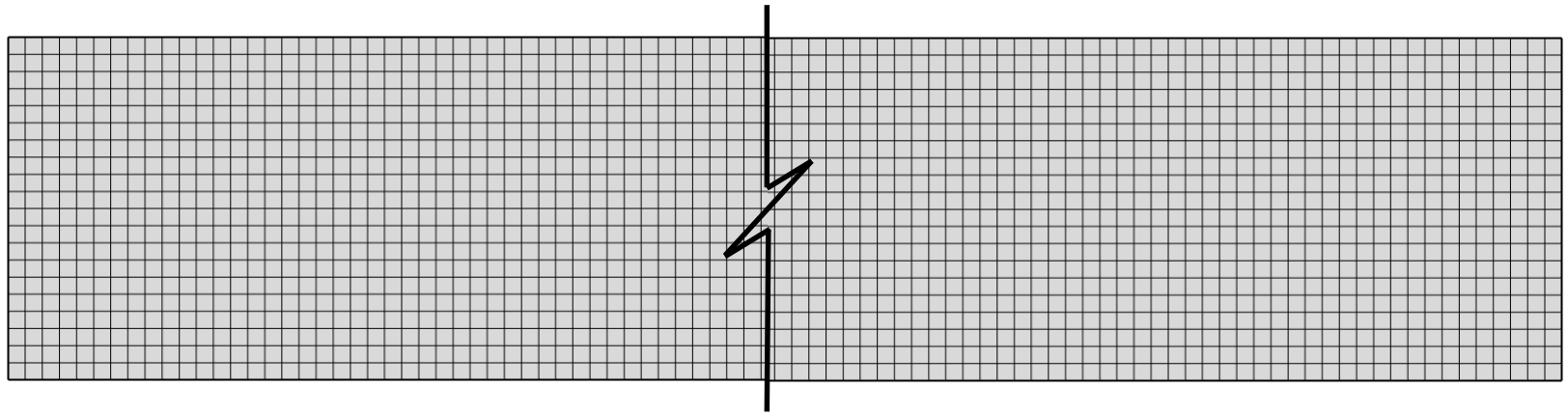
Model file: Mesh

Because the solute concentration is large near the inlet, a fine mesh is needed in the flow direction to avoid potential numerical oscillations. In this model the mesh grid is created with 800 cells in the x direction and 20 cells in the y direction.

```
dofile('$SCRIPTS/meshLib.lua') -- Loads the 'meshLib' auxiliary functions

local xpoints = meshLib.regularSpacing(0.0, 200, 800)
local ypoints = meshLib.regularSpacing(0.0, 10, 20)

local nodes, elements, borders = meshLib.build2DGrid('quad4', xpoints, ypoints, nil, {MatProp = 1})
```



Model file: Boundary conditions

The problem begins with a 1.0 concentration for C1 at the left border and without any concentration for the other two species.

```
BoundaryCondition {
  id   = 'Border concentration',
  type = 'node concentration',
  mesh = 'mesh',

  properties  = {
    {id = `C',  description = 'External concentration applied on the node', dim = 3, unit = ''},
  },

  nodeValues = {
    {'gridLeft',  {1.00, 0.0, 0.0}},
  }
}
```

Solution file: Physics & Orchestration

```
PhysicalMethod {
  id         = 'ChemistryPhysics',
  typeName   = 'ChemicalFemPhysics.linear',
  type       = 'fem',

  mesh = 'mesh',
  boundaryConditions = {'bc1'},

  multiRetardation = true,

  reactionMatrix = {{-0.01,  0.00,  0.00},
                    { 0.01, -0.10,  0.00},
                    { 0.00,  0.10,  0.00}},

  reactionMatrixUnit = '1/h',
}
```

The simulation physics is implemented by the "linear" object type from the ChemialFemPhysics plugin

Enable one retardation factor per species

Matrix with species reaction rates

Reaction rate unit

$$k_1 \frac{\partial C_1}{\partial t} = \nabla \cdot (D \nabla C_1) + \mathbf{v} \nabla C_1 - k_1 \lambda_1 C_1$$

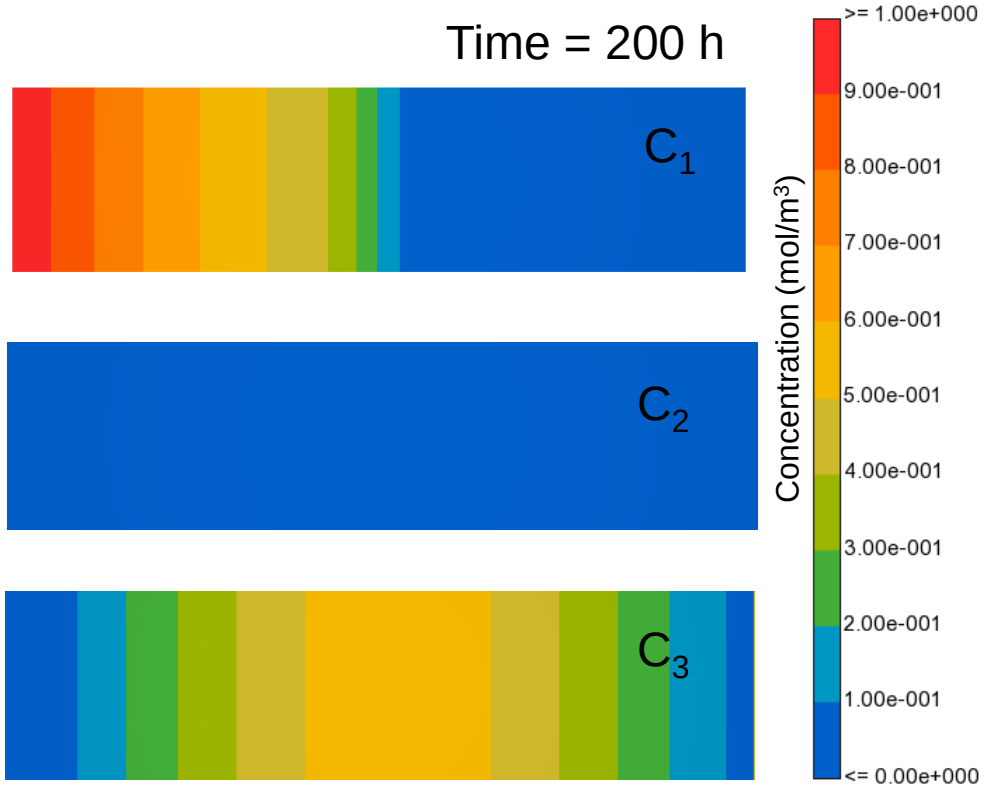
$$k_2 \frac{\partial C_2}{\partial t} = \nabla \cdot (D \nabla C_2) + \mathbf{v} \nabla C_2 + k_1 \lambda_1 C_1 - k_2 \lambda_2 C_2$$

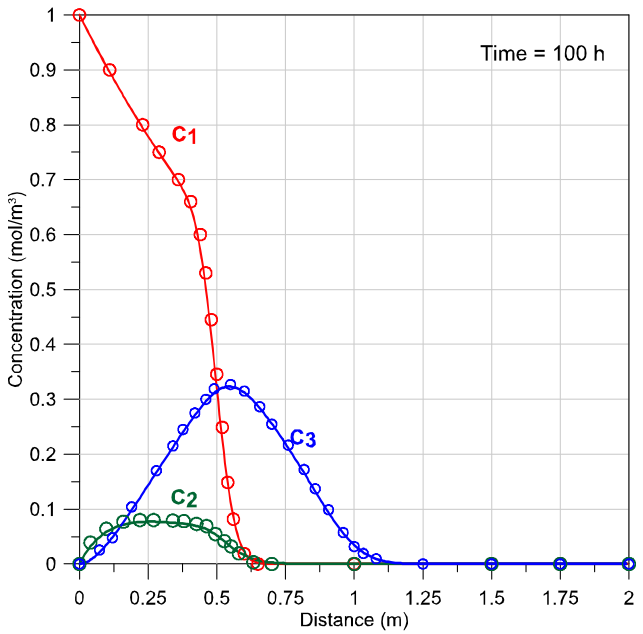
$$k_3 \frac{\partial C_3}{\partial t} = \nabla \cdot (D \nabla C_3) + \mathbf{v} \nabla C_3 + k_2 \lambda_2 C_2 - k_3 \lambda_3 C_3$$

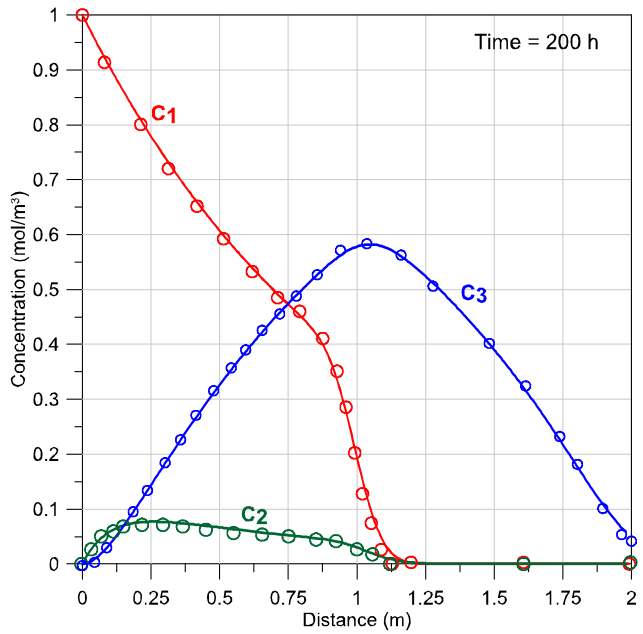
The orchestration uses the non-linear solver to take advantage of its adaptive time step solver. See the solution file for more details

Results

The results of simulation are:







The numerical results are shown in solid lines and analytical results in circles. The results agree well with the numerical and analytical solutions mentioned in van Genuchten (1985).