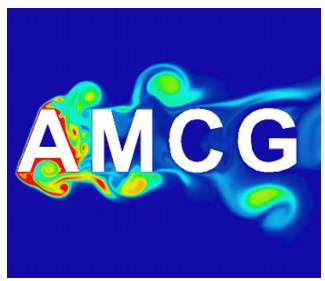
Exponential integrators and two-scale approaches for unsaturated water flow in heterogeneous soils

Elliot Carr
Postdoctoral Research Associate, AMCG, NORMS



NORMS Meeting, 26 November 2014

Overview of Presentation

- Modelling of unsaturated water flow
- Time integration methods for unsaturated water flow: Exponential integrators
- Two-scale numerical modelling of unsaturated water flow in soils exhibiting small-scale heterogeneities

Overview of Presentation

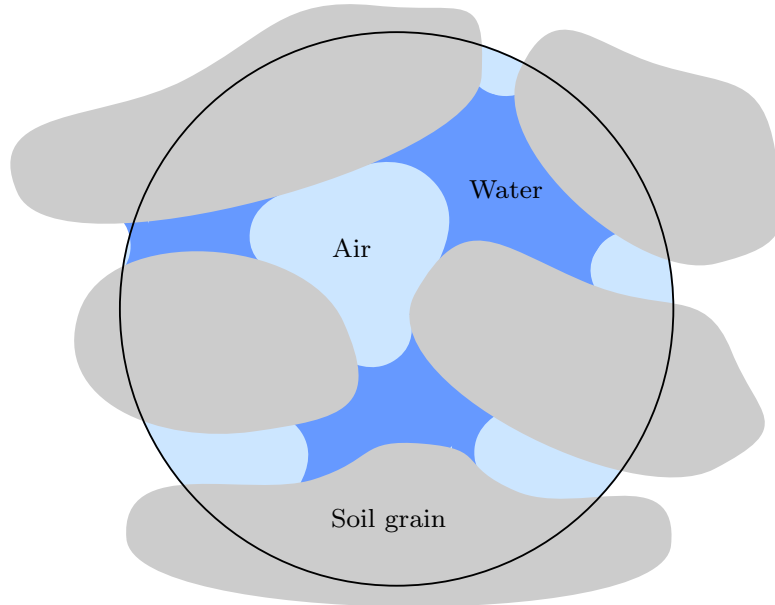
- Modelling of unsaturated water flow
- Time integration methods for unsaturated water flow: Exponential integrators
- Two-scale numerical modelling of unsaturated water flow in soils exhibiting small-scale heterogeneities

Unsaturated water flow

- Immiscible two-phase (air and water) flow
- Darcy's Law:

$$\boldsymbol{q}_\alpha = -\frac{k_{r,\alpha}}{\mu_\alpha} K \left(\nabla p_\alpha - \rho_\alpha \boldsymbol{g} \right)$$

- Capillary pressure $p_c = p_a - p_w$



- Mass conservation for each phase (assuming incompressibility)

$$\sum_{\alpha \in \{a,w\}} \boldsymbol{q}_\alpha = \boldsymbol{0}\,, \qquad \frac{\partial \left(\phi S_\alpha \right)}{\partial t} + \nabla \cdot \boldsymbol{q}_\alpha = 0$$

- Sum of saturations $S_w + S_a = 1$

Richards' equation

- Assumptions:
  (a) air phase is at constant and atmospheric pressure
  (b) gravity acts in the negative y direction (2D)

- Single equation for the water phase:

$$\frac{\partial \theta(h)}{\partial t} + \nabla \cdot (-K(h) \nabla (h + x_2)) = 0$$

  where $h = p_c/(\rho_w g)$ is the capillary pressure head, $\theta = \phi S_w$ is the moisture content and K is the hydraulic conducitivity.

- Closure relationships [van Genuchten (1980)]:

$$\theta(h) = \theta_r + (\theta_s - \theta_r) S_e(h)$$

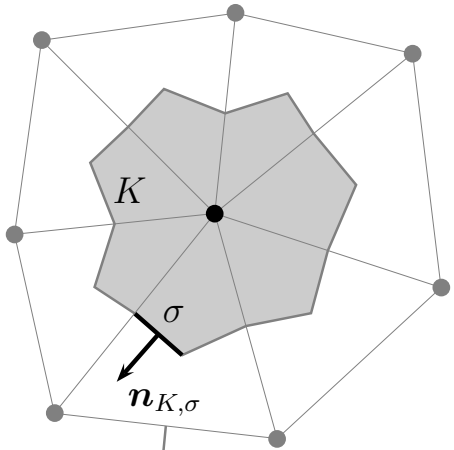
$$K(h) = K_{\text{sat}} \sqrt{S_e} \left(1 - \left(1 - S_e^{1/m}\right)^m\right)^2$$

$$S_e(h) = (1 + (-\alpha h)^n)^{-m}$$

Overview of Presentation

- Modelling of unsaturated water flow
- Time integration methods for unsaturated water flow: Exponential integrators
- Two-scale numerical modelling of unsaturated water flow in soils exhibiting small-scale heterogeneities

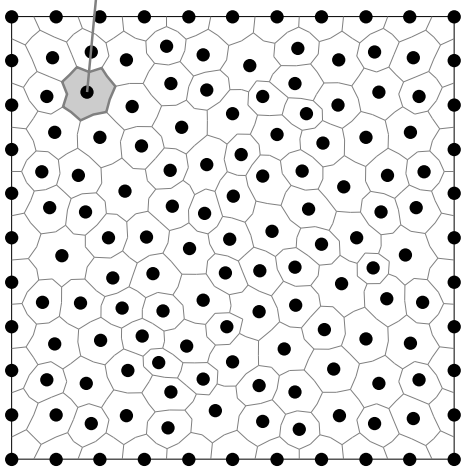
Control Volume Method



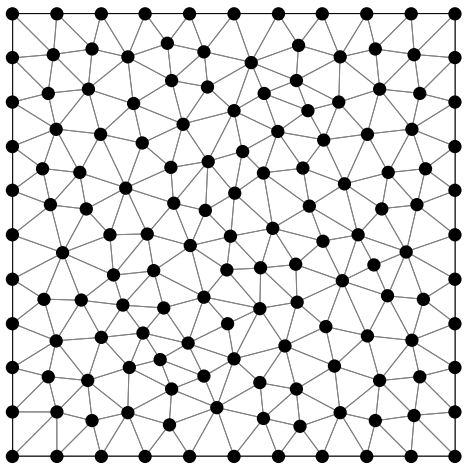
Spatial discretisation:

$$\frac{d}{dt}\int_K \theta \, dV + \sum_{\sigma \subset \partial K} \int_\sigma -K(h)\nabla\left(h + x_2\right) \cdot \boldsymbol{n}_{K,\sigma} \, dS = 0$$

CV mesh



Mesh



Time integration methods

- Spatial discretisation can be expressed in the form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}(\boldsymbol{u}) \,, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0 \in \mathbb{R}^N \,,$$

  where $\boldsymbol{u} = (h_1, \ldots, h_N)^T$ and N is the number of nodes.

- Classical explicit and implicit methods

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = \int_{t_n}^{t_{n+1}} \boldsymbol{g}(\boldsymbol{u}) \, dt$$

$$\begin{aligned}
\boldsymbol{u}_{n+1} - \boldsymbol{u}_n &= \tau_n g(\boldsymbol{u}_n) && \textit{Forward Euler method} \\
\boldsymbol{u}_{n+1} - \boldsymbol{u}_n &= \tau_n g(\boldsymbol{u}_{n+1}) && \textit{Backward Euler method} \\
\boldsymbol{u}_{n+1} - \boldsymbol{u}_n &= \tau_n \left[(1-\theta)\boldsymbol{g}(\boldsymbol{u}_n) + \theta \boldsymbol{g}(\boldsymbol{u}_{n+1}) \right] && \textit{Theta method}
\end{aligned}$$

  where $\theta = 1/2$ gives the classical Crank-Nicolson method.

- Fluidity/IC-FERST uses the Theta method.

Exponential integrators

An *exponential integrator* is any time integration method that involves the exponential of the Jacobian matrix [Minchev and Wright (2005)]

Prototype method (Exponential Euler): At each time step, solve the linearised problem:

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}(\boldsymbol{u}) \quad \xrightarrow{\text{Linearise}} \quad \frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}_n + \boldsymbol{J}_n(\boldsymbol{u} - \boldsymbol{u}_n)$$

where $\boldsymbol{g}_n = \boldsymbol{g}(\boldsymbol{u}_n)$ and $\boldsymbol{J}_n = \boldsymbol{J}(\boldsymbol{u}_n)$, exactly over a single time step:

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \tau_n \varphi(\tau_n \boldsymbol{J}_n)\boldsymbol{g}_n$$

where $\varphi(\boldsymbol{A}) = \boldsymbol{A}^{-1}(e^{\boldsymbol{A}} - \boldsymbol{I})$.

- Explicit
- Second order [Hochbruck et al. (1998)]
- Exact for linear problems $\Rightarrow$ A-stable in the linear stability sense

Krylov subspace methods for matrix functions

- Extract an approximation from a small m dimensional Krylov subspace

$$\mathcal{K}_m(\boldsymbol{J}_n, \boldsymbol{g}_n) = \text{span}\left\{\boldsymbol{g}_n, \boldsymbol{J}_n\boldsymbol{g}_n, \ldots, \boldsymbol{J}_n^{m-1}\boldsymbol{g}_n\right\} \subset \mathbb{R}^N$$

- Construct an orthonormal basis for the subspace using Arnoldi's method

$$\boldsymbol{J}_n\boldsymbol{V}_m = \boldsymbol{V}_m\boldsymbol{H}_m + \beta_m\boldsymbol{v}_{m+1}\boldsymbol{e}_m^T$$

  where the columns of $\boldsymbol{V}_m = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m]$ form an orthonormal basis for $\mathcal{K}_m$.

- Arnoldi's method requires only matrix-vector products with $\boldsymbol{J}_n$

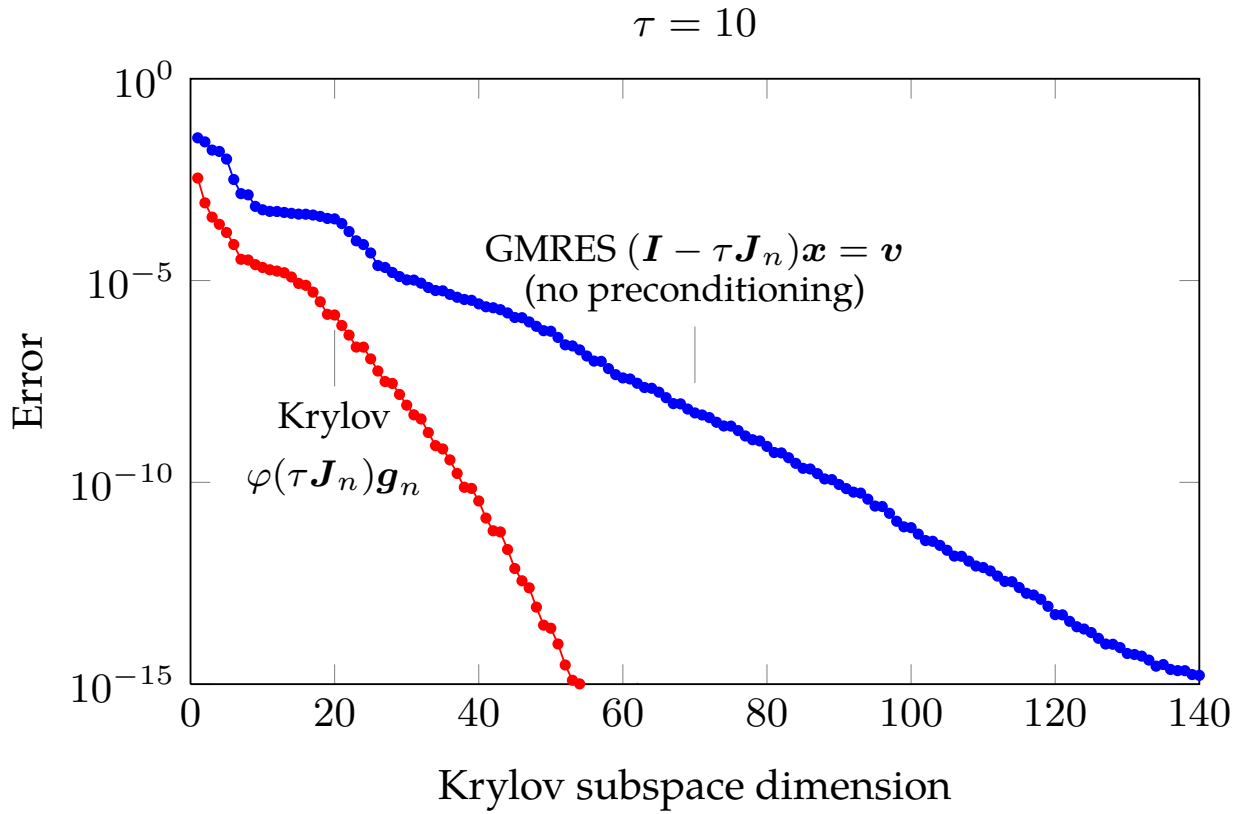
$$\boldsymbol{J}_n\boldsymbol{v} \approx [\boldsymbol{g}(\boldsymbol{u}_n + \varepsilon\boldsymbol{v}) - \boldsymbol{g}(\boldsymbol{u}_n)]/\varepsilon$$

- Standard Krylov subspace approximation [Hochbruck et al. (1998)]

$$\varphi(\tau_n\boldsymbol{J}_n)\boldsymbol{g}_n \approx \boldsymbol{V}_m\left[\|\boldsymbol{g}_n\|_2\varphi(\tau_n\boldsymbol{H}_m)\boldsymbol{e}_1\right]$$

- Very accurate approximations can be obtained for $m \ll N$ so the matrix $\varphi(\tau\boldsymbol{H}_m)$ can be computed cheaply (e.g., using Padé approximation).

One attraction of exponential integrators



Higher order exponential integrators

- Write the ODE system in the form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}_n + \boldsymbol{J}_n(\boldsymbol{u} - \boldsymbol{u}_n) + \boldsymbol{r}(\boldsymbol{u}) \,,$$

  where $\boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{g}(\boldsymbol{u}) - \boldsymbol{g}_n - \boldsymbol{J}_n(\boldsymbol{u} - \boldsymbol{u}_n)$.

- Using the integrating factor $e^{-t\boldsymbol{J}_n}$:

$$\frac{d}{dt}\left(e^{-t\boldsymbol{J}_n}\boldsymbol{u}(t)\right) = e^{-t\boldsymbol{J}_n}\left(\boldsymbol{g}_n - \boldsymbol{J}_n\boldsymbol{u}_n\right) + e^{-t\boldsymbol{J}_n}\boldsymbol{r}(\boldsymbol{u}) \,.$$

  the exact solution can be expressed as

$$\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}_n + \tau_n\varphi_1(\tau_n\boldsymbol{J}_n)\boldsymbol{g}_n + \int_{t_n}^{t_{n+1}} e^{(t_{n+1}-t)\boldsymbol{J}_n}\boldsymbol{r}(\boldsymbol{u})\,dt \,.$$

- This formula is the starting point for developing higher order methods [Tokman (2006), Hochbruck et al. (2009)].

Backward Differentiation Formula (BDF)

- Higher-order generalisations of the Backward Euler method
- Evaluate ODE system at $t = t_{n+1}$ to give

$$\frac{d\boldsymbol{u}}{dt}(t_{n+1}) = \boldsymbol{g}(\boldsymbol{u}_{n+1})$$

- For a BDF of order q: Interpolate the solution at $(t_{n+1-i}, \boldsymbol{u}_{n+1-i})$, $i = 0, \ldots, q$ using a polynomial of degree q:

$$\boldsymbol{u}(t) \approx \sum_{i=0}^{q} \ell_i(t) \boldsymbol{u}_{n+1-i} \qquad \ell_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^{q} \frac{t - t_{n+1-j}}{t_{n+1-i} - t_{n+1-j}}$$

  where $\ell_i(t)$ are the usual Lagrange interpolating polynomials.

- Differentiate interpolant and evaluate at $t = t_{n+1}$ to obtain

$$\frac{d\boldsymbol{u}}{dt}(t_{n+1}) \approx \sum_{i=0}^{q} \ell_i'(t) \boldsymbol{u}_{n+1-i}$$

Tested schemes

Exponential integrators

- Exponential Euler method (2nd order) [Carr et al. (2011)]
- Exponential Rosenbrock method (4th order) [Hochbruck et al. (2009)]
- At every time step, the error is estimated $\boldsymbol{\varepsilon}_{n+1} \approx \boldsymbol{u}_{n+1} - \hat{\boldsymbol{u}}_{n+1}$ using an embedded method $\hat{\boldsymbol{u}}_{n+1}$ and the time step adjusted to satisfy user prescribed error tolerances. Stepsize accepted if $\|\boldsymbol{\varepsilon}_{n+1}\|_{tol} \leq 1$ where

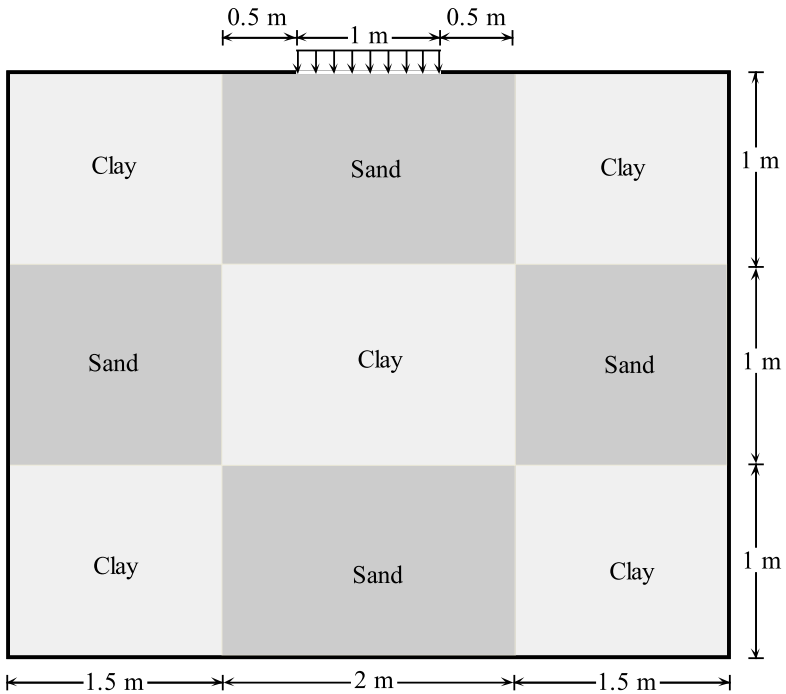
$$\|\boldsymbol{y}\|_{tol} = \|\boldsymbol{x}\|_2/\sqrt{N}\,, \qquad x_i = y_i/(tol_a + u_{n,i} tol_r)$$

Backward Differentiation Formulae (BDF)

- BDF Methods of order 2 and 4
- CVODE module of SUNDIALS [Hindmarsh et al. (2005)]
- Employs Newton iteration with an ILU preconditioned GMRES method.
- Periodically adjusts the order, with the goal of maximizing the stepsize
- Stepsize control

Test Case: Unsaturated water flow

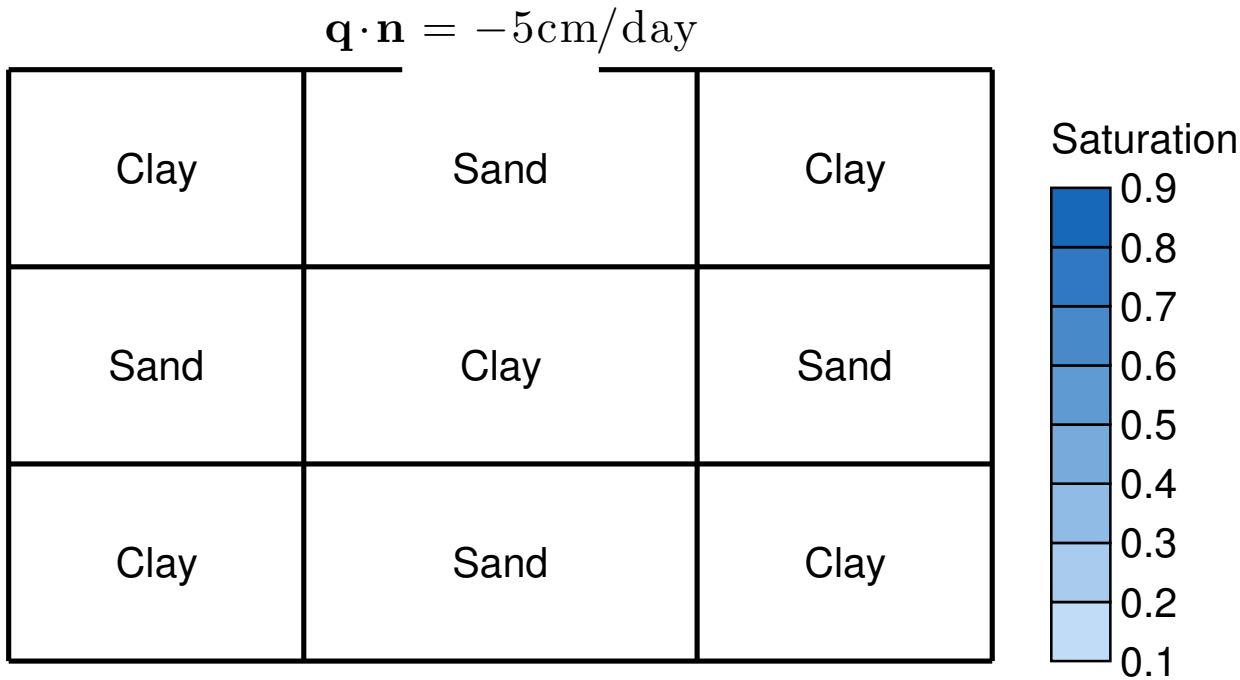
- Test Case [Kirkland et al. (1992), McBride et al. (2006), Carr et al. (2011, 2013)]



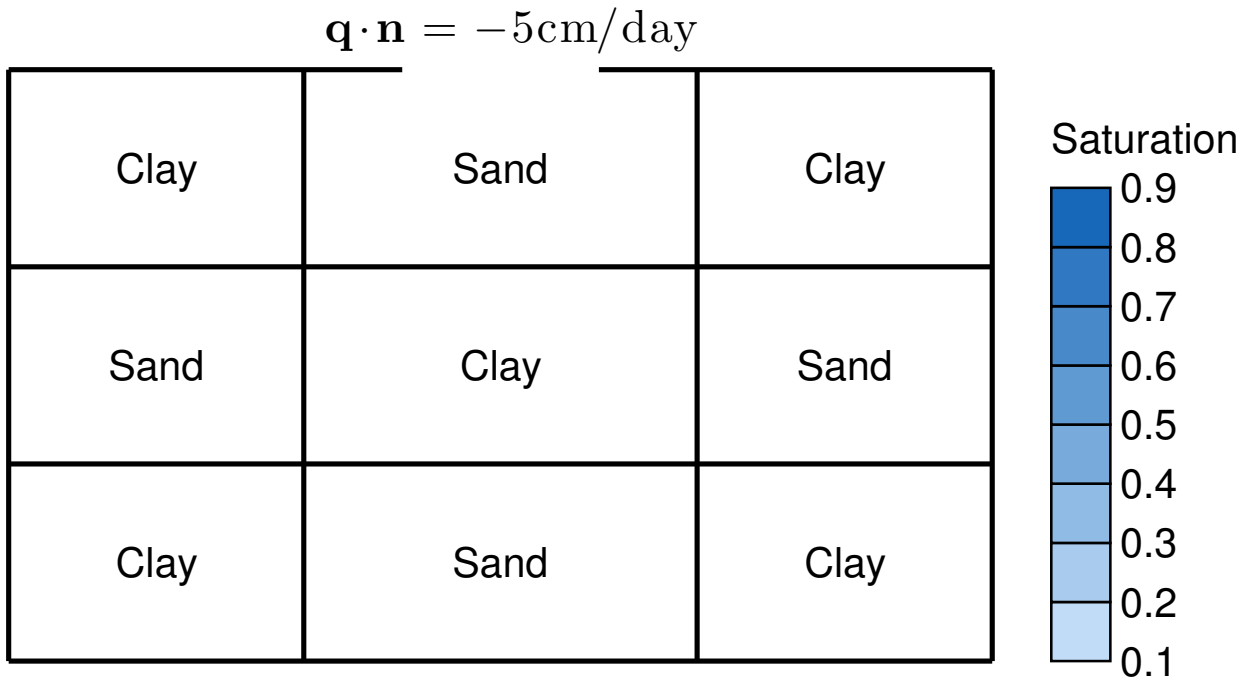
	Sand	Clay
K_{sat}	6.262e-5	1.516e-6
θ_r	0.0286	0.1060
θ_s	0.3658	0.4686
α	2.80	1.04
n	2.2390	1.3964

- Mesh: 132×132 grid, $N = 16641$.

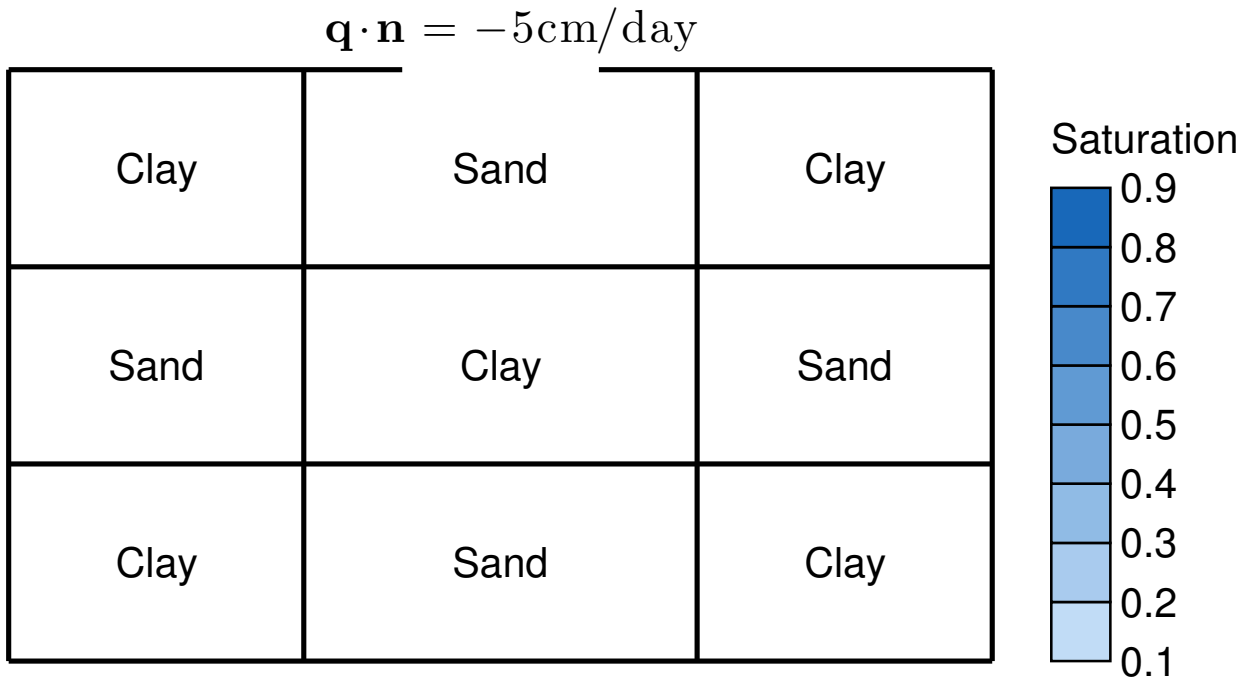
Test Problem: Water infiltration



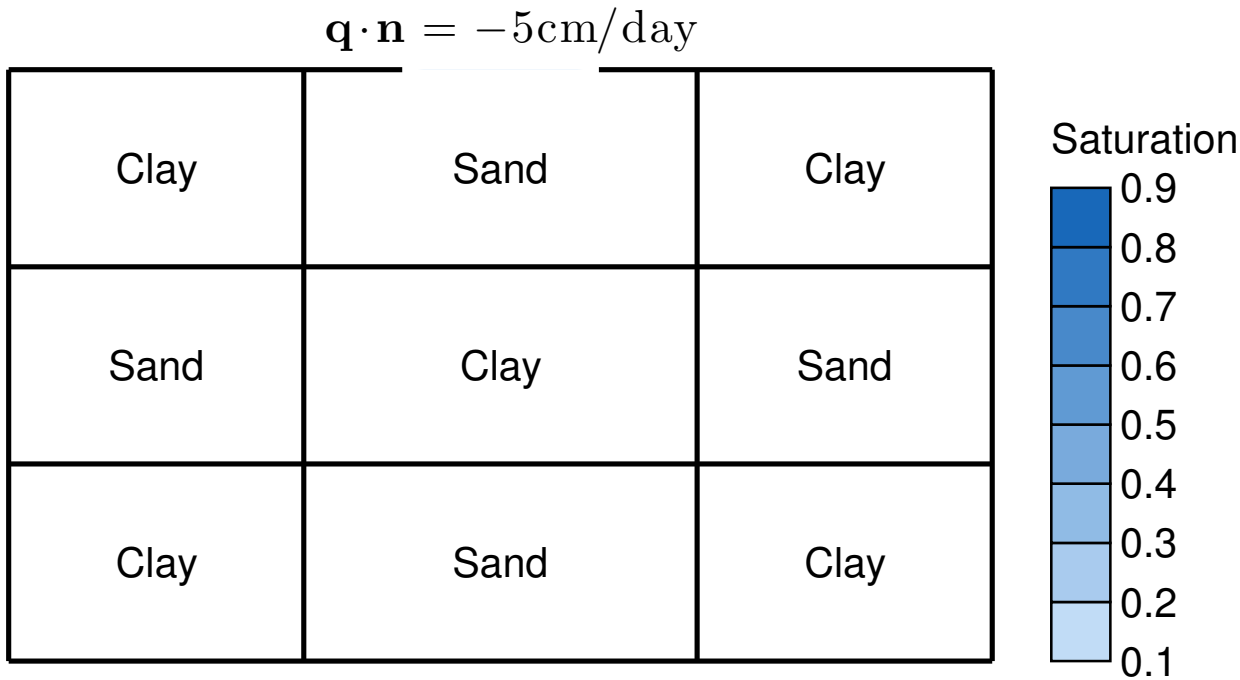
Test Problem: Water infiltration



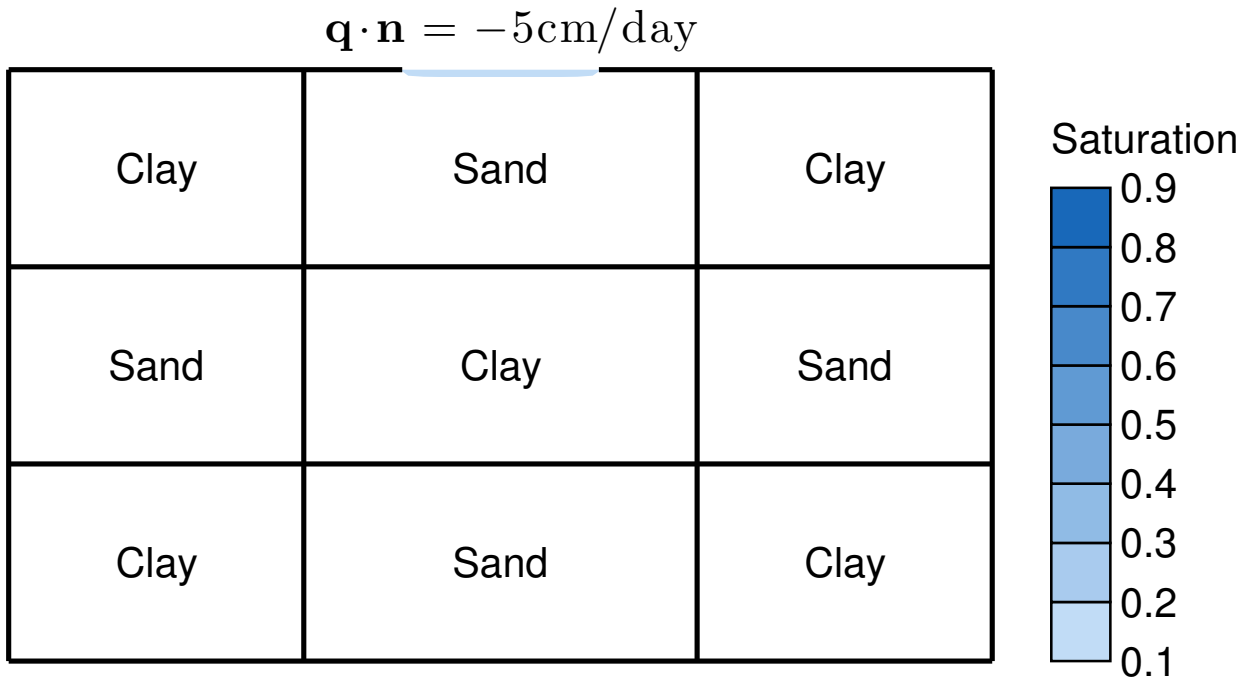
Test Problem: Water infiltration



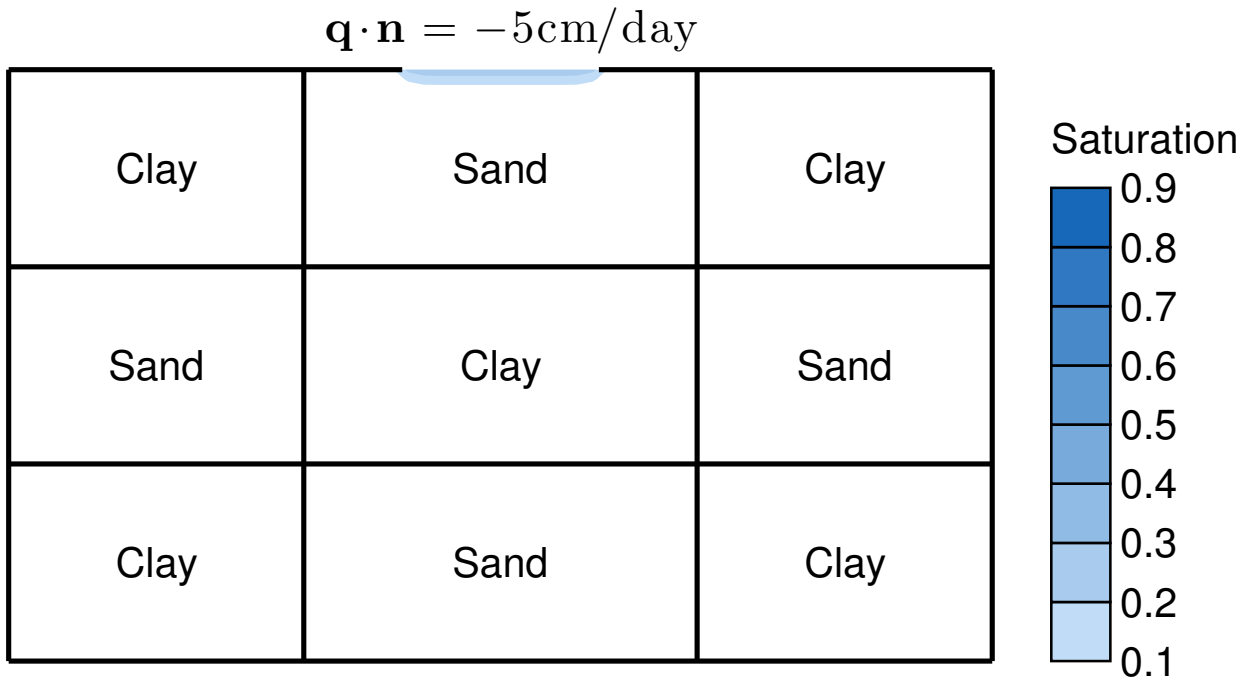
Test Problem: Water infiltration



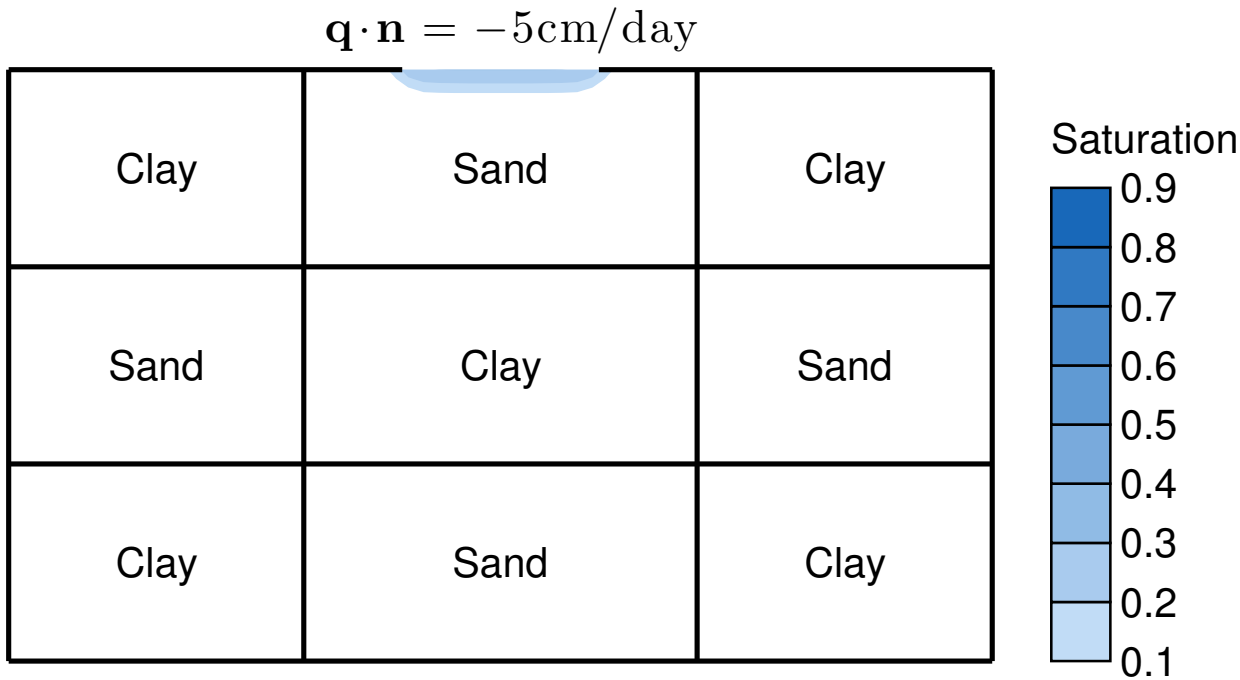
Test Problem: Water infiltration



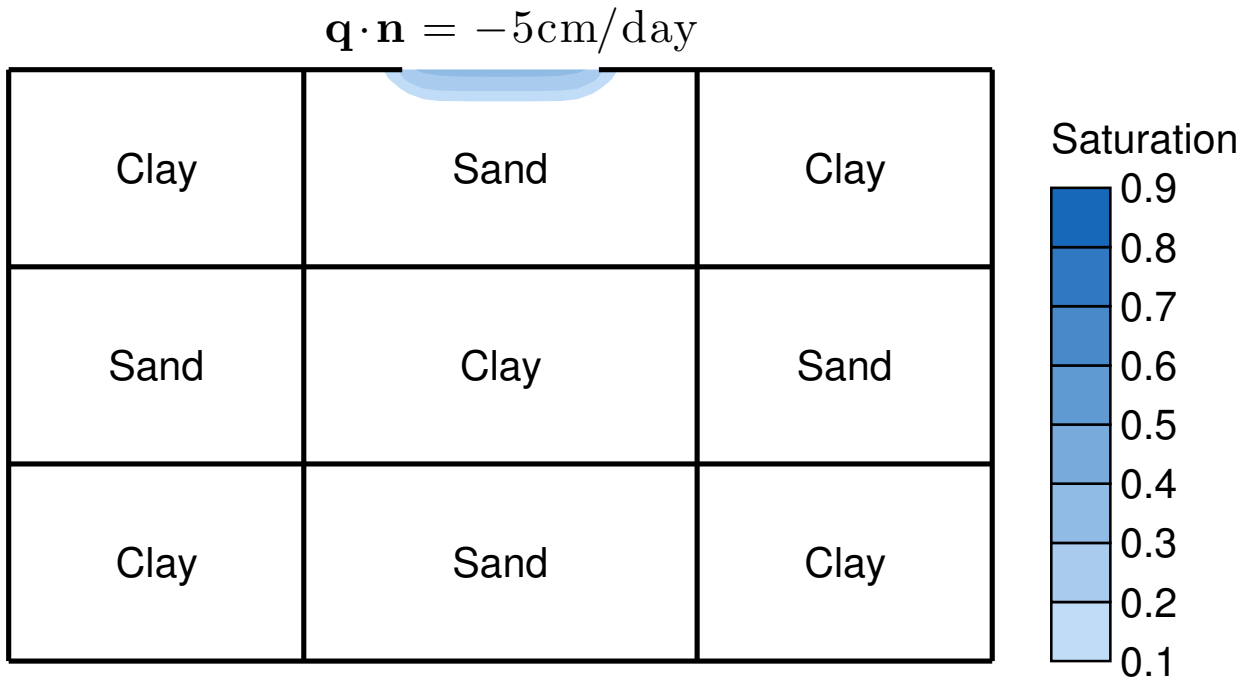
Test Problem: Water infiltration



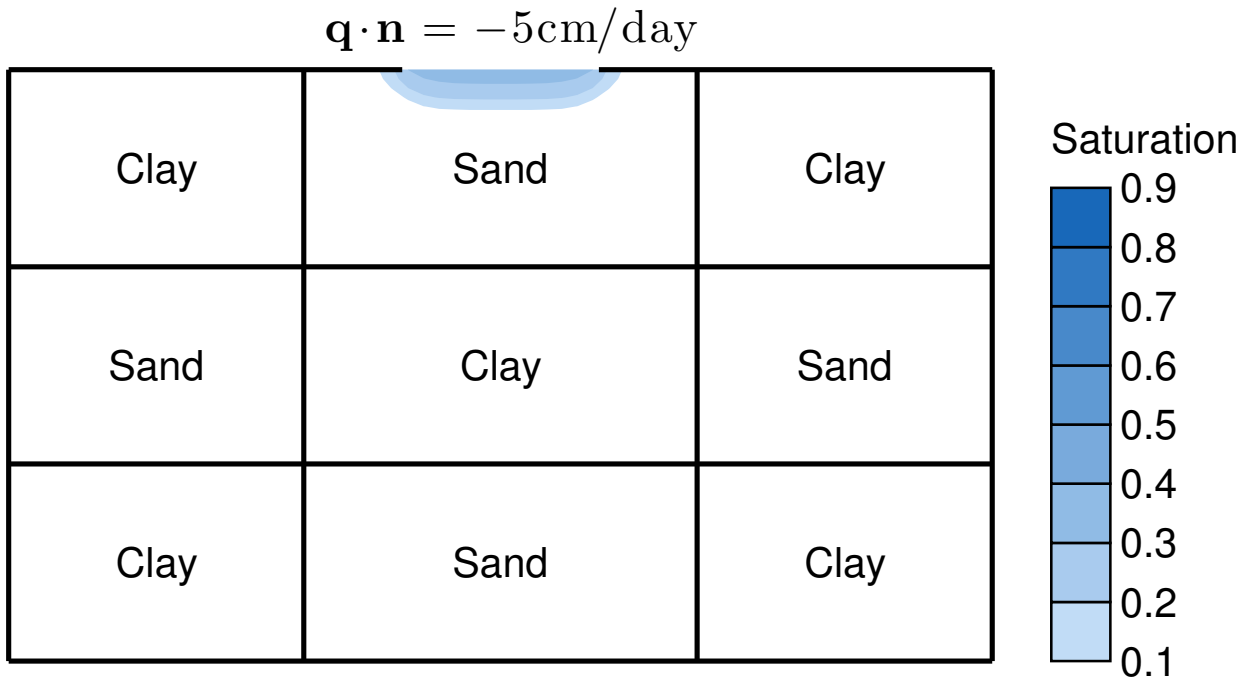
Test Problem: Water infiltration



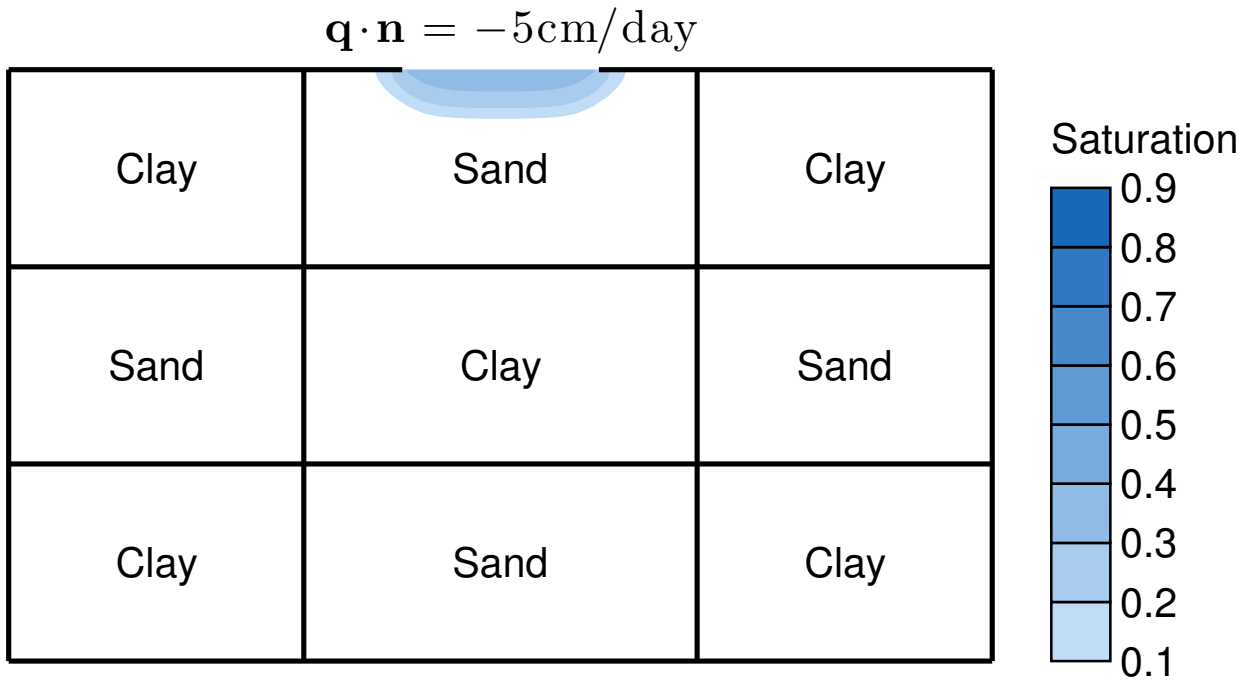
Test Problem: Water infiltration



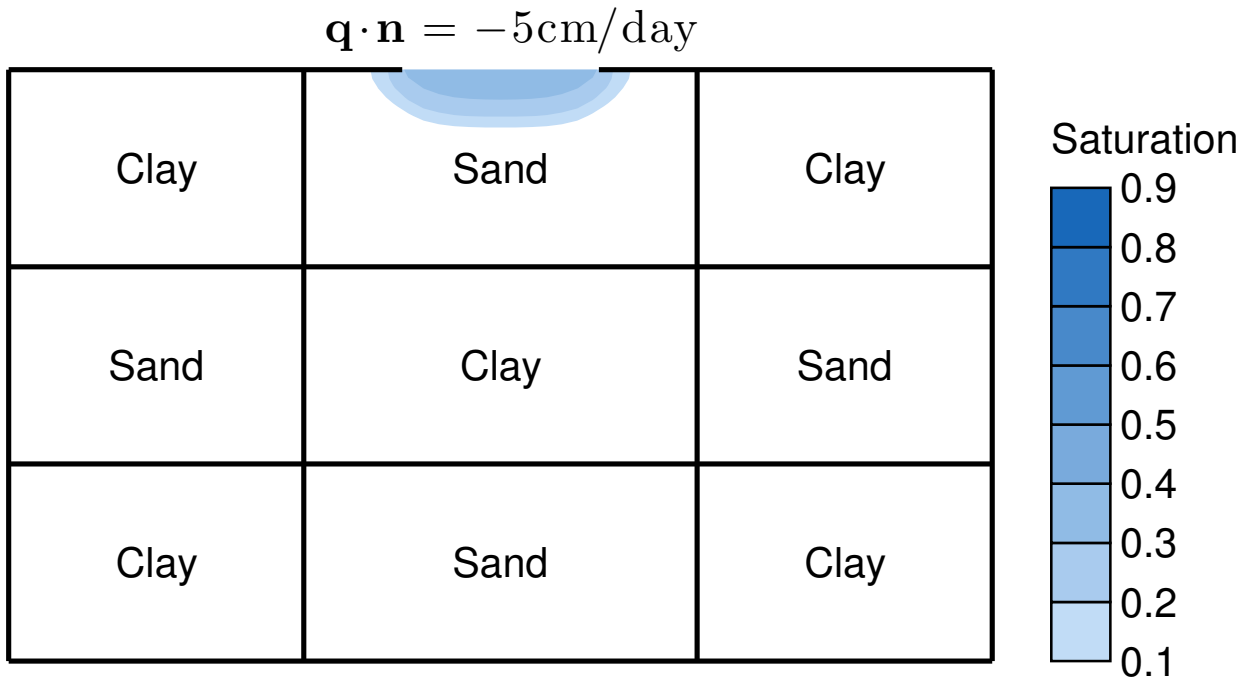
Test Problem: Water infiltration



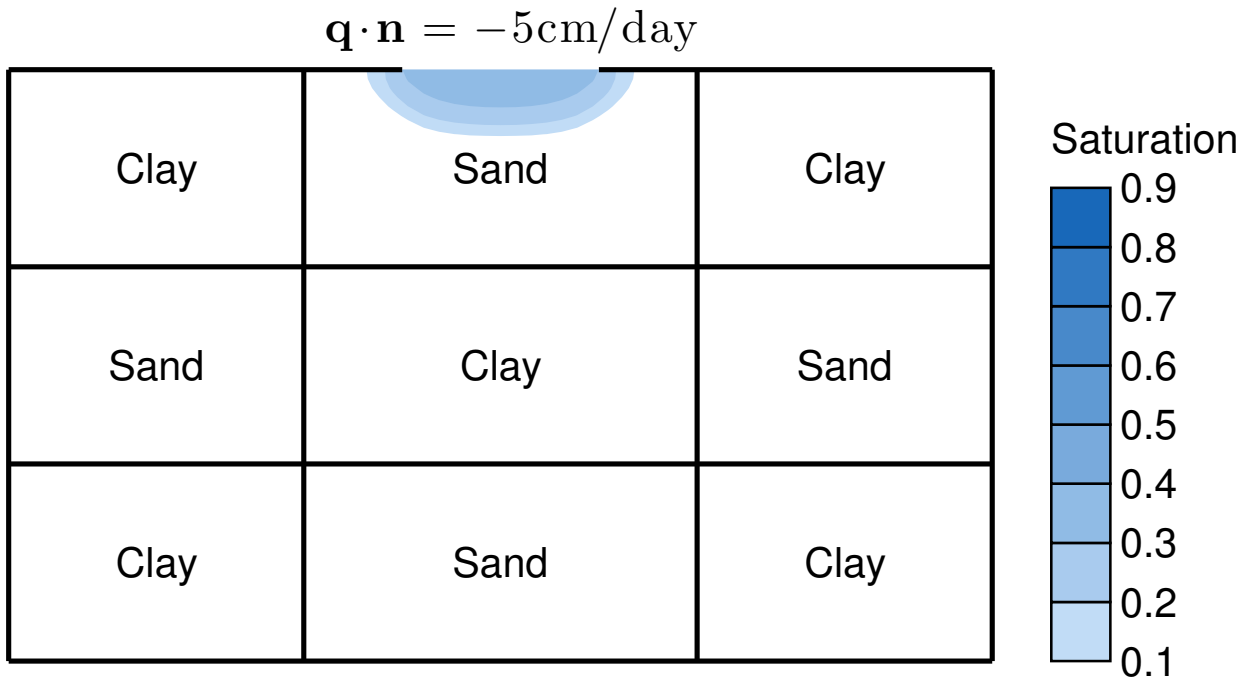
Test Problem: Water infiltration



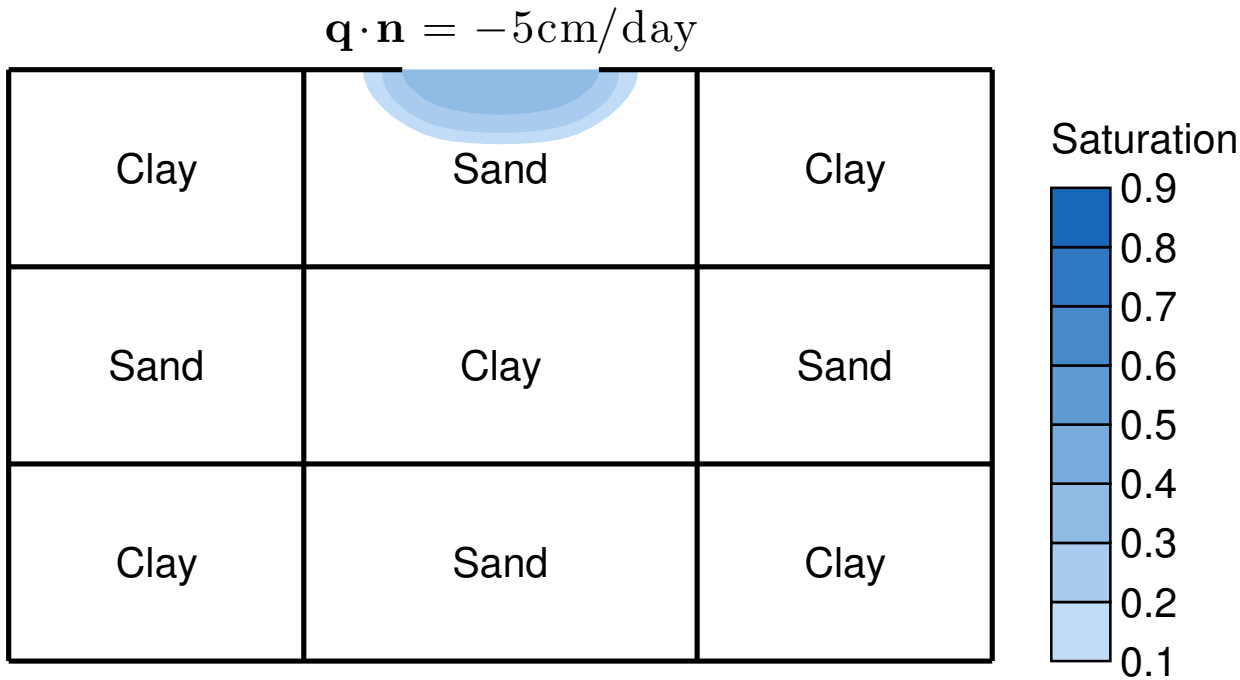
Test Problem: Water infiltration



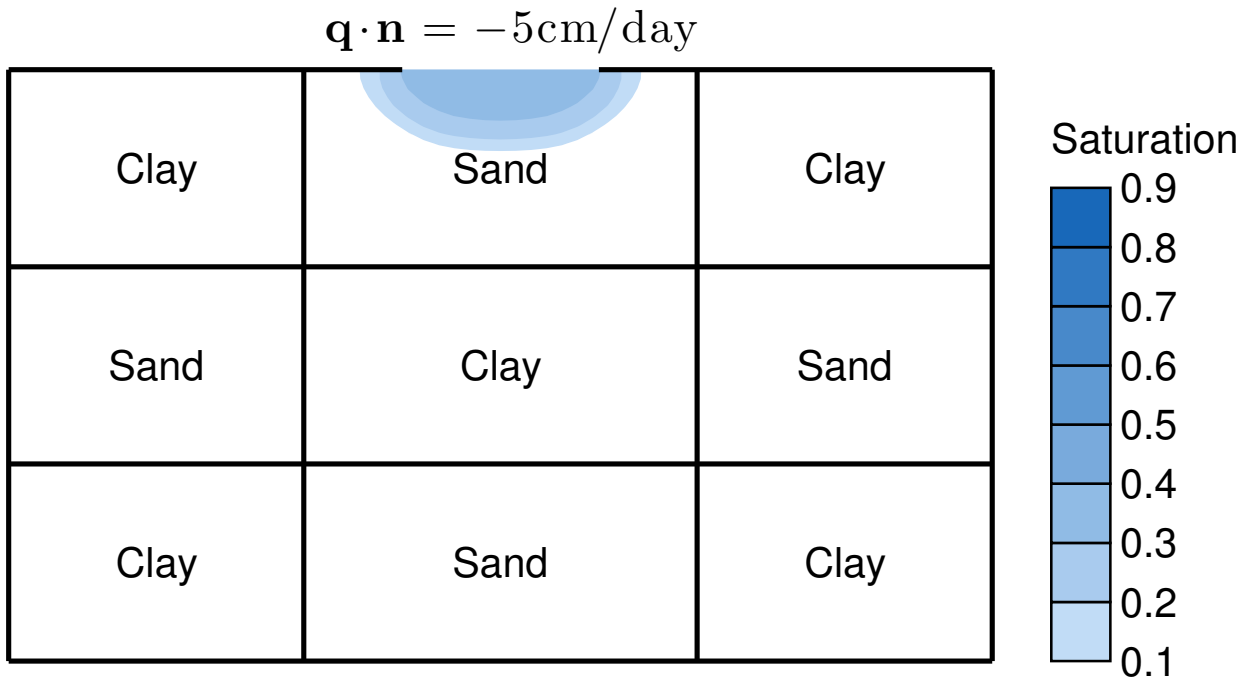
Test Problem: Water infiltration



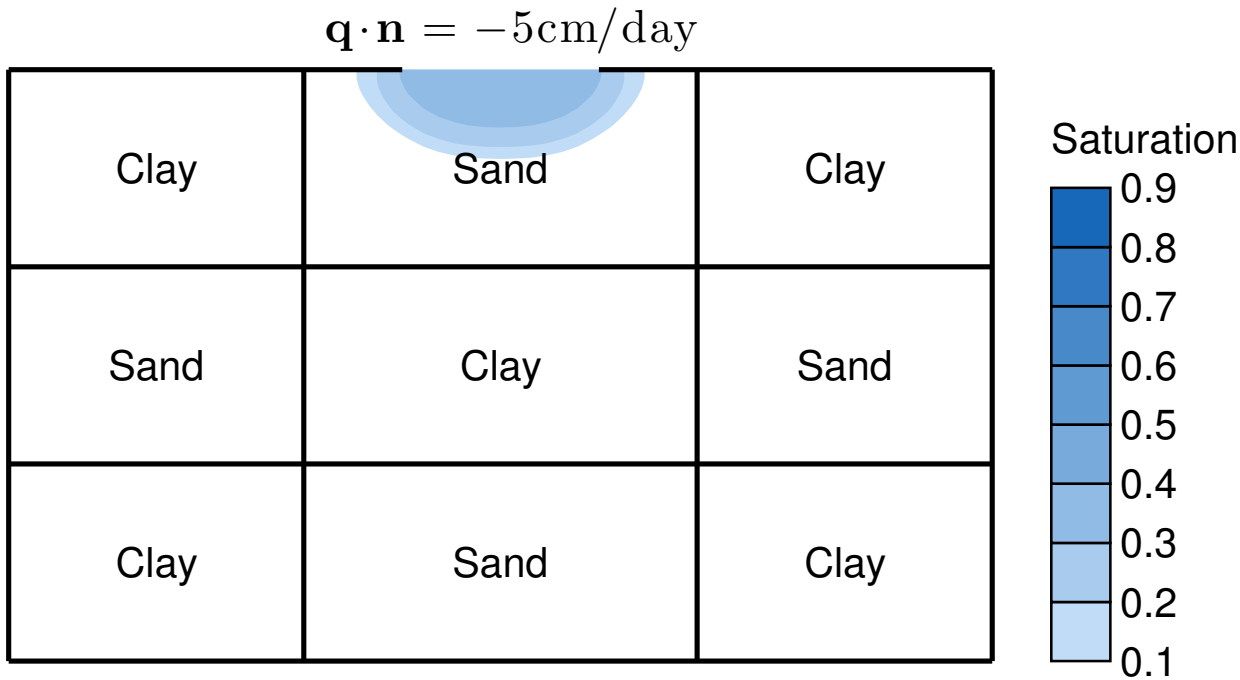
Test Problem: Water infiltration



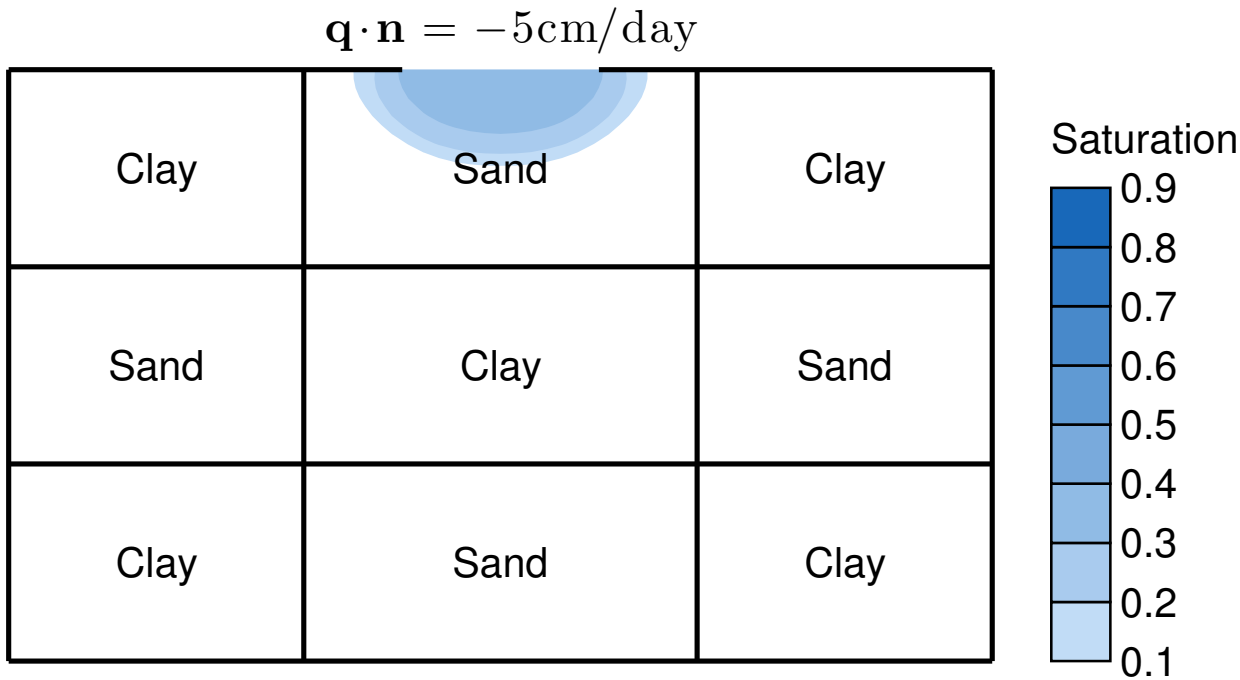
Test Problem: Water infiltration



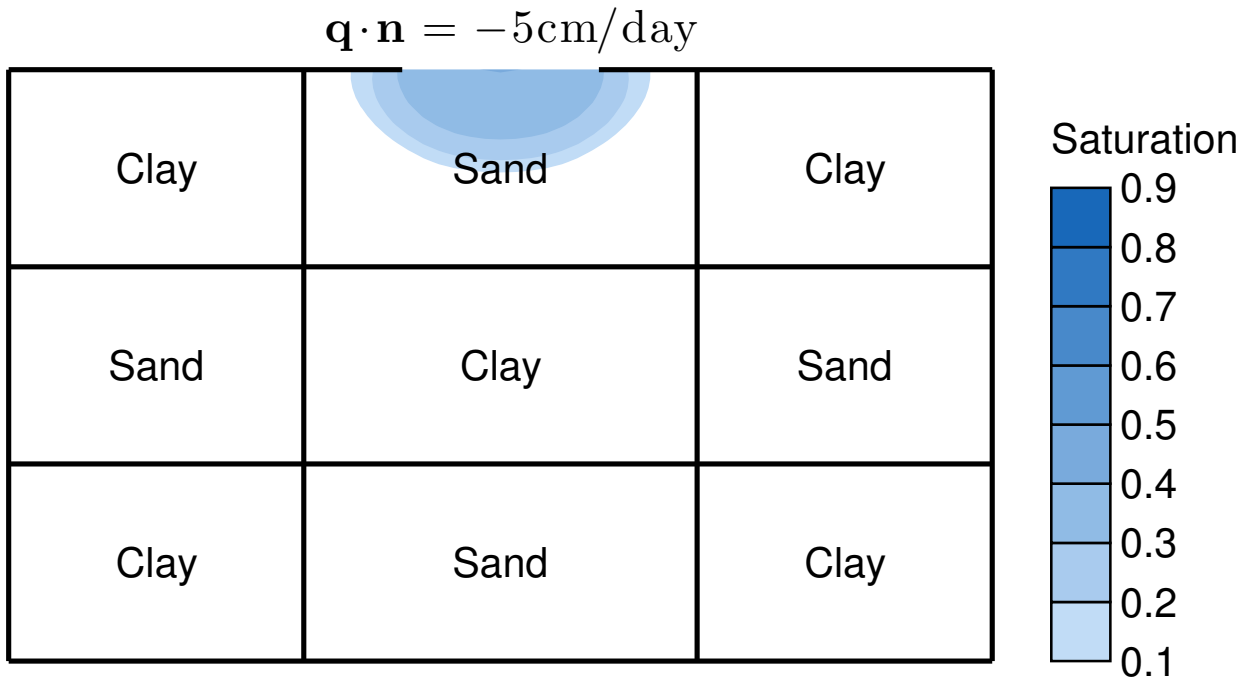
Test Problem: Water infiltration



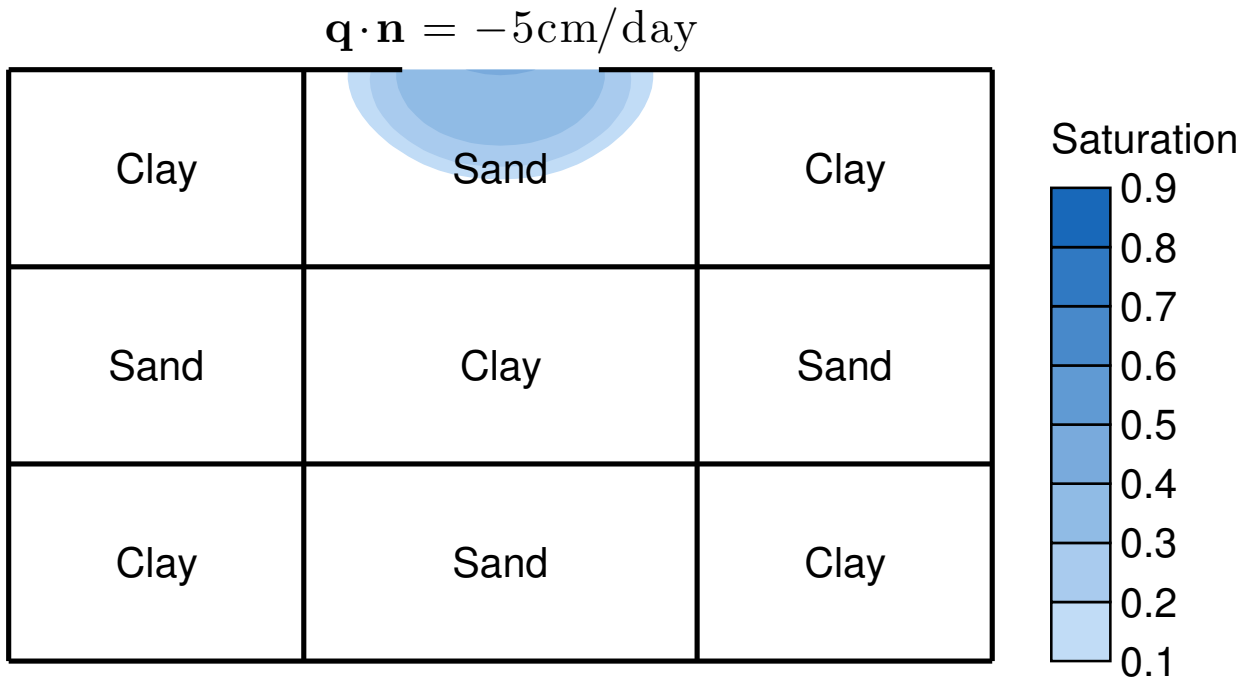
Test Problem: Water infiltration



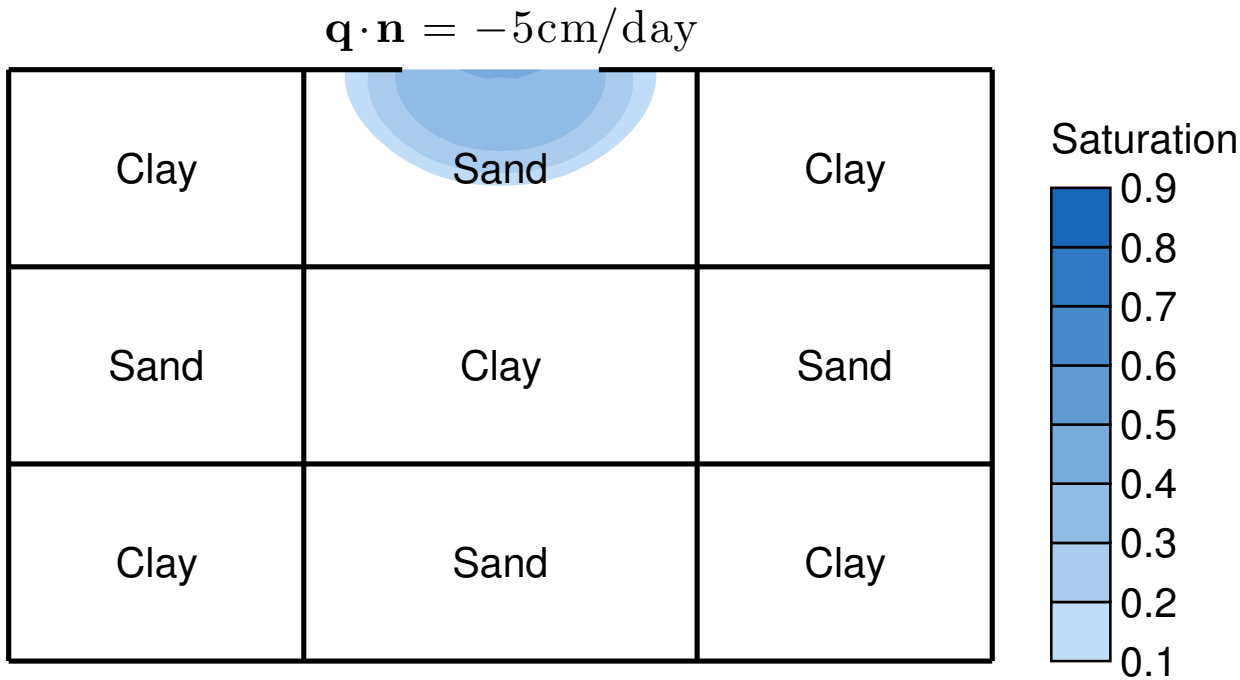
Test Problem: Water infiltration



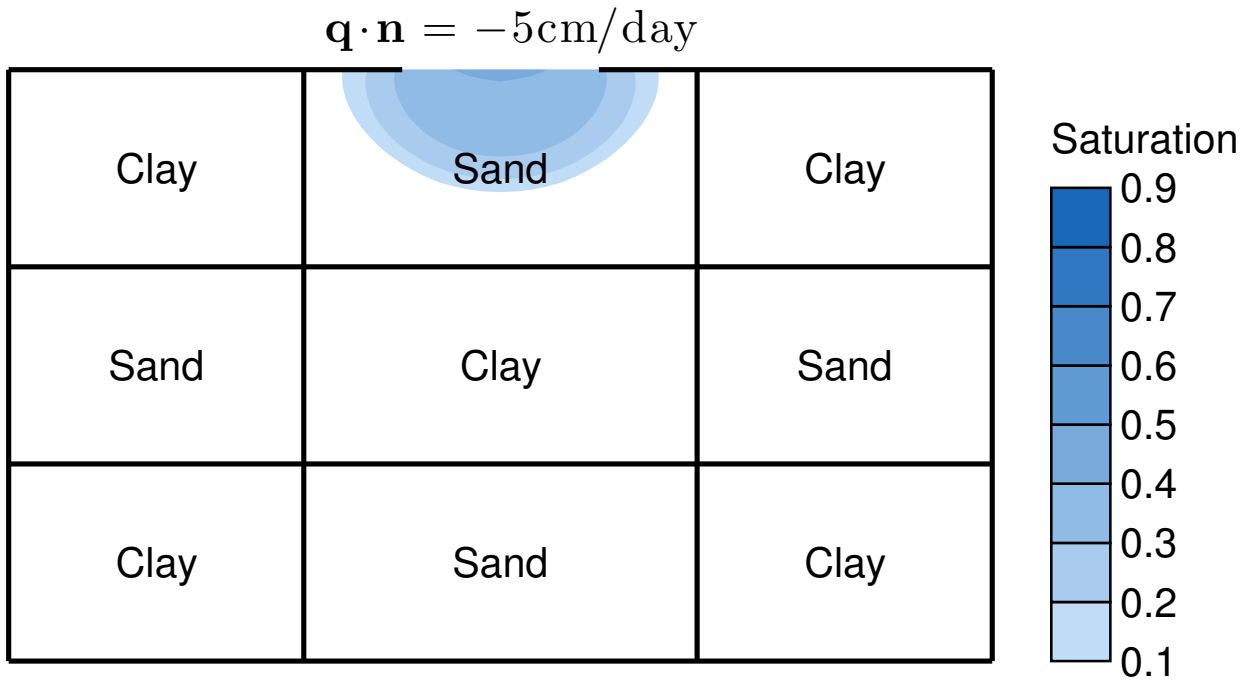
Test Problem: Water infiltration



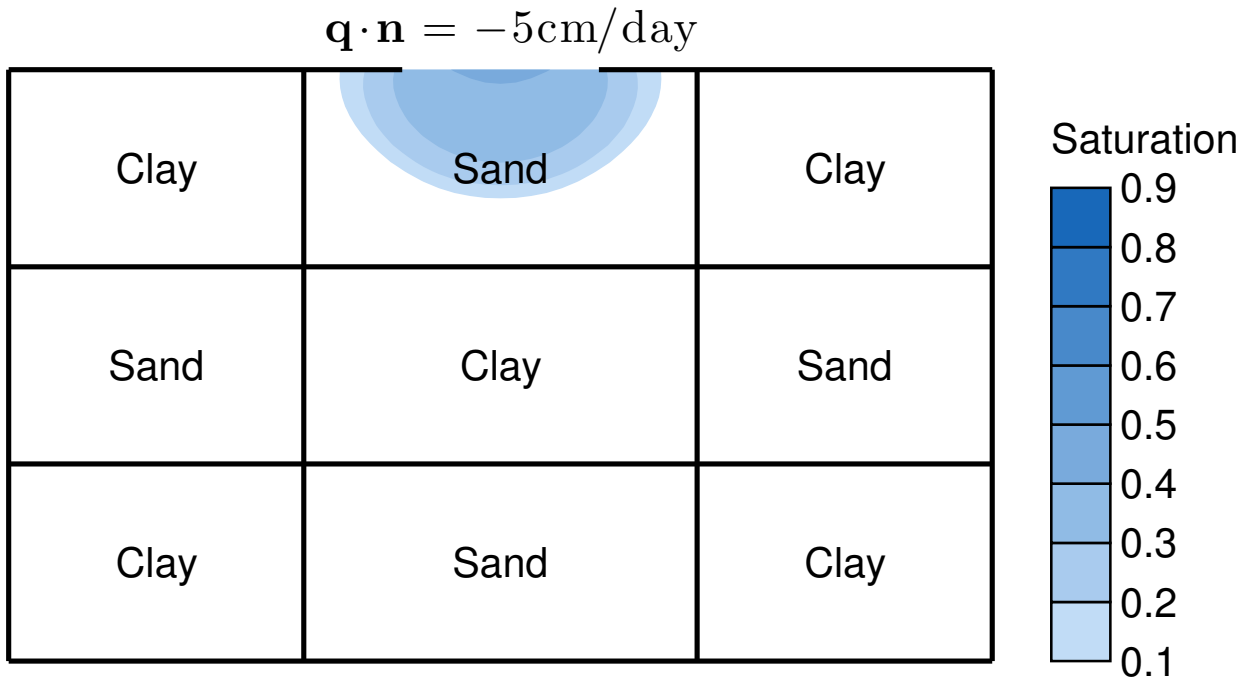
Test Problem: Water infiltration



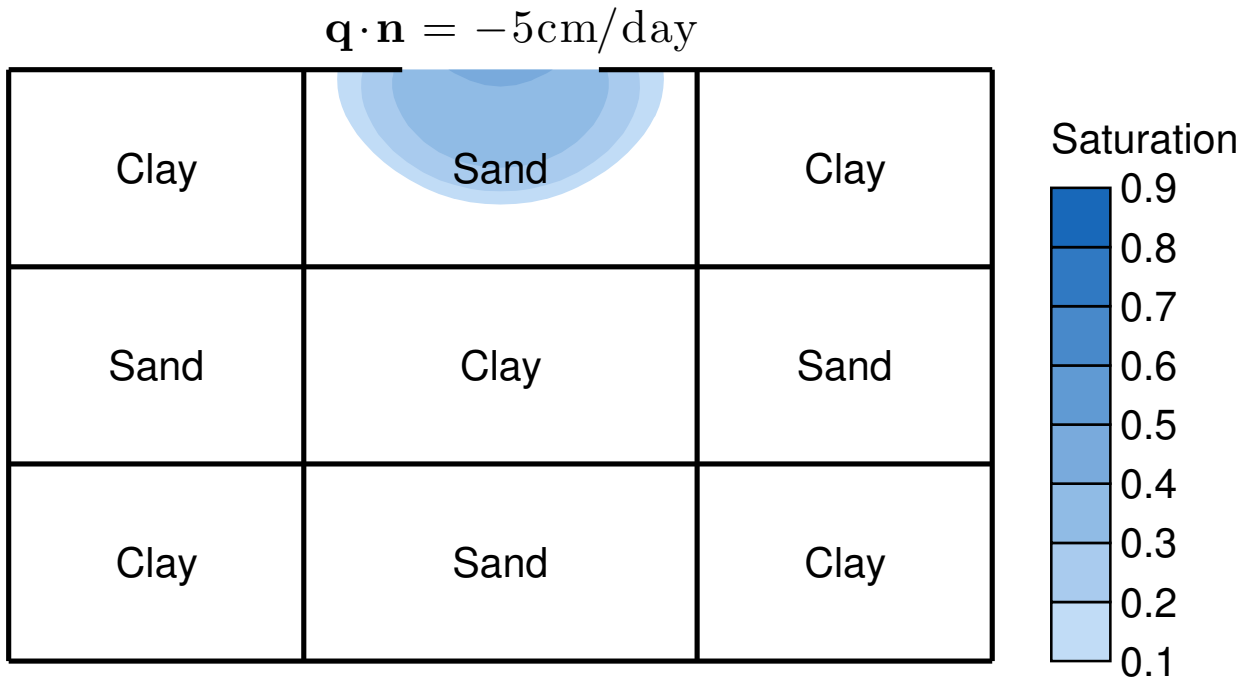
Test Problem: Water infiltration



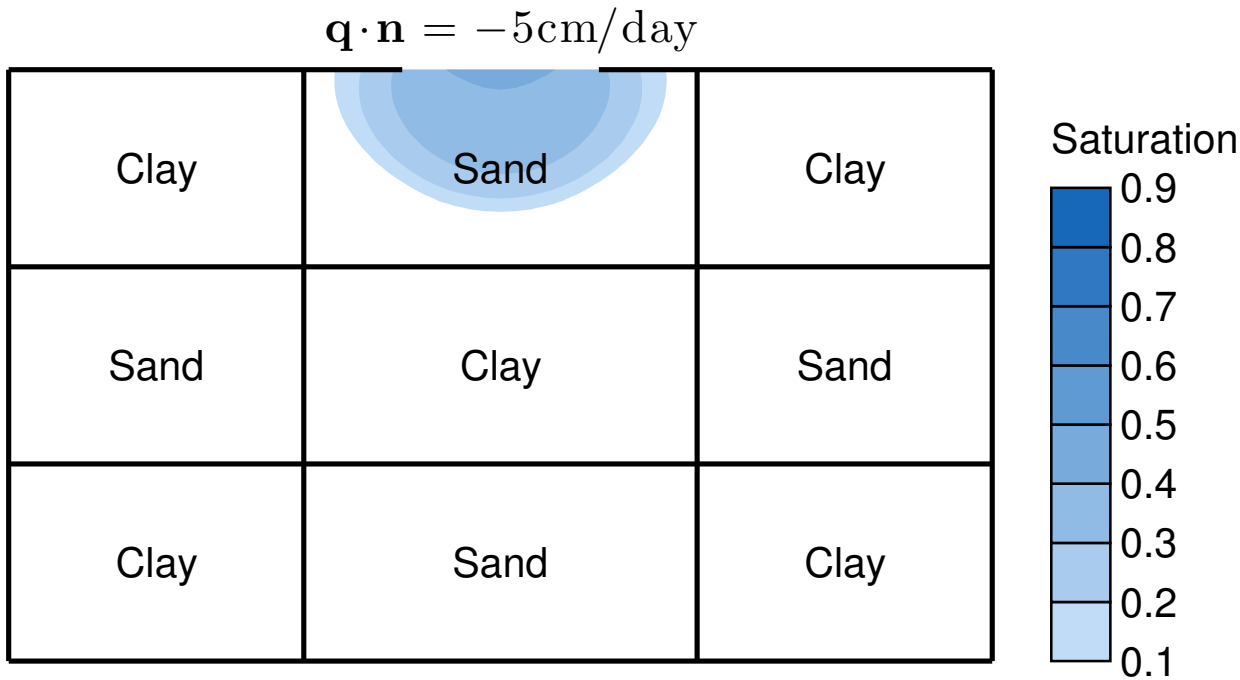
Test Problem: Water infiltration



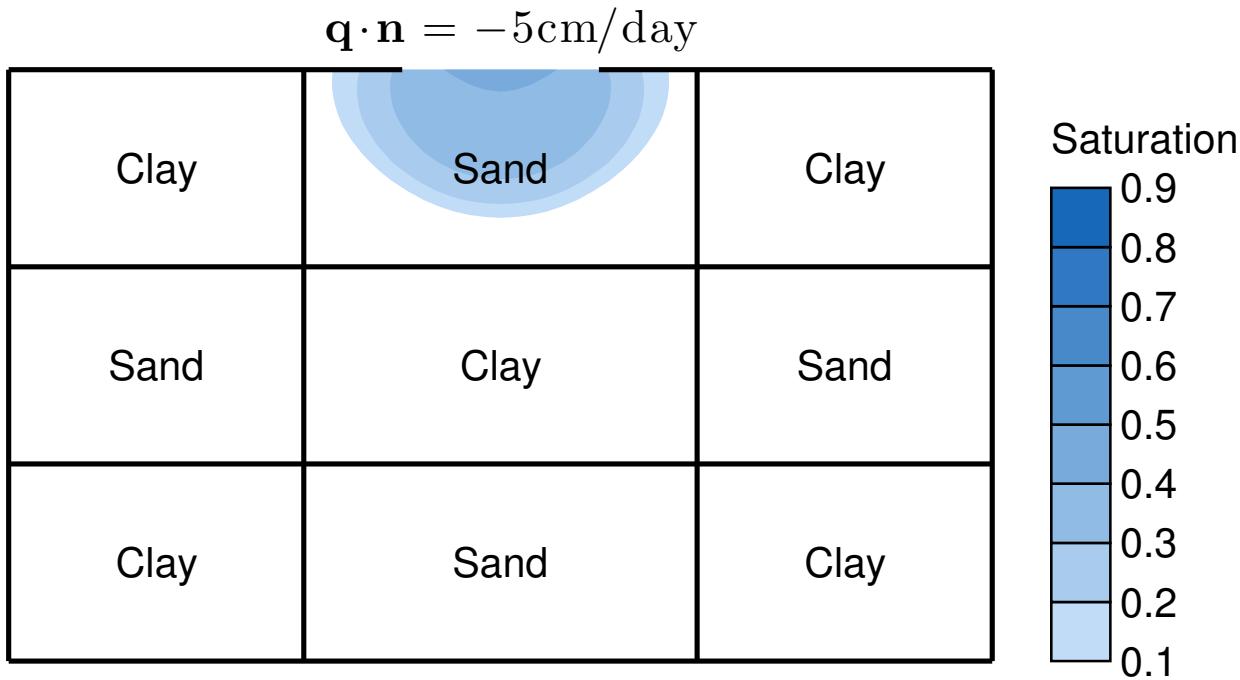
Test Problem: Water infiltration



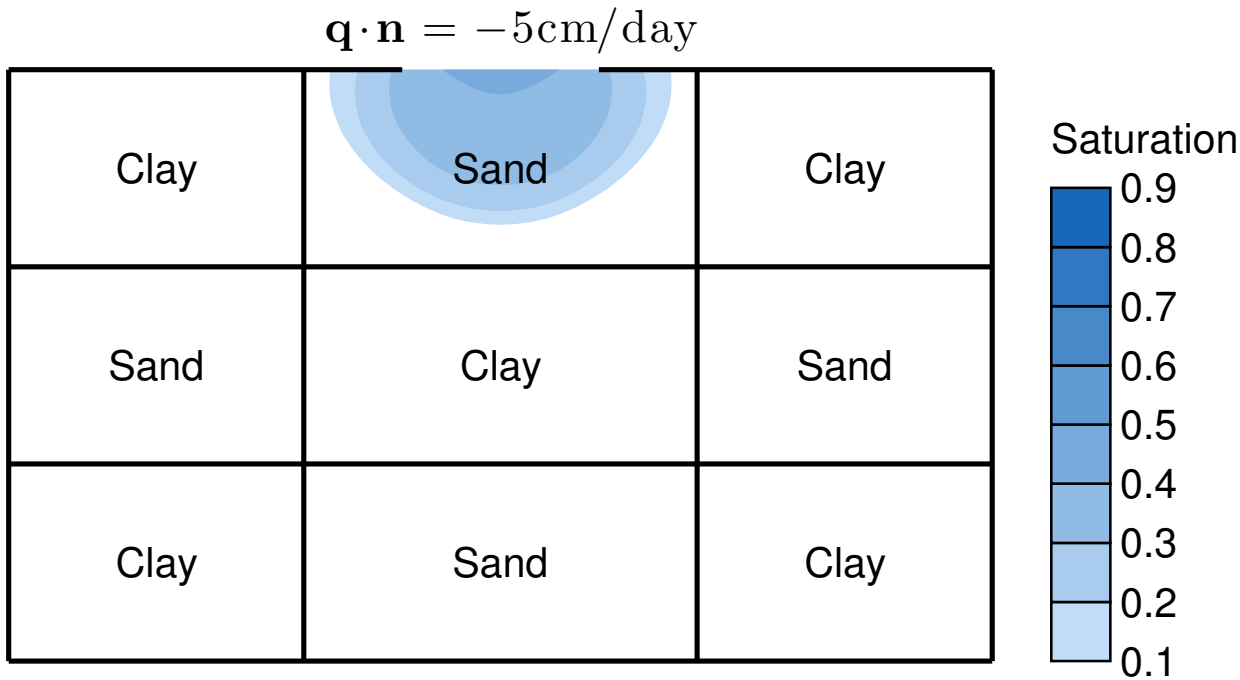
Test Problem: Water infiltration



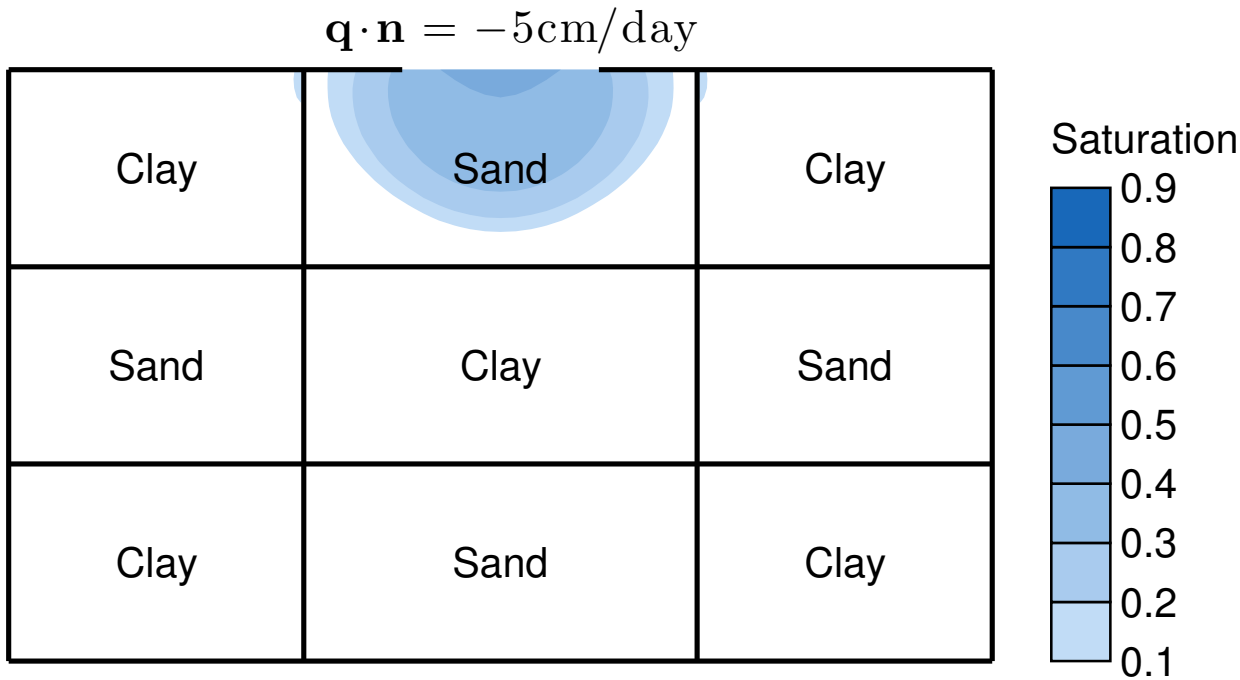
Test Problem: Water infiltration



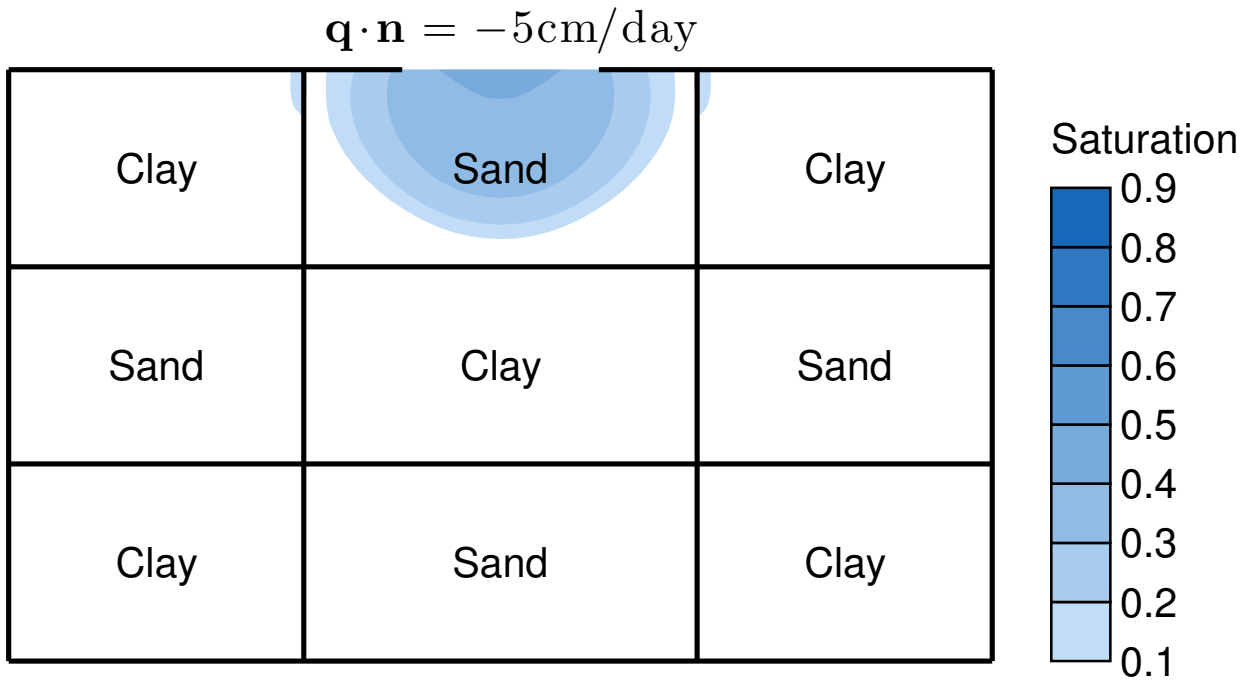
Test Problem: Water infiltration



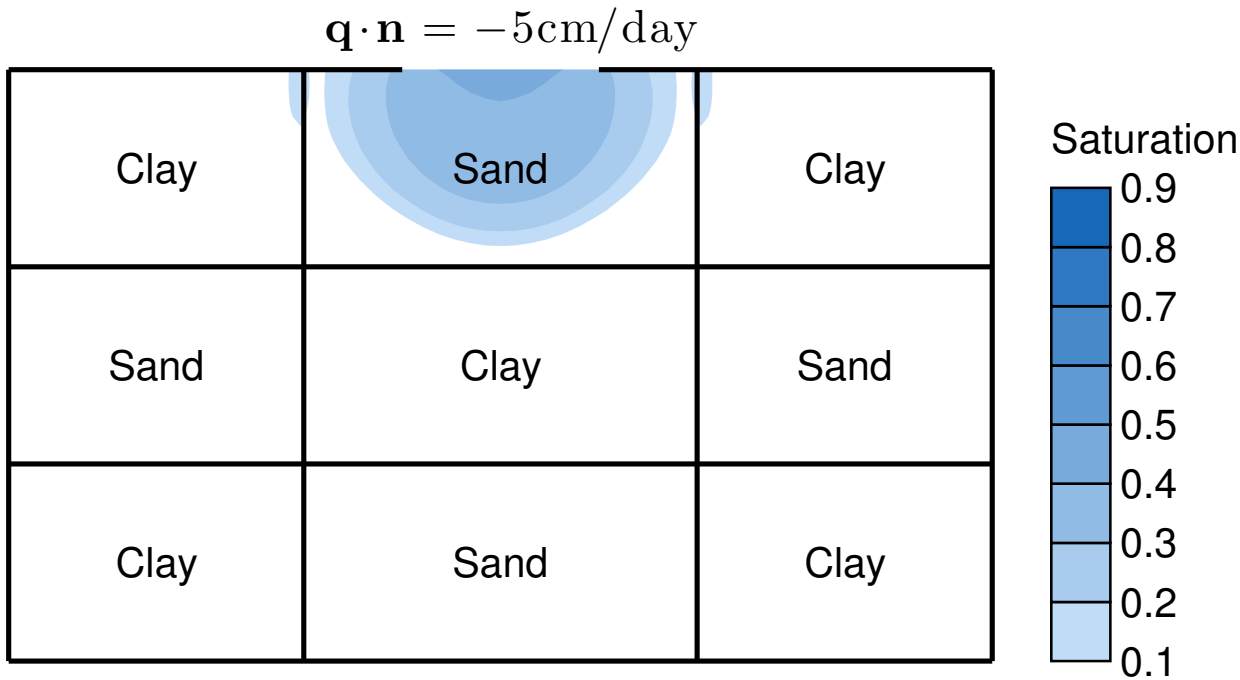
Test Problem: Water infiltration



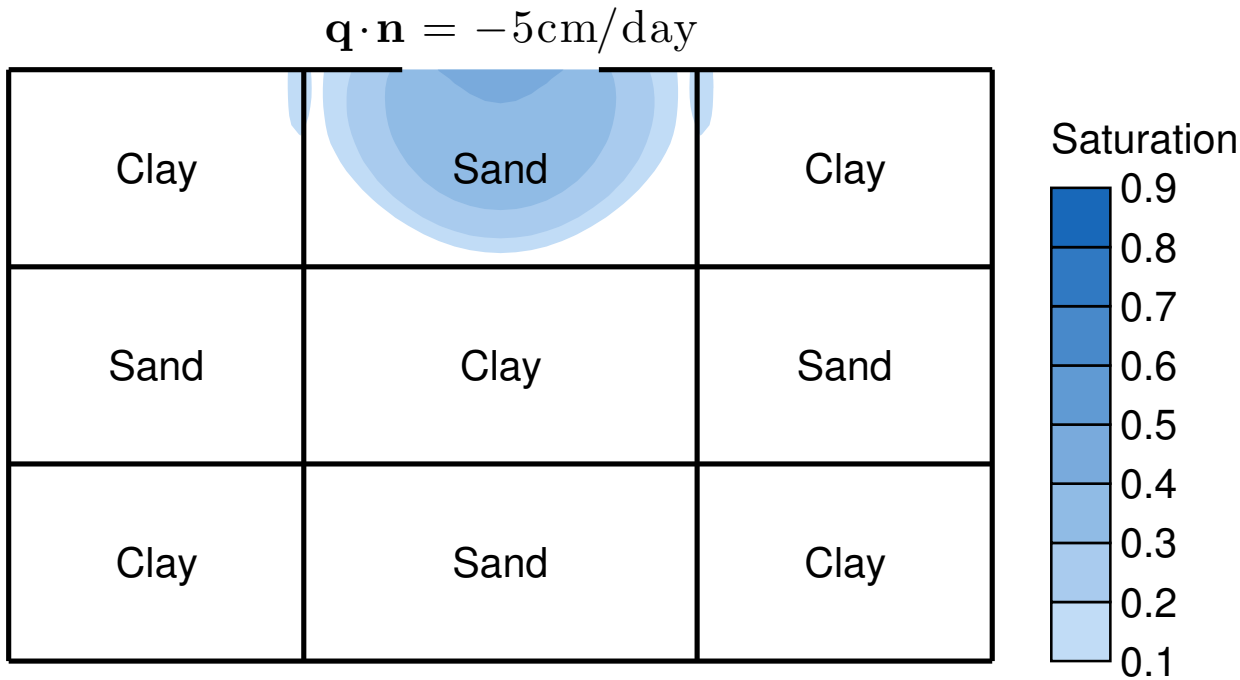
Test Problem: Water infiltration



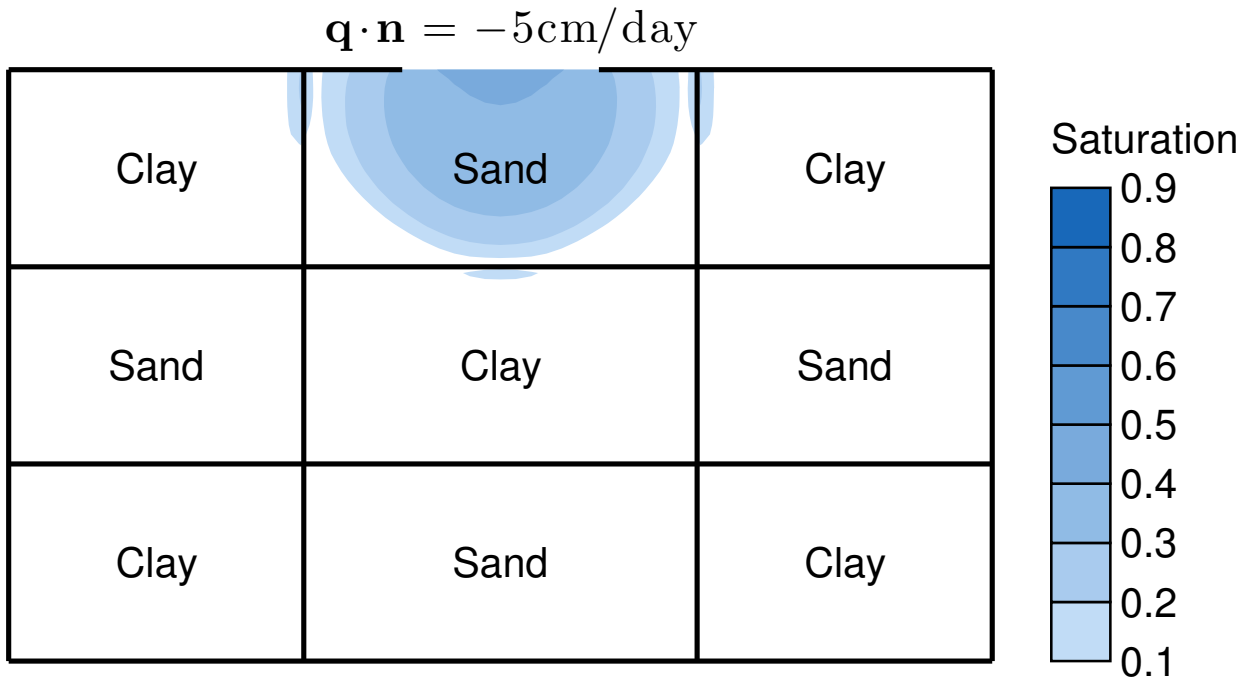
Test Problem: Water infiltration



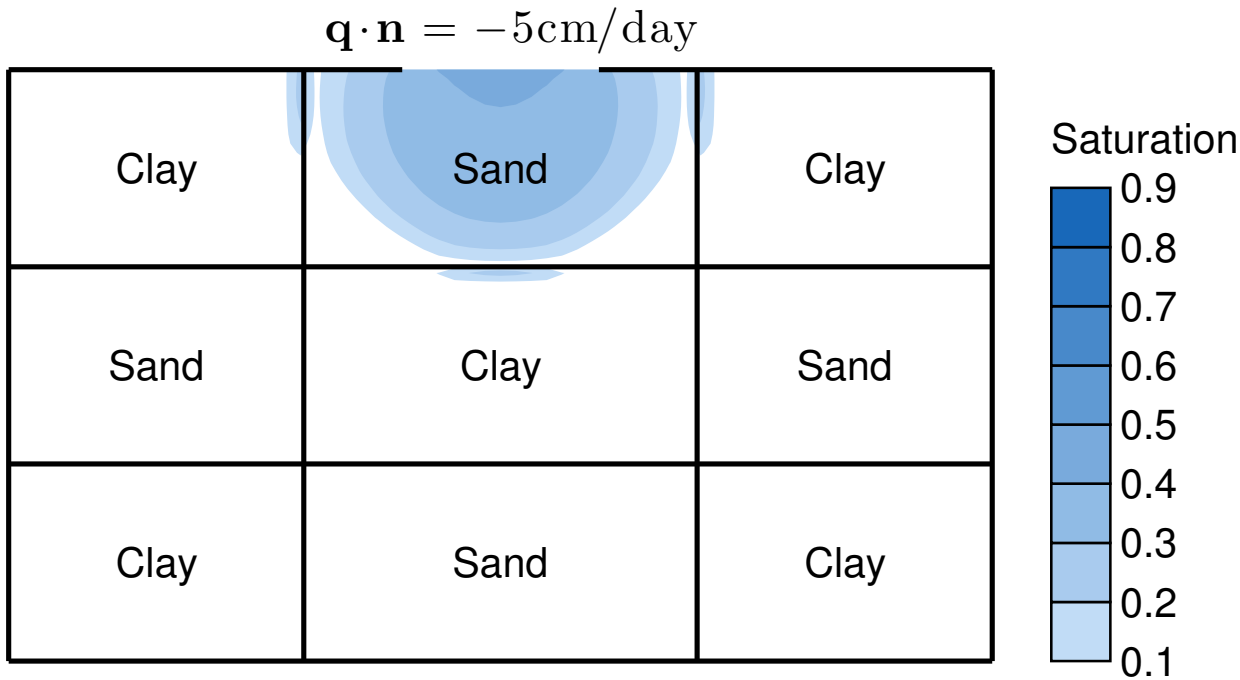
Test Problem: Water infiltration



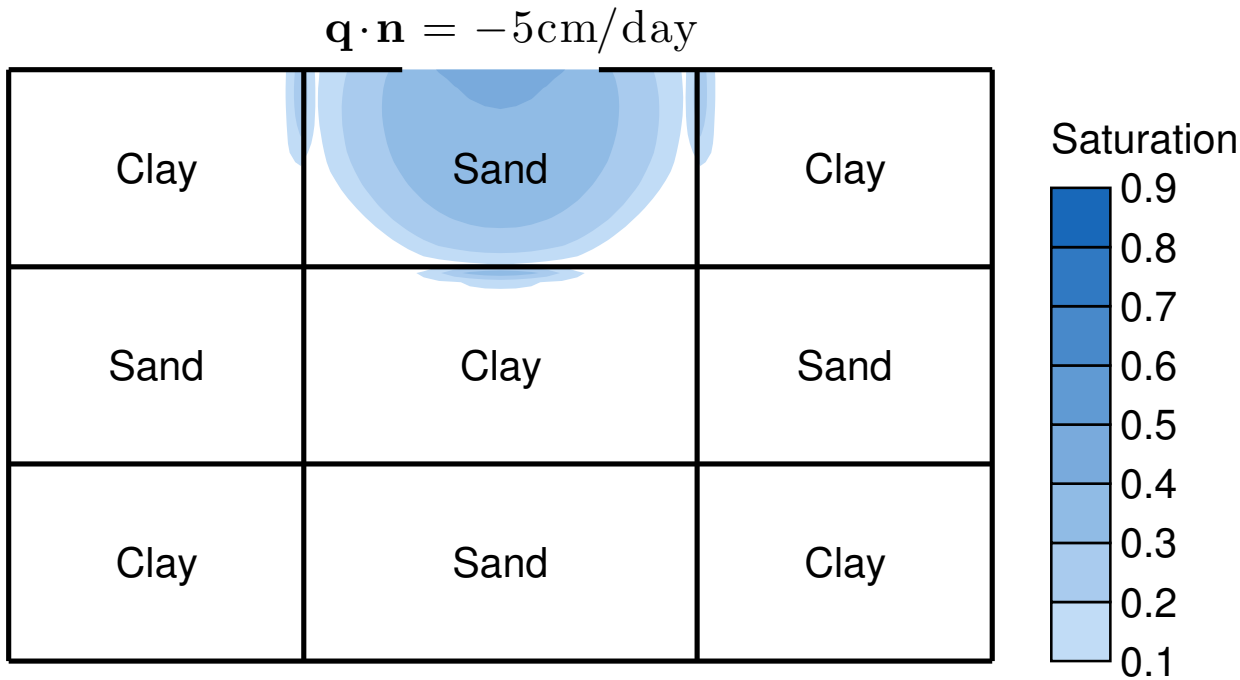
Test Problem: Water infiltration



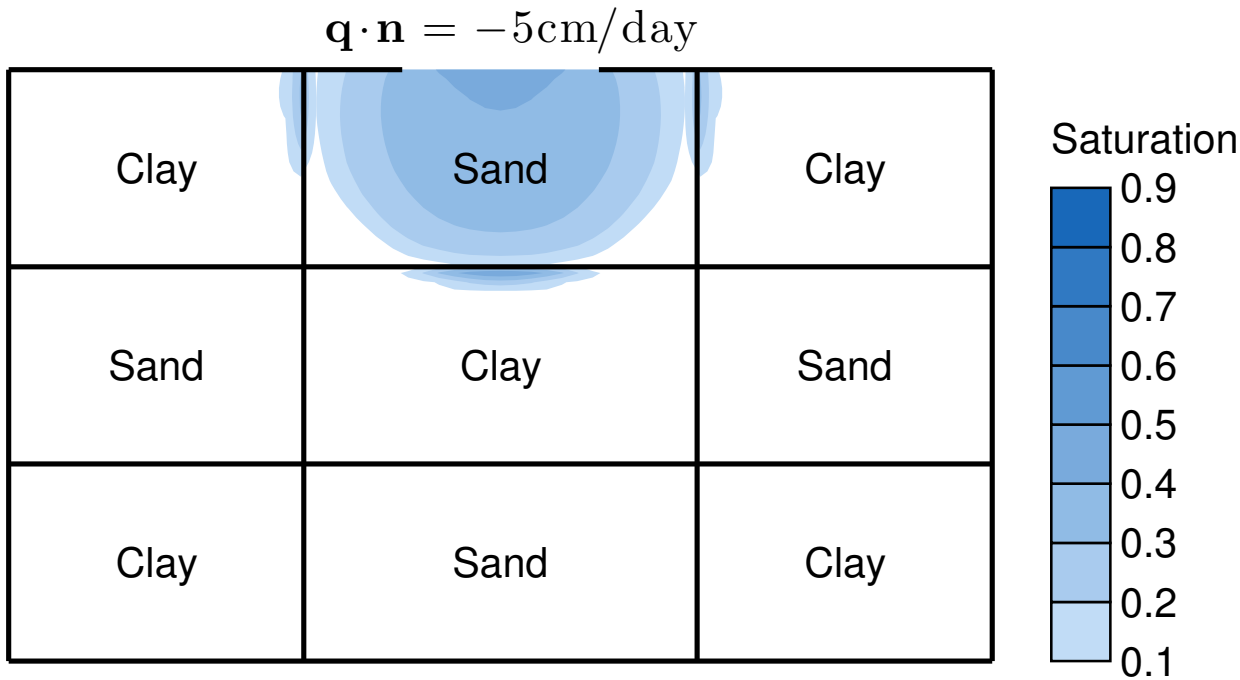
Test Problem: Water infiltration



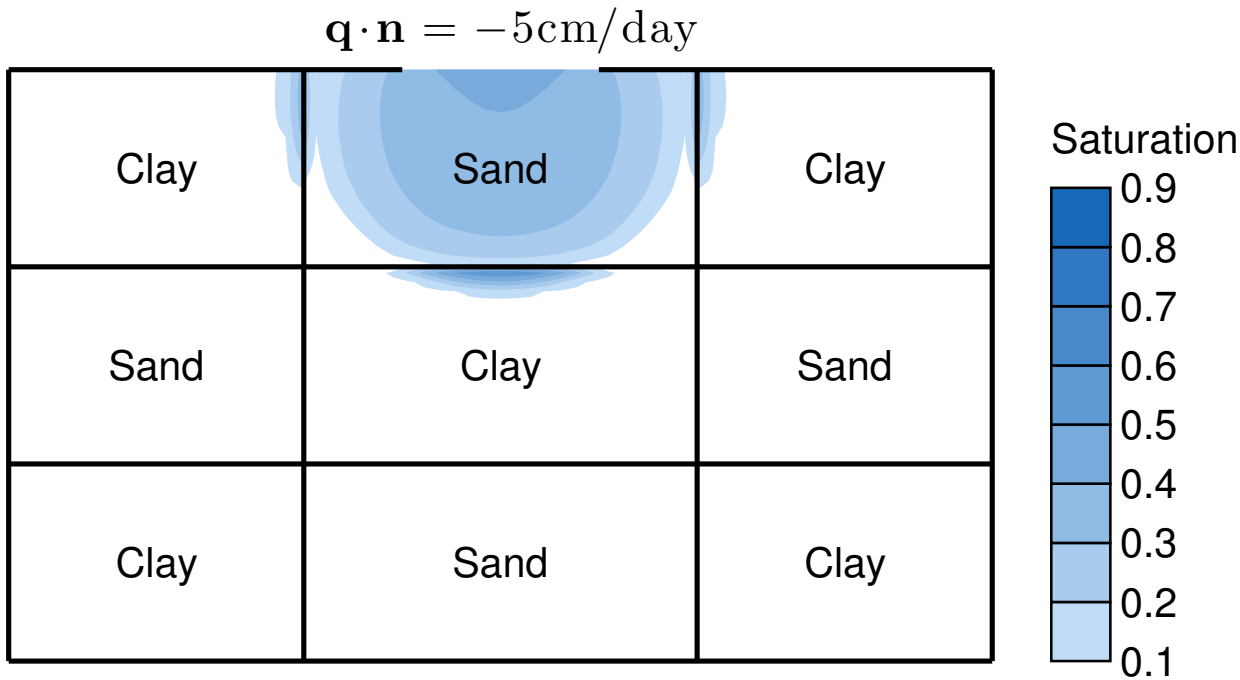
Test Problem: Water infiltration



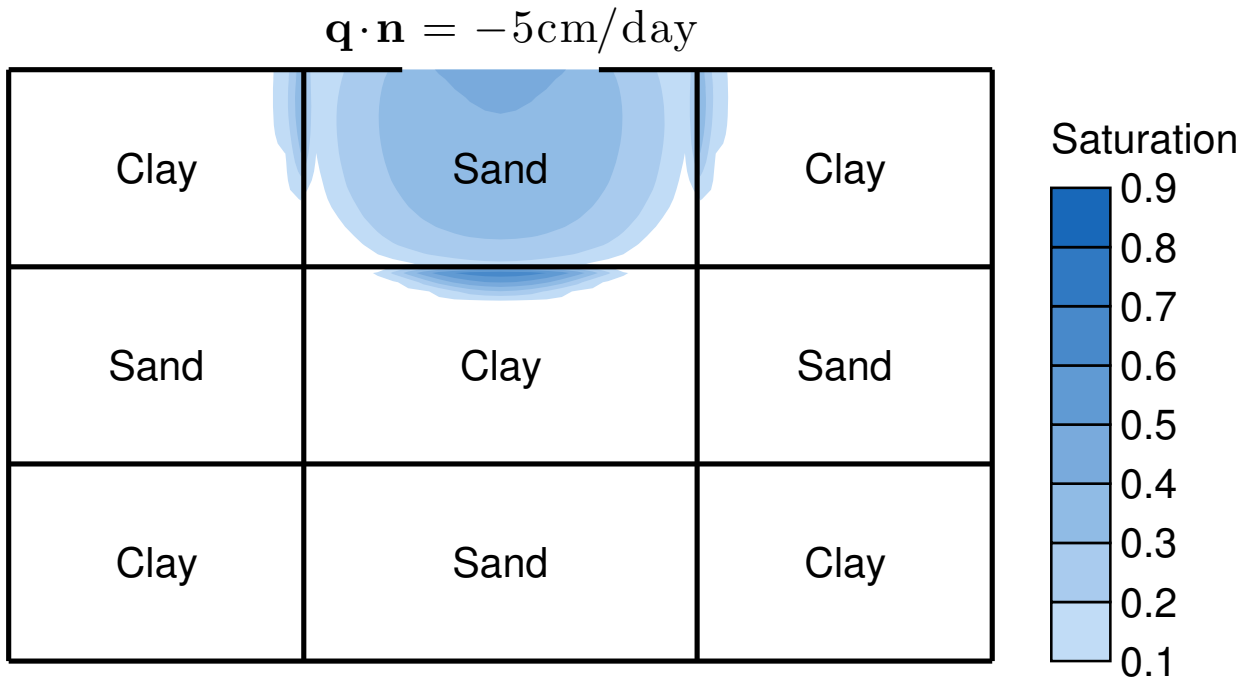
Test Problem: Water infiltration



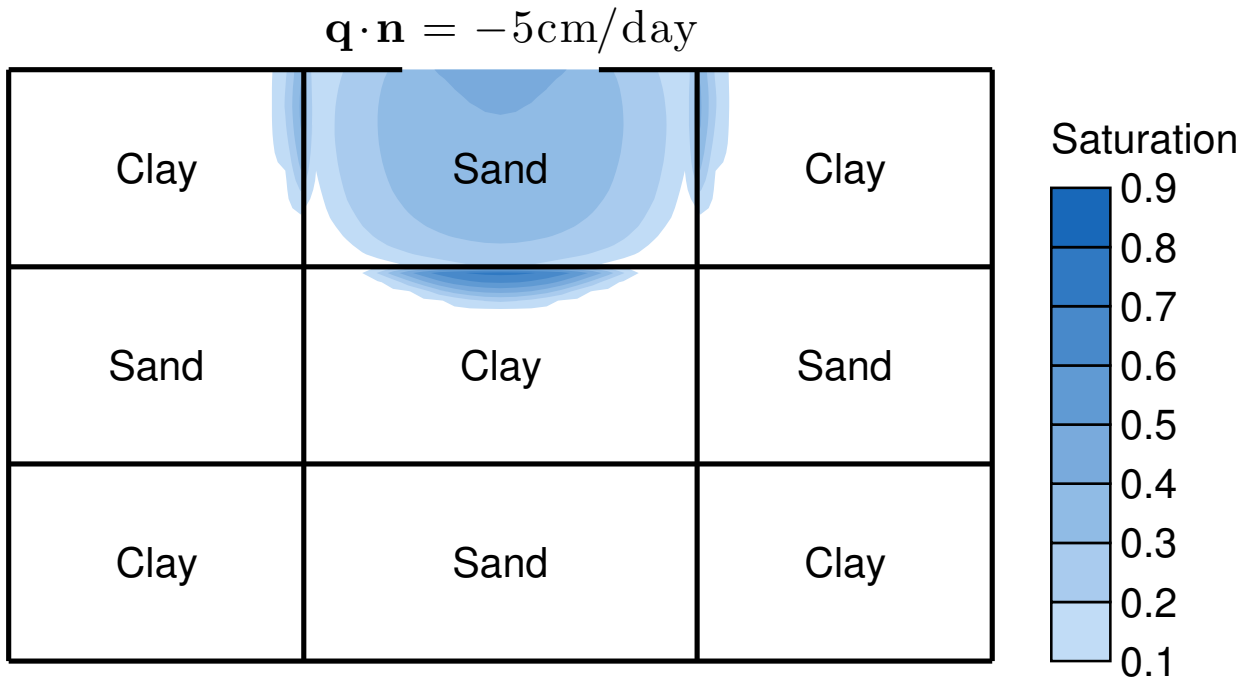
Test Problem: Water infiltration



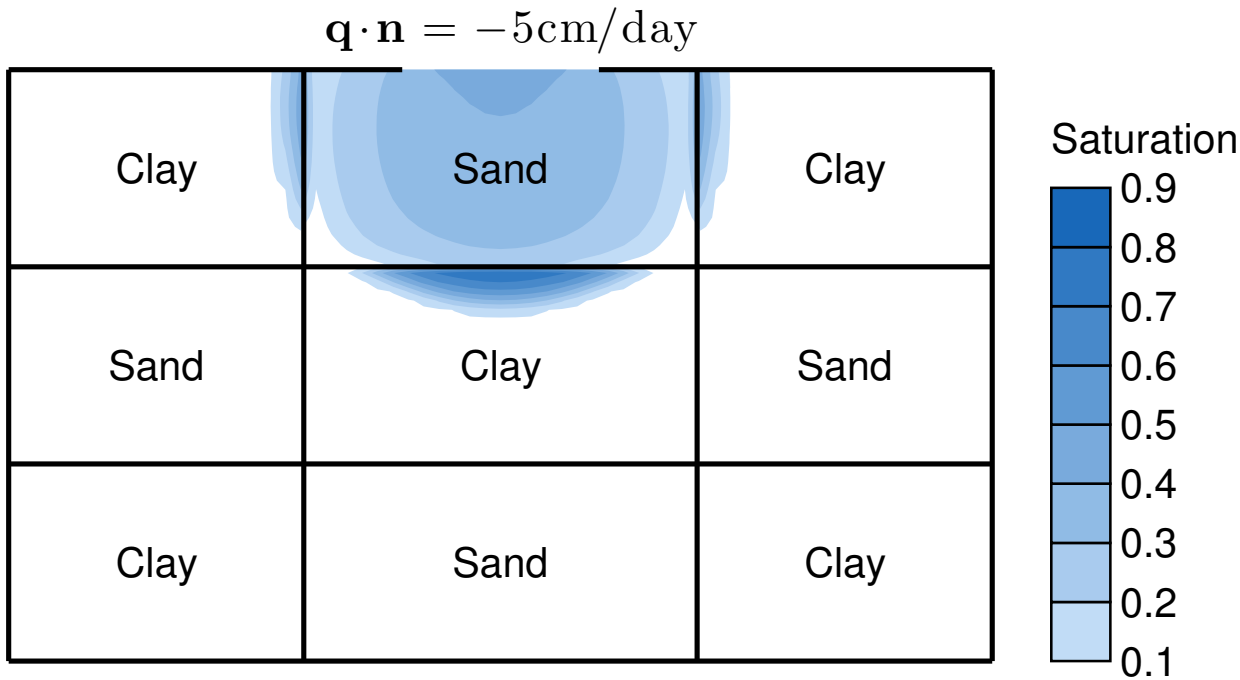
Test Problem: Water infiltration



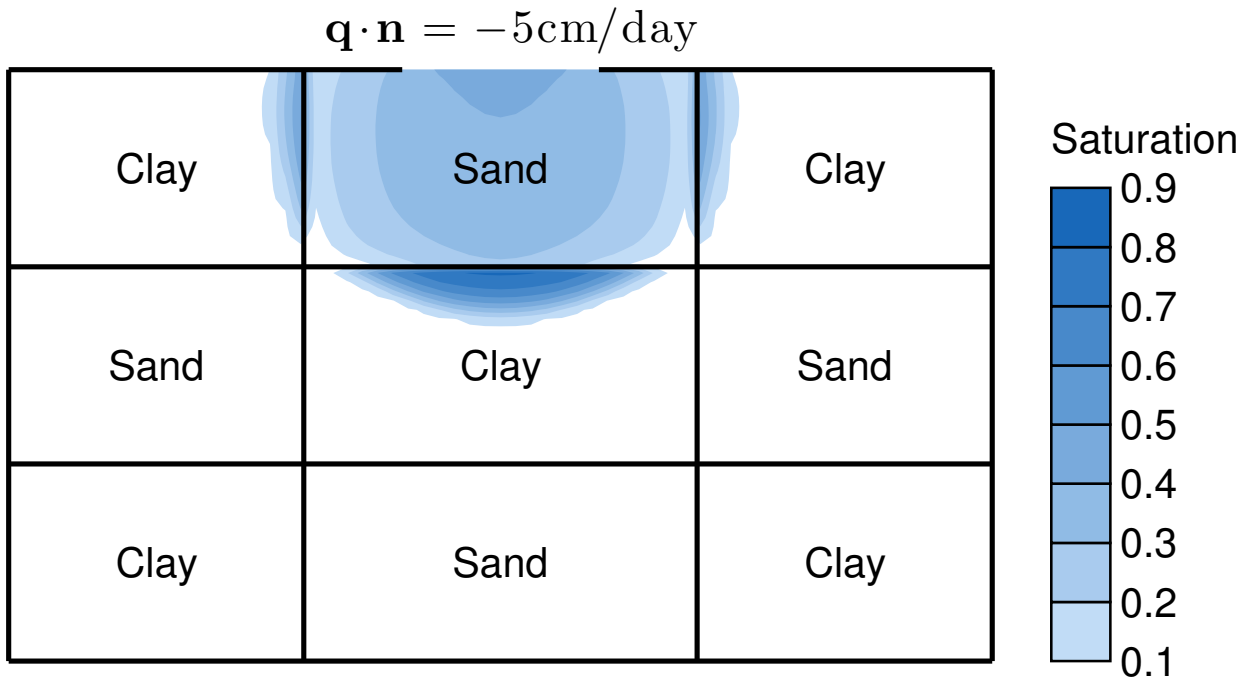
Test Problem: Water infiltration



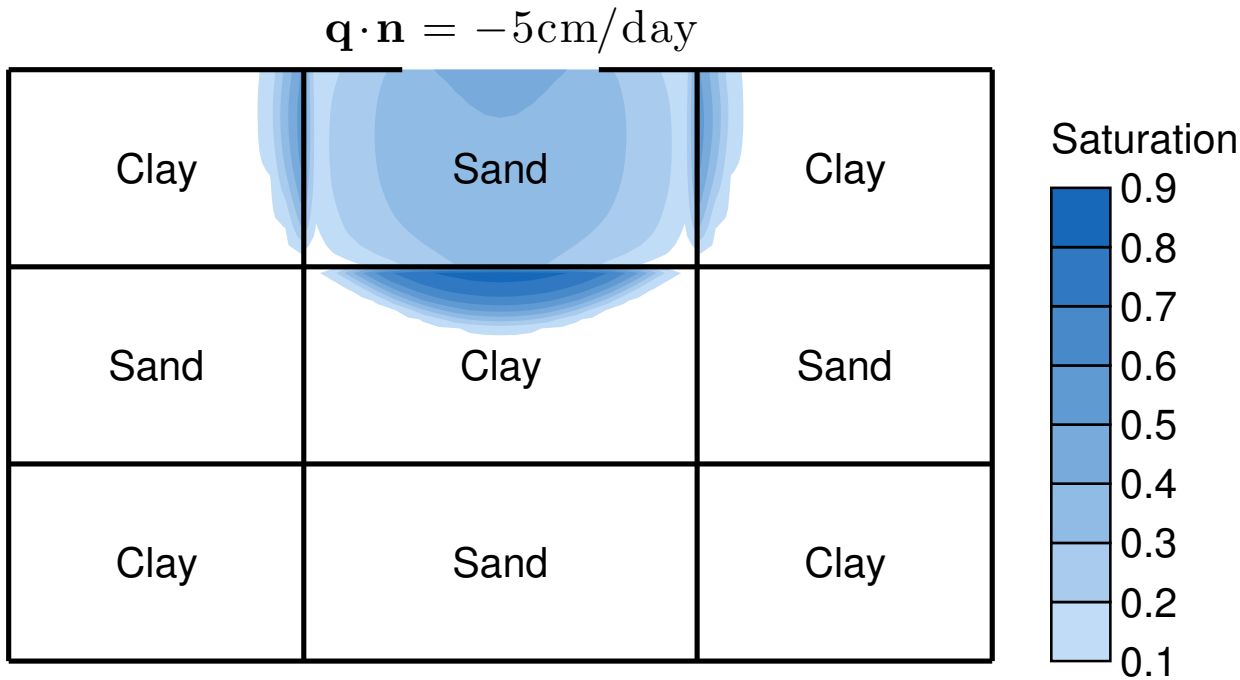
Test Problem: Water infiltration



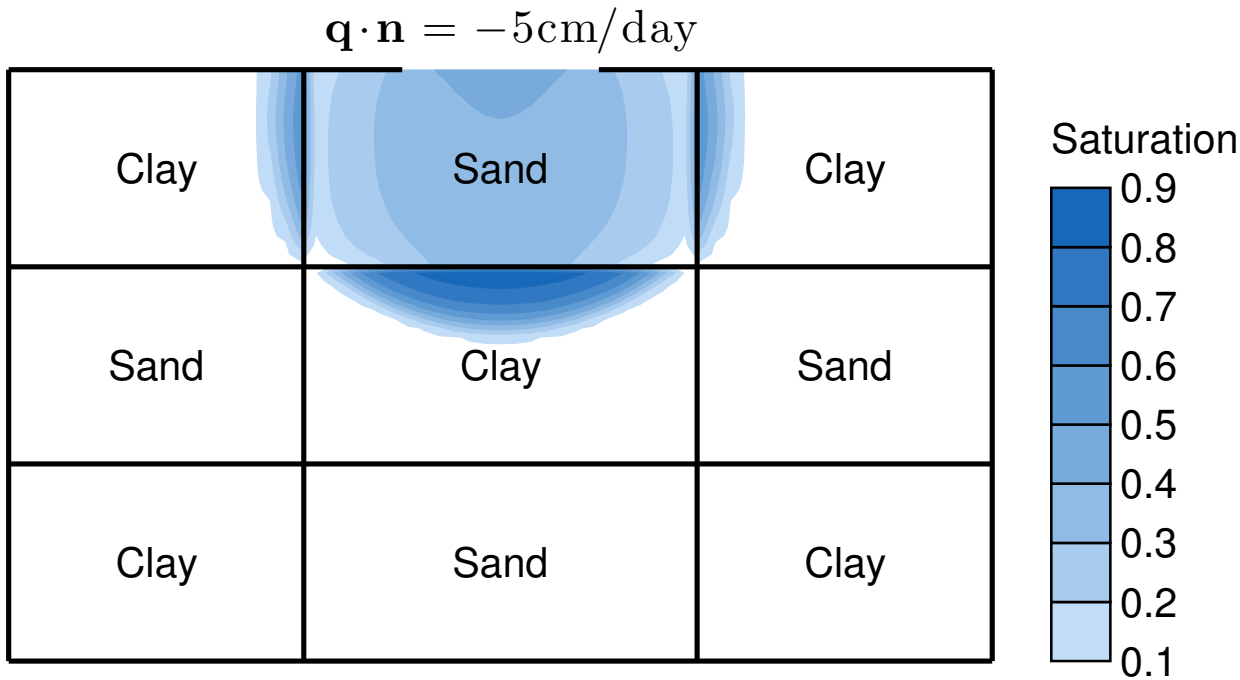
Test Problem: Water infiltration



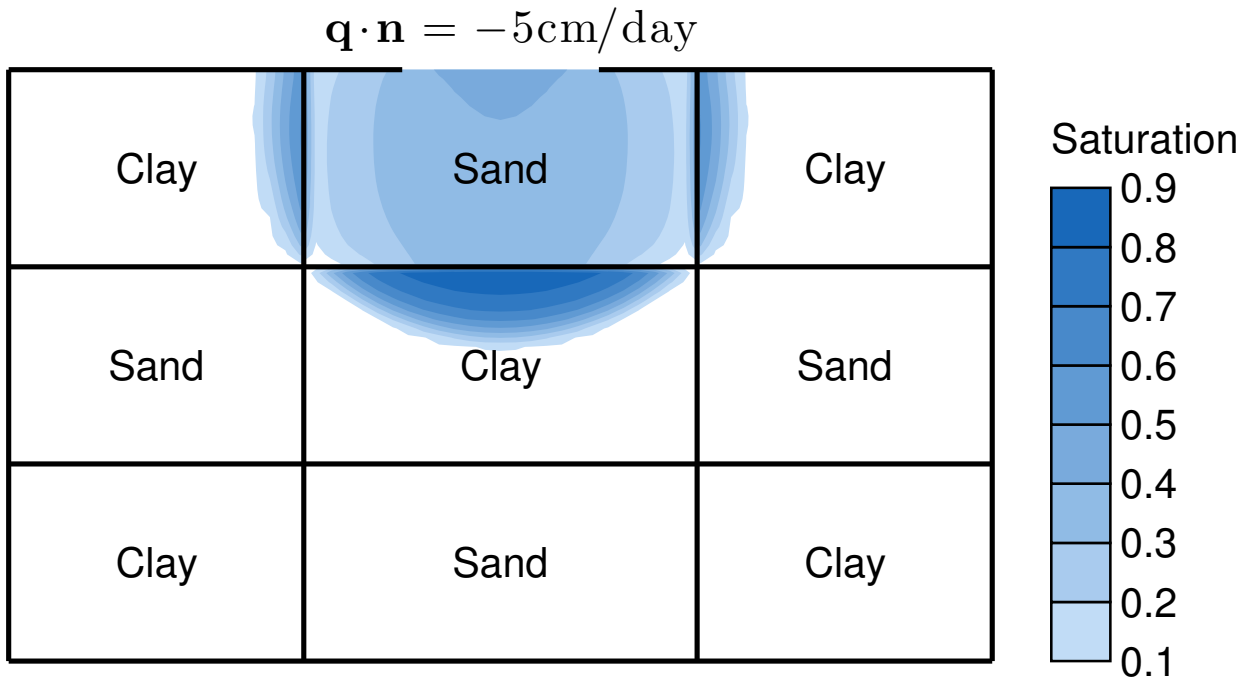
Test Problem: Water infiltration



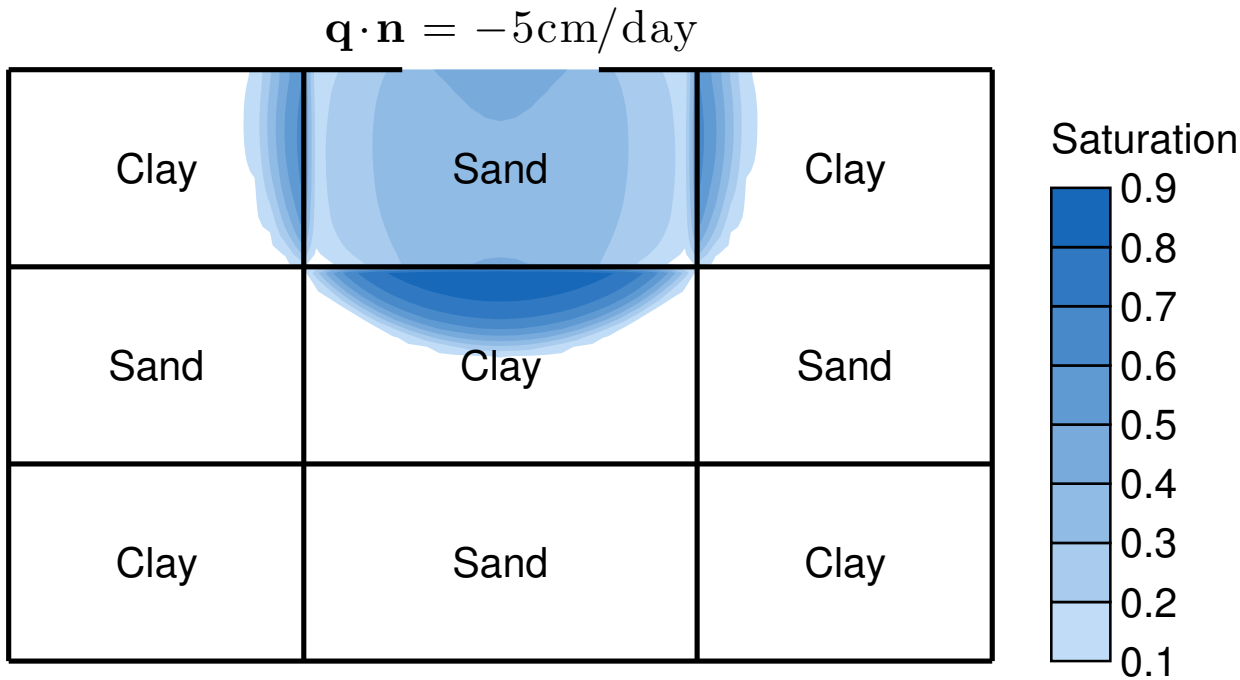
Test Problem: Water infiltration



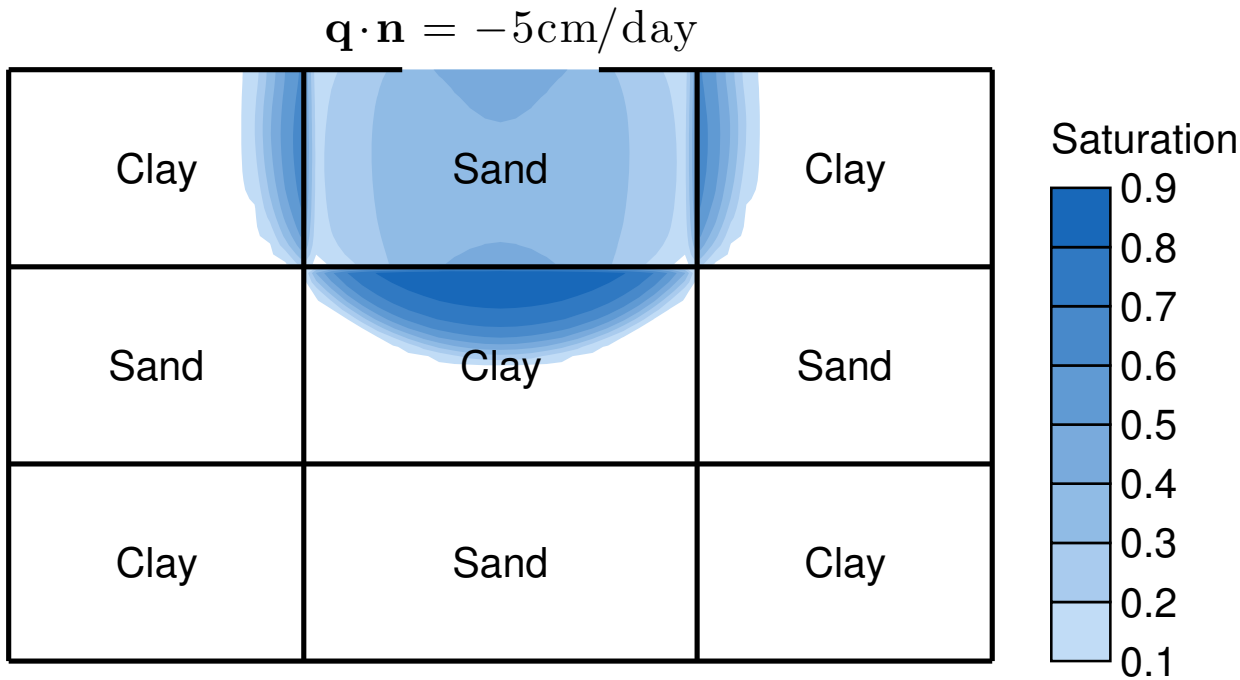
Test Problem: Water infiltration



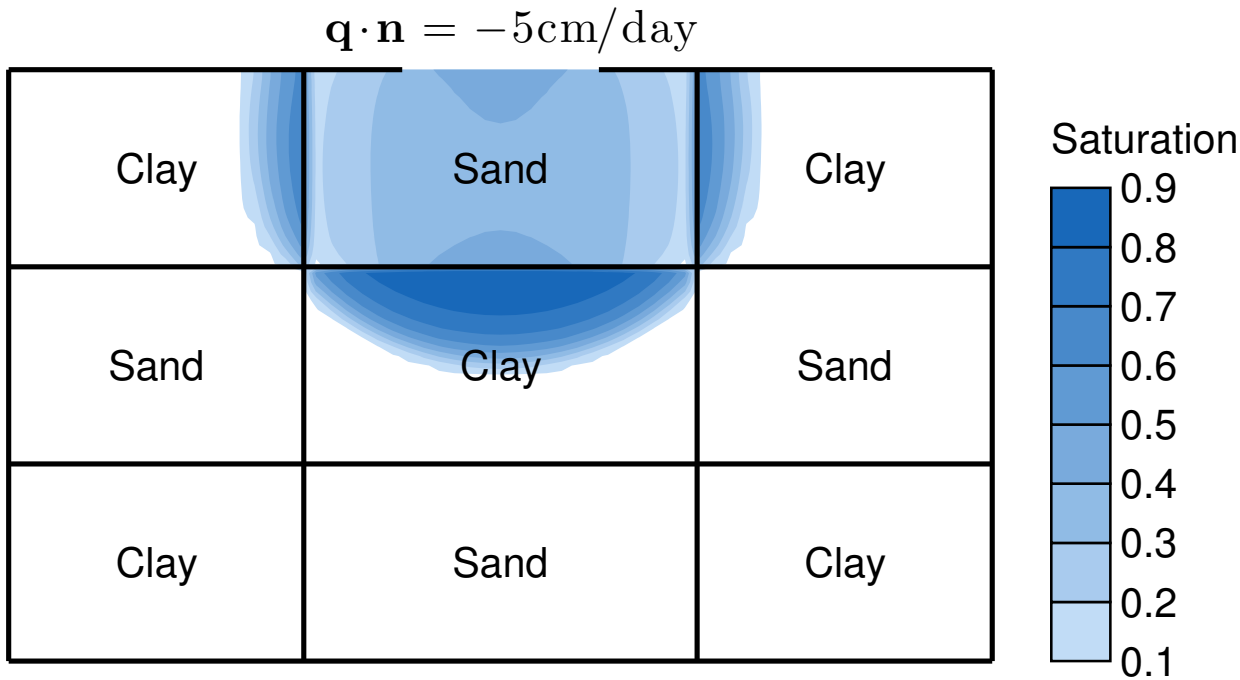
Test Problem: Water infiltration



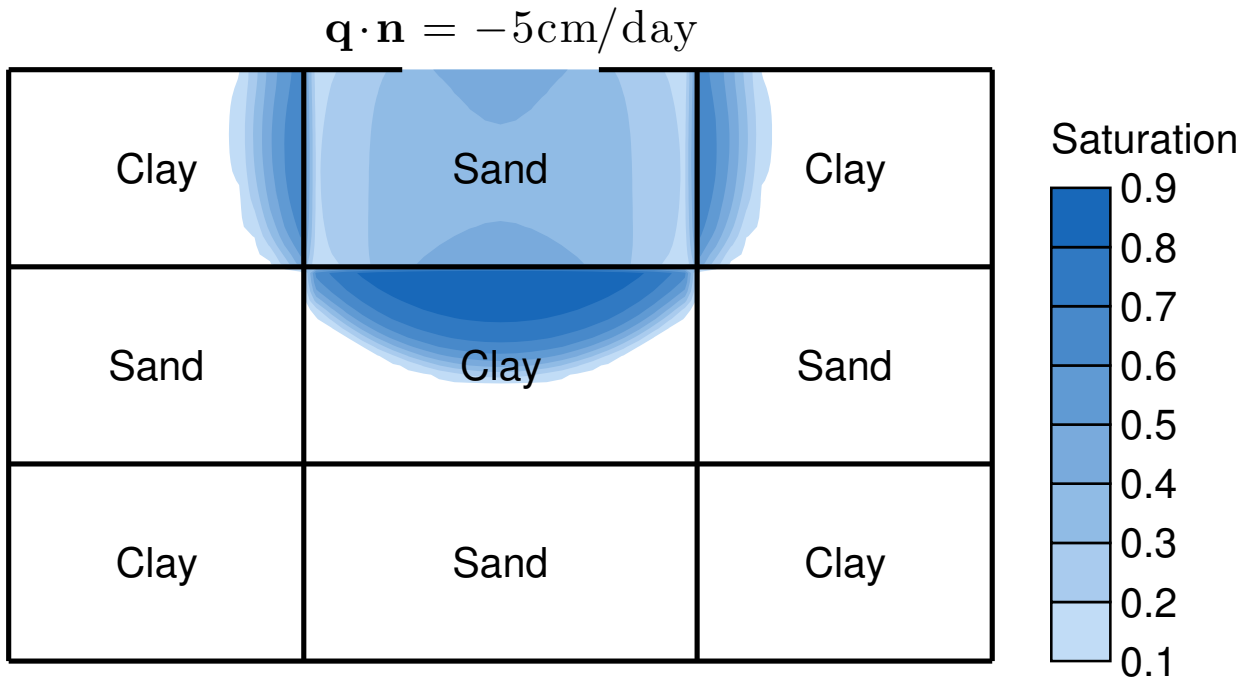
Test Problem: Water infiltration



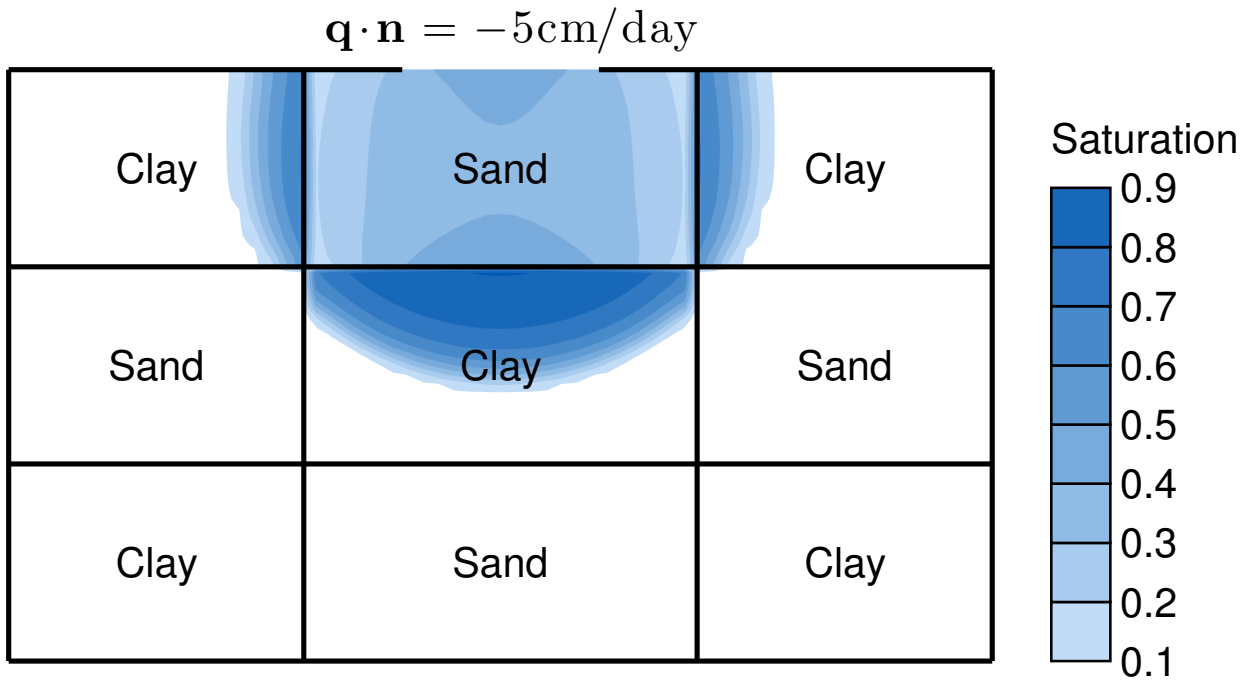
Test Problem: Water infiltration



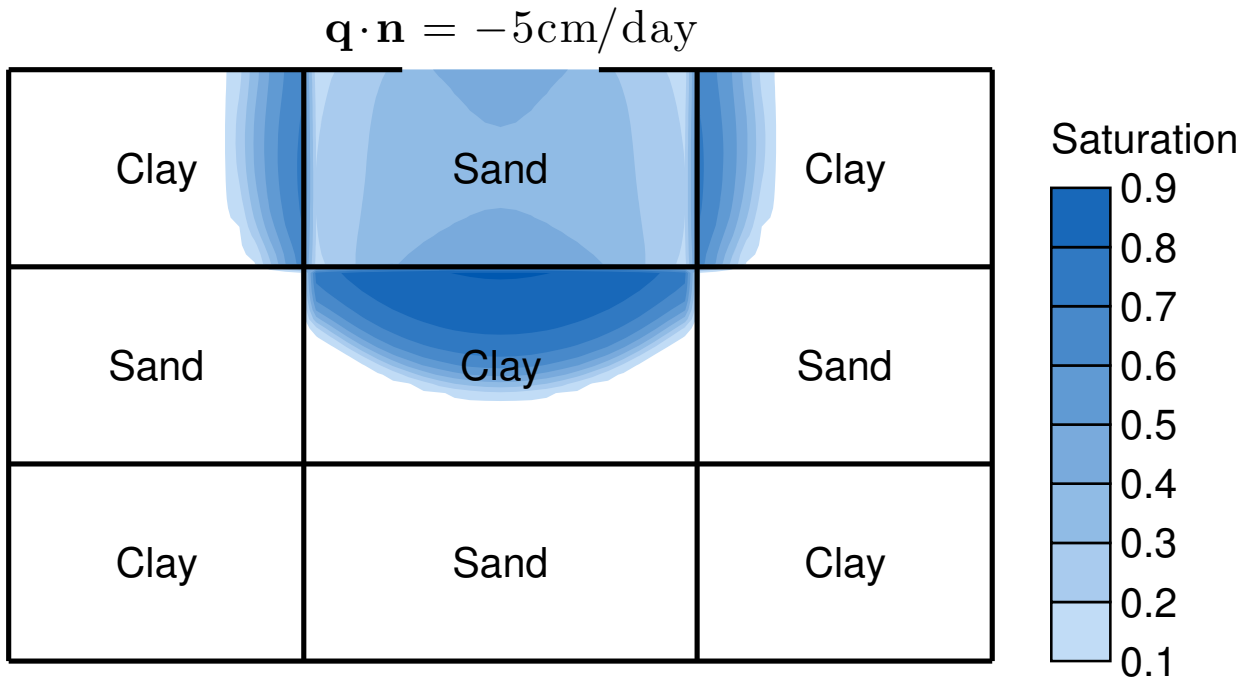
Test Problem: Water infiltration



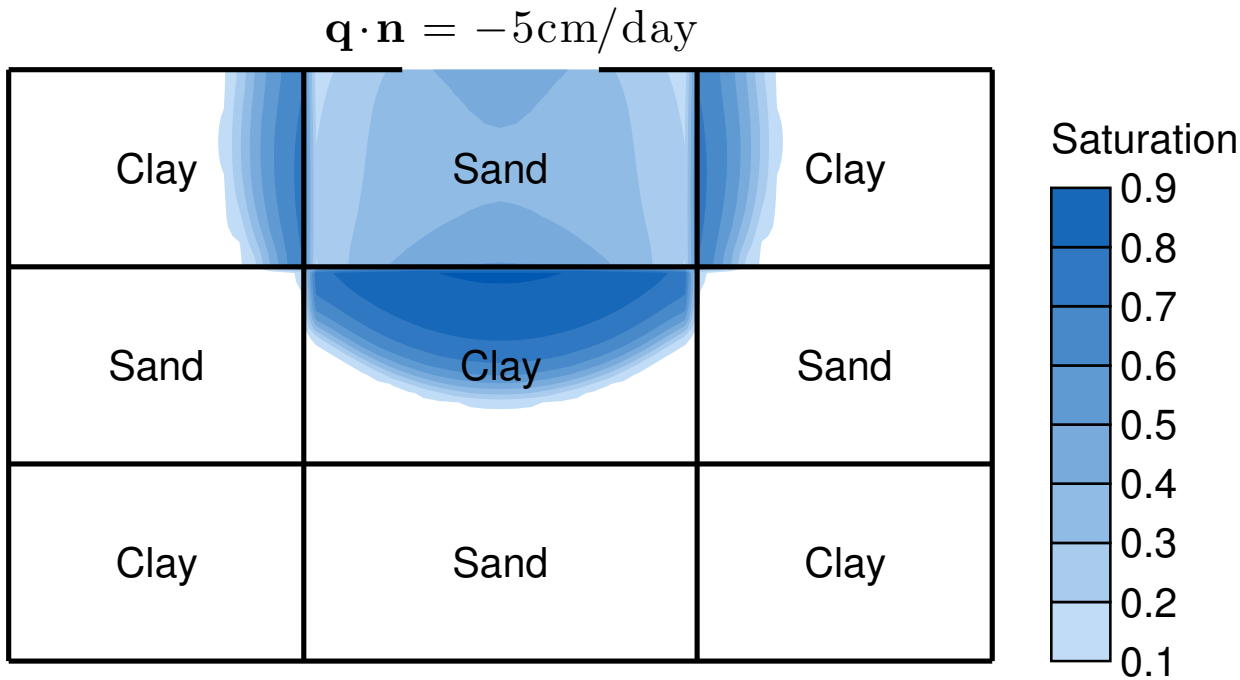
Test Problem: Water infiltration



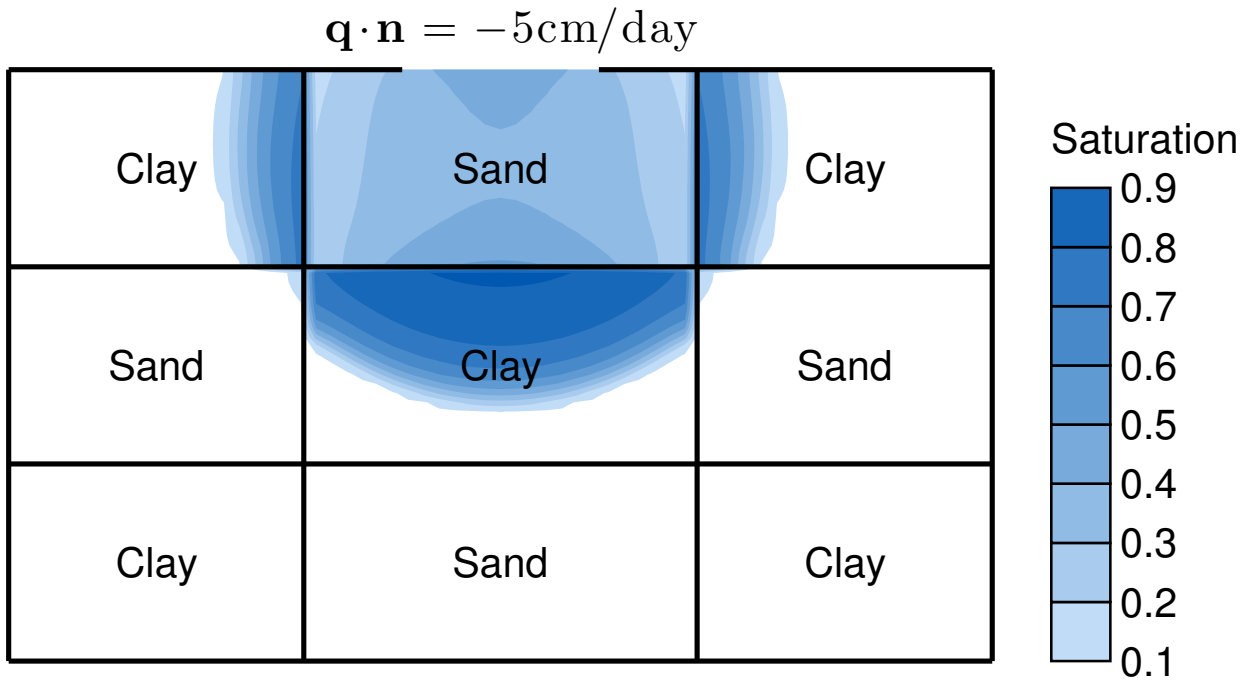
Test Problem: Water infiltration



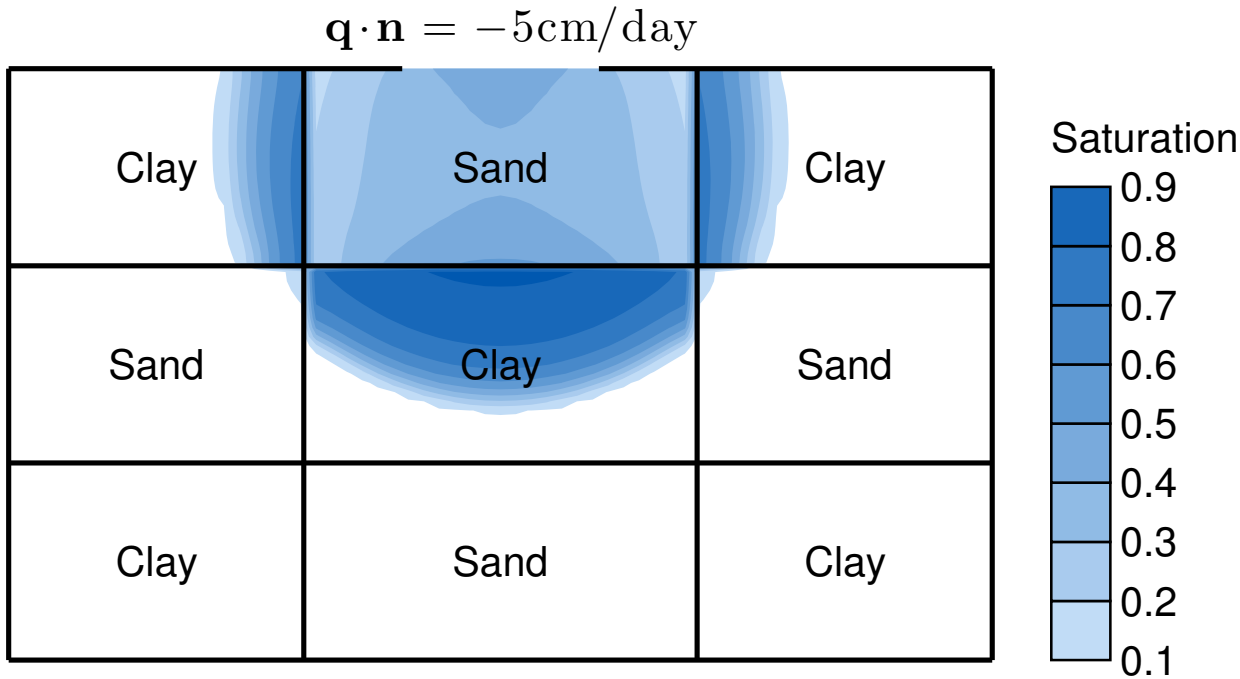
Test Problem: Water infiltration



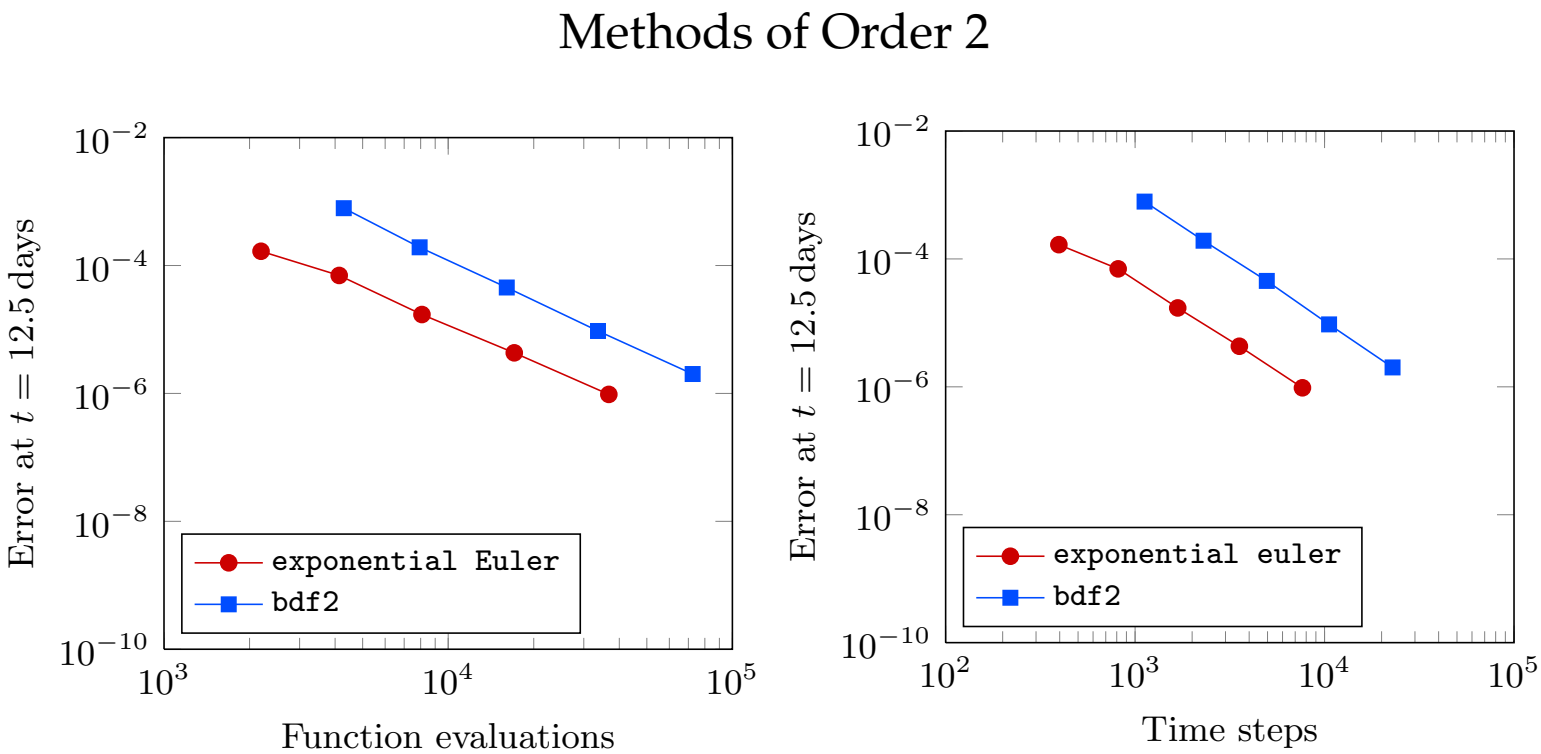
Test Problem: Water infiltration



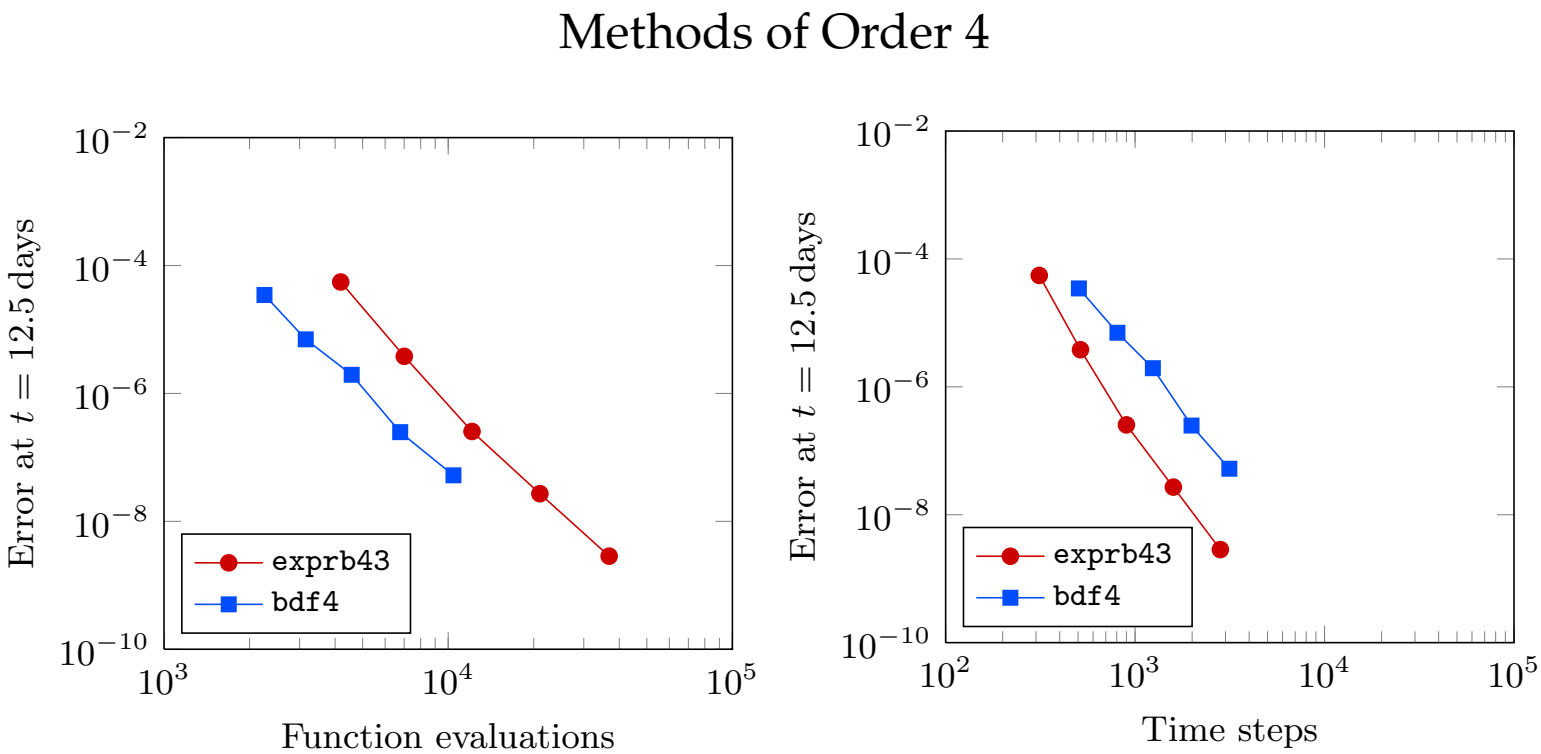
Test Problem: Water infiltration



Test Case: Unsaturated water flow



Test Case: Unsaturated water flow



Summary and Conclusions

Pros and Cons of exponential integrators:

- ✓ Krylov subspace methods converge rapidly
- ✓ Implemented efficiently without preconditioning $\Rightarrow$ Jacobian free
- ✓ Good stability properties $\Rightarrow$ Large time steps
- ✗ Spatial discretisation must be expressed in the form of

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}(\boldsymbol{u}), \quad \boldsymbol{u}(0) = \boldsymbol{u}_0$$

- ✗ Performance declines in terms of function evaluations as we move up in order

Open Problem: Higher order methods require multiple matrix-function vector products with the same Jacobian matrix. How to recycle Krylov subspace information from one subspace to the next?

Overview of Presentation

- Modelling of unsaturated water flow
- Time integration methods for unsaturated water flow: Exponential integrators
- Two-scale numerical modelling of unsaturated water flow in soils exhibiting small-scale heterogeneities

Problem Statement

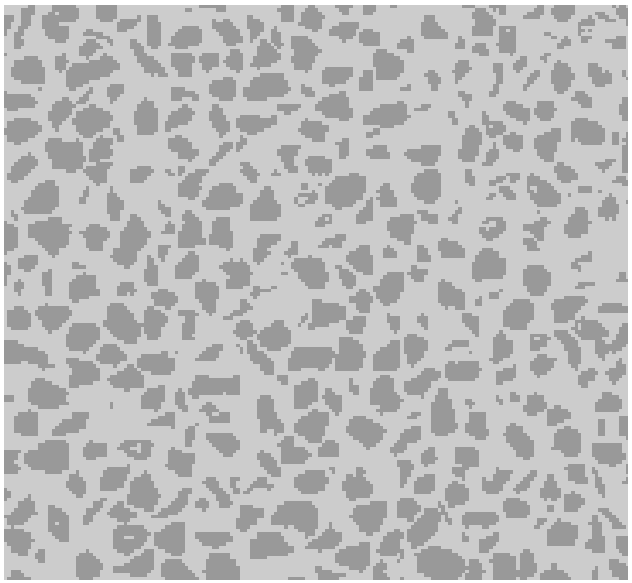
- Richards' equation in two-dimensions

$$\frac{\partial}{\partial t}\theta(h) + \nabla \cdot [-K(h)\nabla (h + x_2)] = 0$$

- Domain comprised of two sub-domains Ω_a (connected) and Ω_b (disconnected) with different hydraulic properties:

$$\theta(h) = \begin{cases} \theta_a(h) & \text{in } \Omega_a \\ \theta_b(h) & \text{in } \Omega_b \end{cases}$$

$$K(h) = \begin{cases} K_a(h) & \text{in } \Omega_a \\ K_b(h) & \text{in } \Omega_b \end{cases}$$



Heterogeneous domain

Ω_a Ω_b

- Computational cost of direct numerical simulation is prohibitively expensive when the domain exhibits small-scale heterogeneity.

Two-scale Model for unsaturated flow

- Macroscopic equation ($x \in \Omega$)

$$\frac{\partial}{\partial t}\theta_{\text{eff}}(h_a) + \nabla_x \cdot [-\boldsymbol{K}_{\text{eff}}(h_a)\nabla_x (h_a + x_2)] = Q$$

- Microscopic equation ($y \in \Omega_{x,b}$)

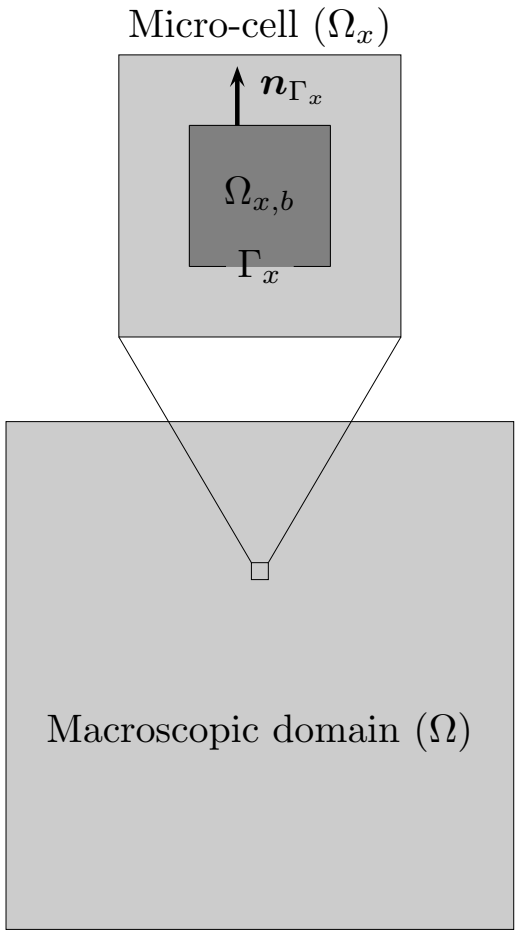
$$\frac{\partial}{\partial t}\theta_b(h_b) + \nabla_y \cdot [-K_b(h_b)\nabla_y (h_b + y_2)] = 0$$

- Microscopic boundary condition ($y \in \Gamma_x$)

$$h_a = h_b$$

- Source term

$$Q = \frac{1}{|\Omega_x|}\int_{\Gamma_x} -K_b(h)\nabla_y (h_b + y_2) \cdot \boldsymbol{n}_{\Gamma_x}\, ds$$



See, e.g., Szymkiewicz and Lewandowska (2006).

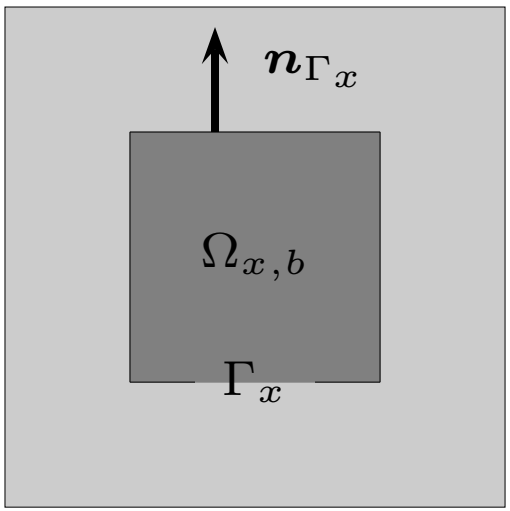
Effective hydraulic conductivity

- $(\boldsymbol{K}_{\text{eff}})_{*,j}$ is a matrix with jth column equal to

$$(\boldsymbol{K}_{\text{eff}})_{*,j} = \frac{1}{|\Omega_x|} \int_{\Omega_{x,a}} K_a(h_a) \nabla_y (u + y_2) \, dA$$

where u is a solution of the periodic cell-problem:

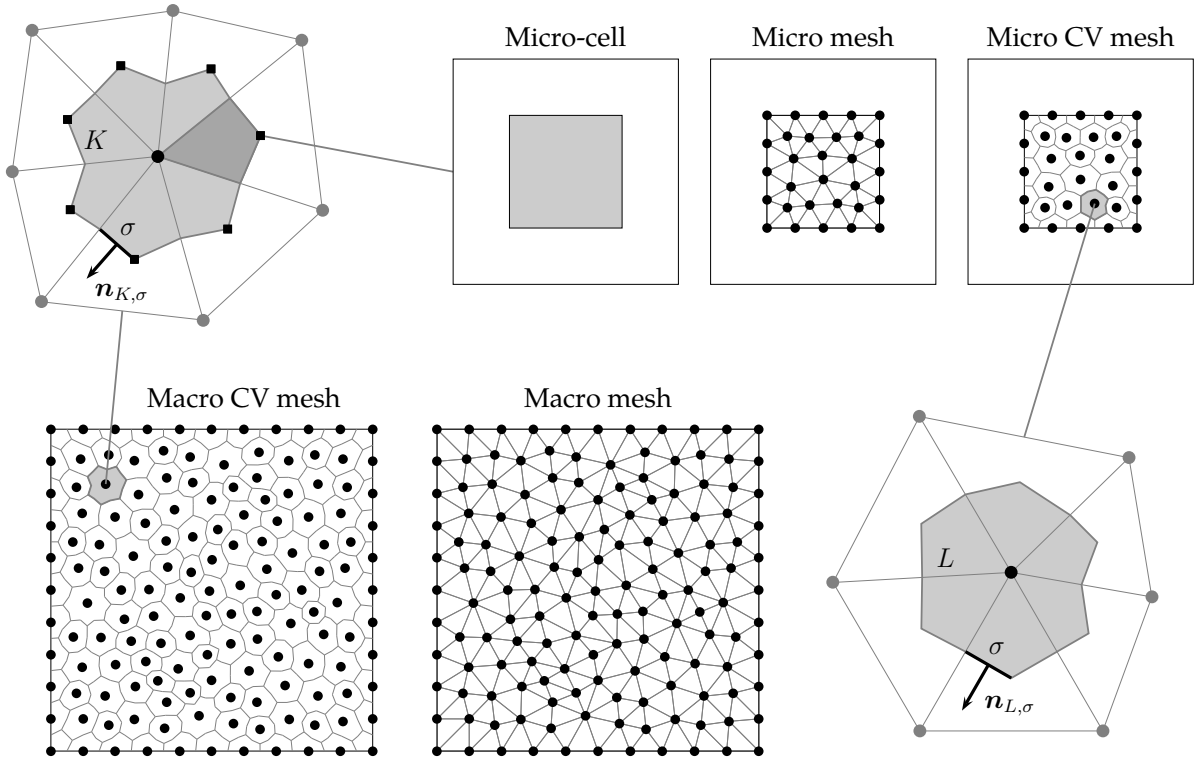
$$\begin{cases} \nabla_y \cdot (K_a(h_a) \nabla_y (u + y_2)) = 0 & y \in \Omega_{x,a} \\ K_a(h_a) \nabla_y (u + y_2) \cdot \boldsymbol{n}_{\Gamma_x} = 0 & y \in \Gamma_x \\ u \text{ is } \Omega_x\text{-periodic} \end{cases}$$



- Pre-processing step:
  (a) Compute entries of $\boldsymbol{K}_{\text{eff}}$ using a range of values of h_a
  (b) Use interpolation to evaluate $\boldsymbol{K}_{\text{eff}}$ at any value of h_a

Unstructured CV spatial discretisation

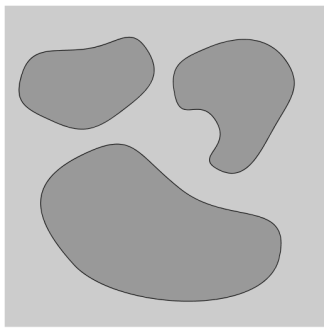
- Unstructured vertex-centered CV method [Carr and Turner (2014)]



- Existing numerical strategies (e.g., Szymkiewicz and Lewandowska (2008)) only applicable to simple inclusion geometries

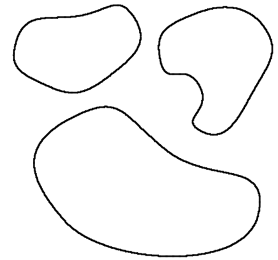
Meshing of micro-cells

Image-based meshing code built on GMSH [Guezaine and Remacle (2009)]

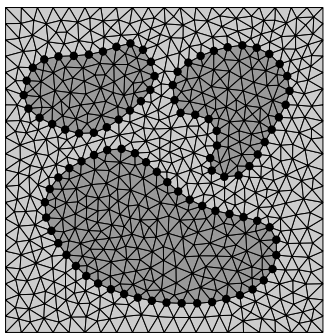


Bitmap image

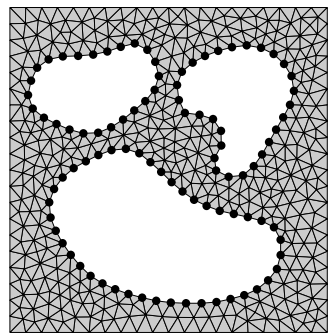
Binary image



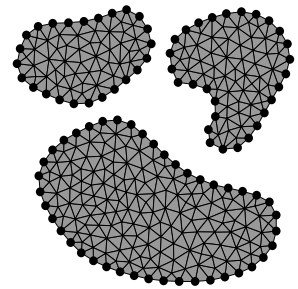
Edge detection



Full micro-cell



Ω_a only



Ω_b only

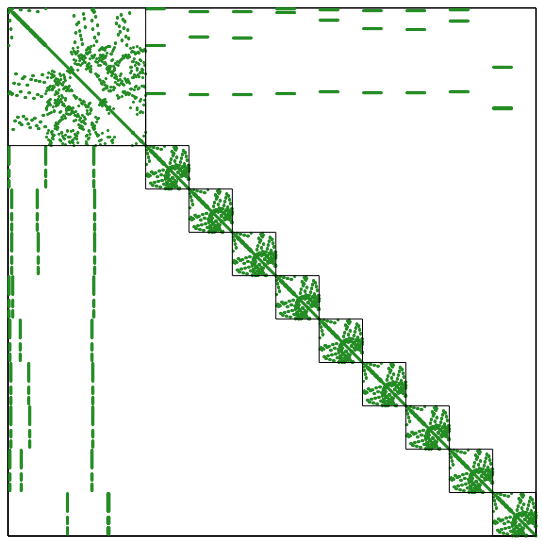
Spatially-discrete system of ODEs

- Spatial discretisation can be expressed in the form

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{g}(\boldsymbol{u}), \quad \boldsymbol{u}(0) = \boldsymbol{u}_0 \in \mathbb{R}^N,$$

  where $N = \text{num\_macro\_nodes} + \text{num\_macro\_elements} \times \text{num\_micro\_nodes}$.

- Example sparsity structure of Jacobian matrix (zoomed in)



$$\boldsymbol{J}(\boldsymbol{u})\boldsymbol{v} \approx [\boldsymbol{g}(\boldsymbol{u} + \varepsilon\boldsymbol{v}) - \boldsymbol{g}(\boldsymbol{u})]/\varepsilon$$

- Time integration: Jacobian-free exponential Euler method [Carr et al. (2011)]

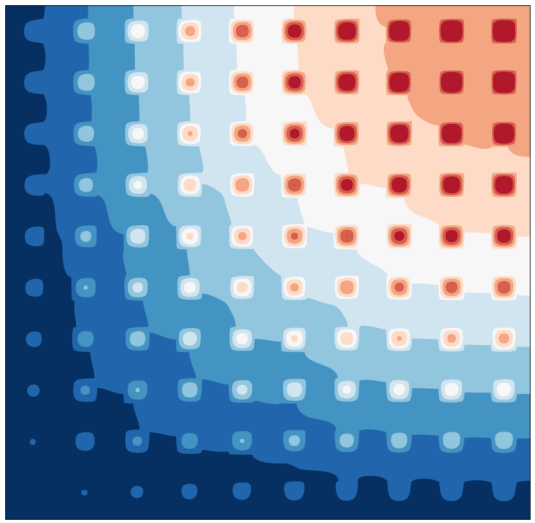
Code implementation details

- Code developed in C++
- Linear algebra operations performed using BLAS and LAPACK libraries:
  (a) Intel MKL (HPC Platform)
  (b) Accelerate framework (Macbook Pro)
- ODE right-hand side function $\boldsymbol{g}(\boldsymbol{u})$ implemented in parallel using OpenMP

```
#include <omp.h>
omp_set_num_threads(NUM_THREADS);
#pragma omp parallel for
for (k=0; k<num_macro_elements; k++)
{
  ...
}
```

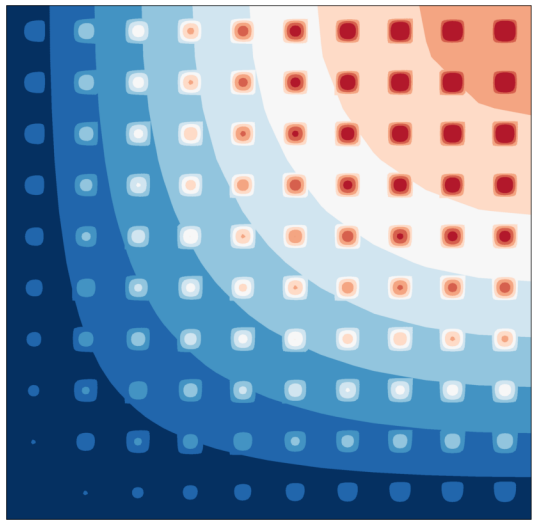
- Code accommodates both triangular and quad elements at both scales

Test Case: Diffusion equation



Full Model
(15 mins)

Macroscopic Model
(1 sec)

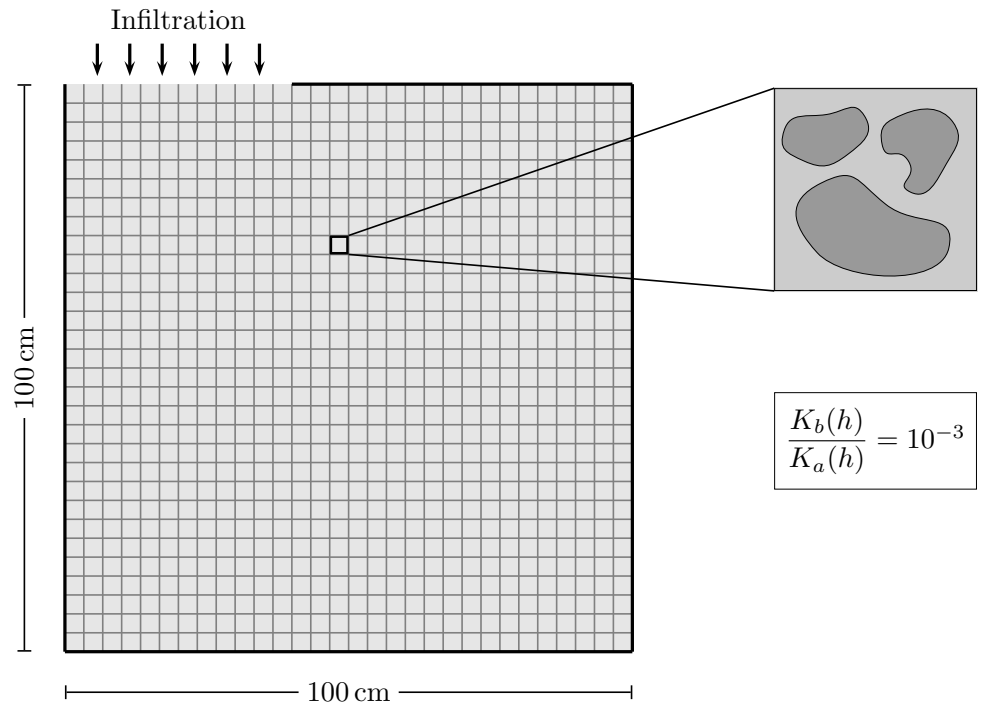


Two-scale Model
(20 secs)

Mesh details:

- Full Model: 400×400 grid, $N = 160,801$
- Macroscopic Model: 20×20 grid, $N = 441$
- Two-scale Model: Macro (20×20 grid), Micro (10×10 grid), $N = 48,841$

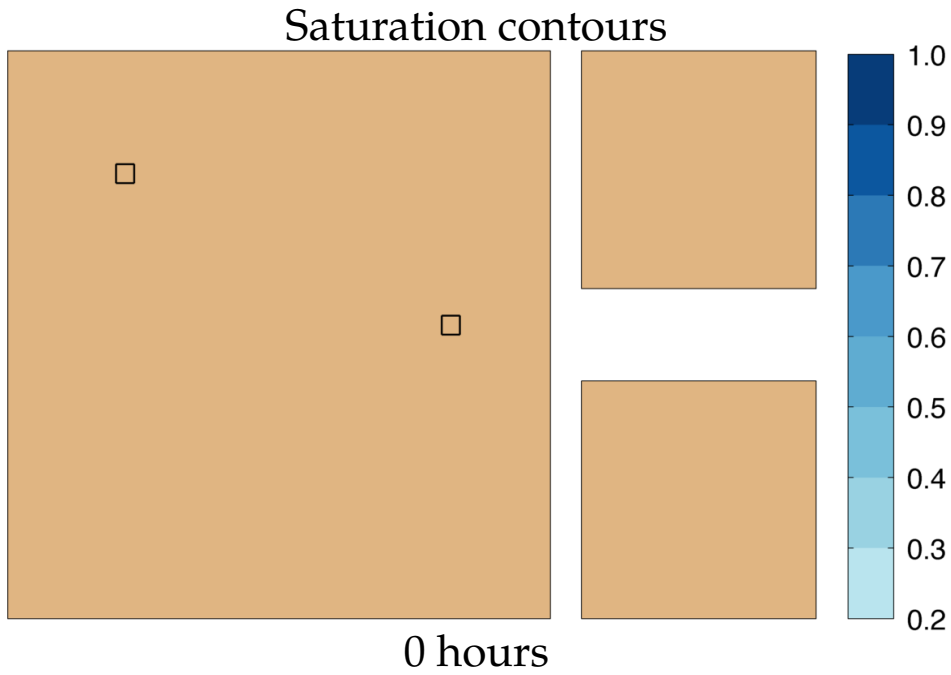
Simulation: Unsaturated water flow



Mesh details:

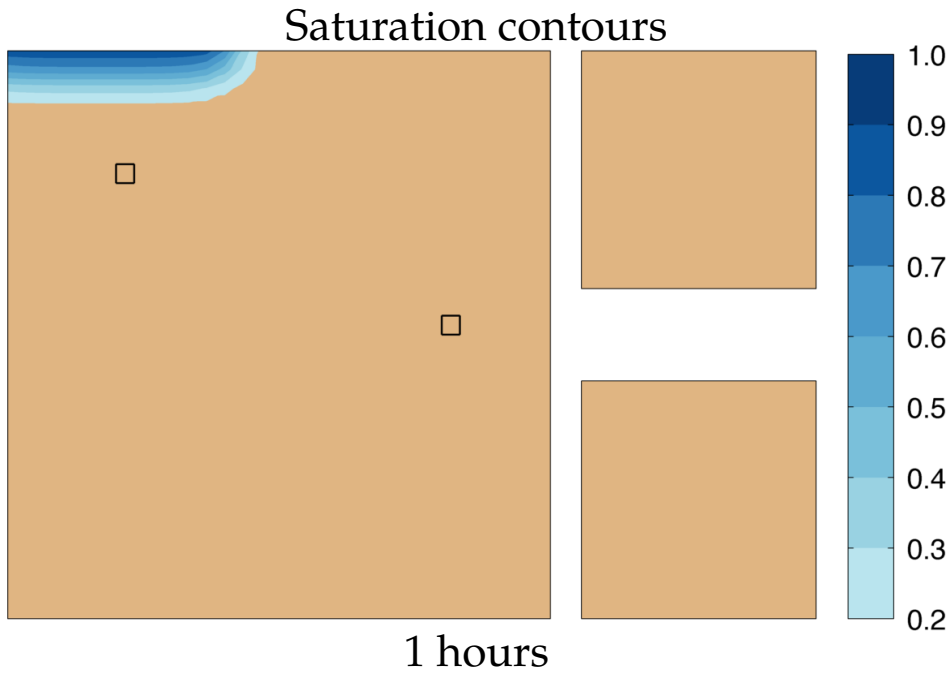
- Macroscopic mesh (structured): 30×30 grid, 961 nodes
- Microscopic mesh (unstructured): 532 elements, 316 nodes
- Total number of unknowns: $N = 961 + 900 \times 316 = 285,361$

Simulation: Unsaturated water flow



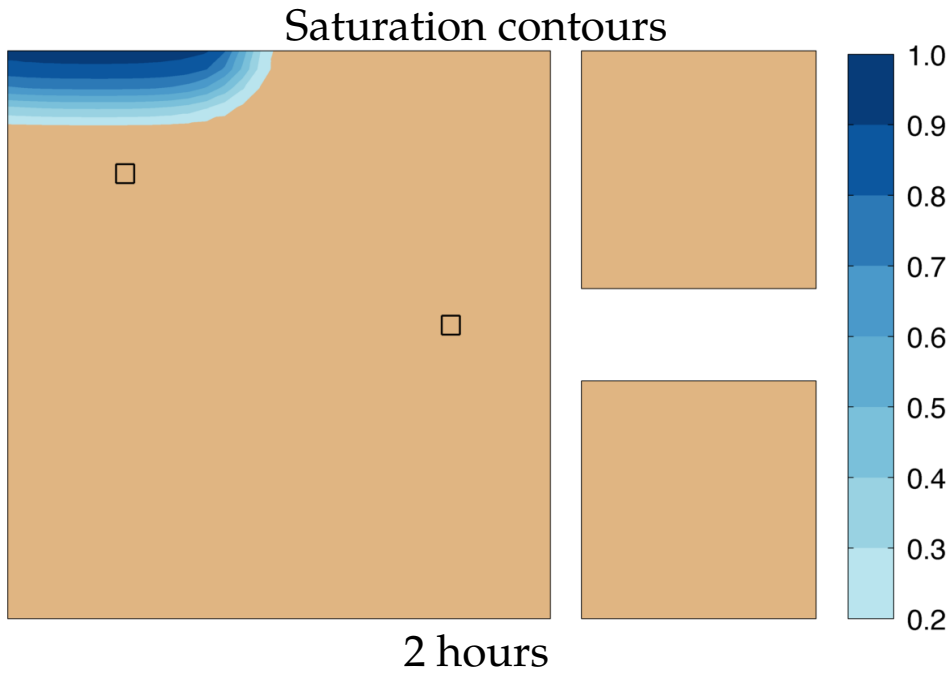
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



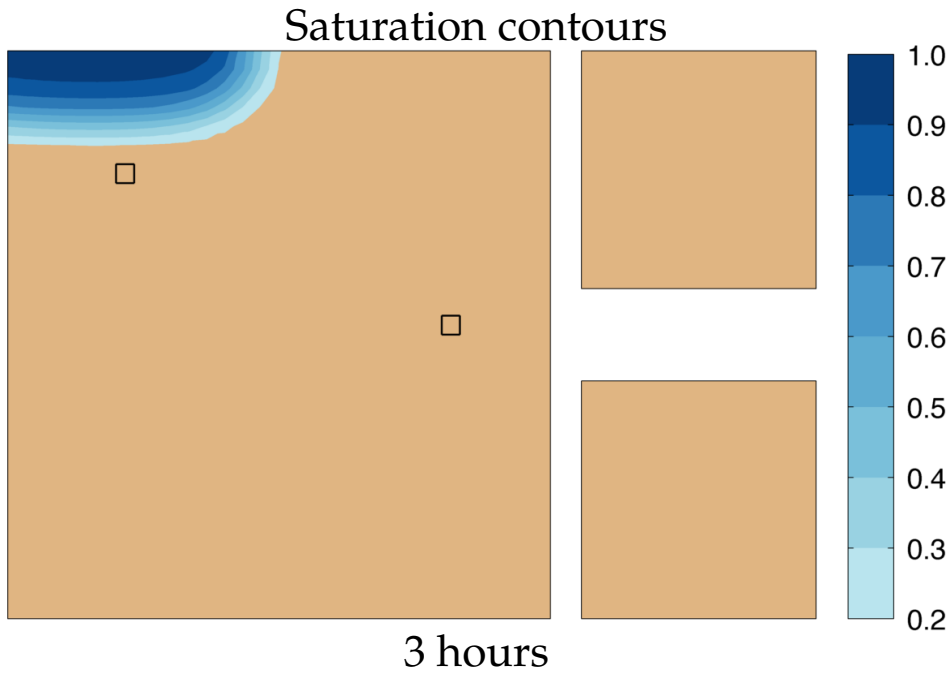
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



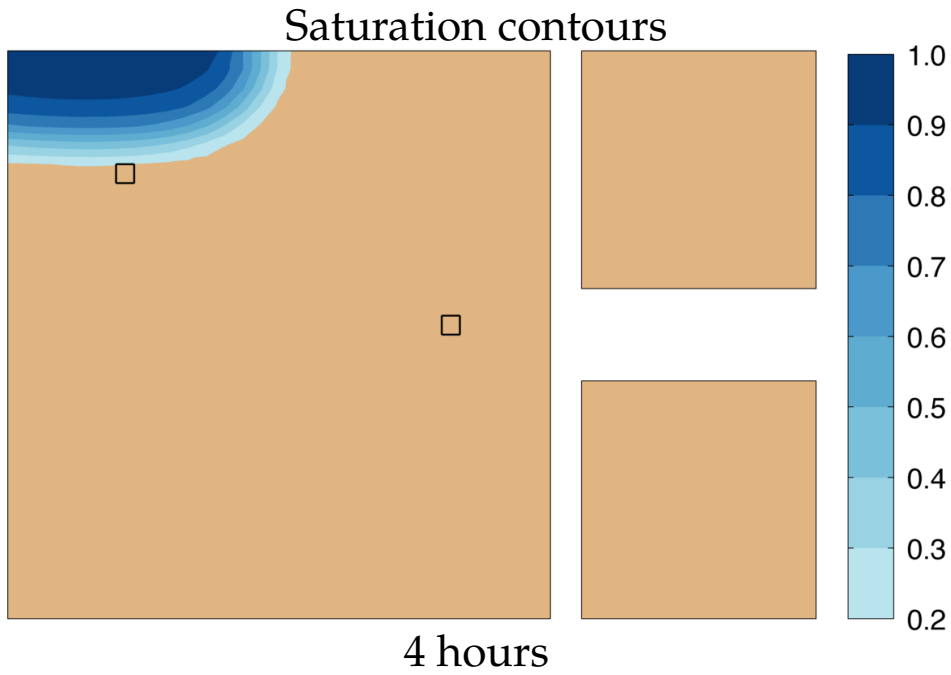
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



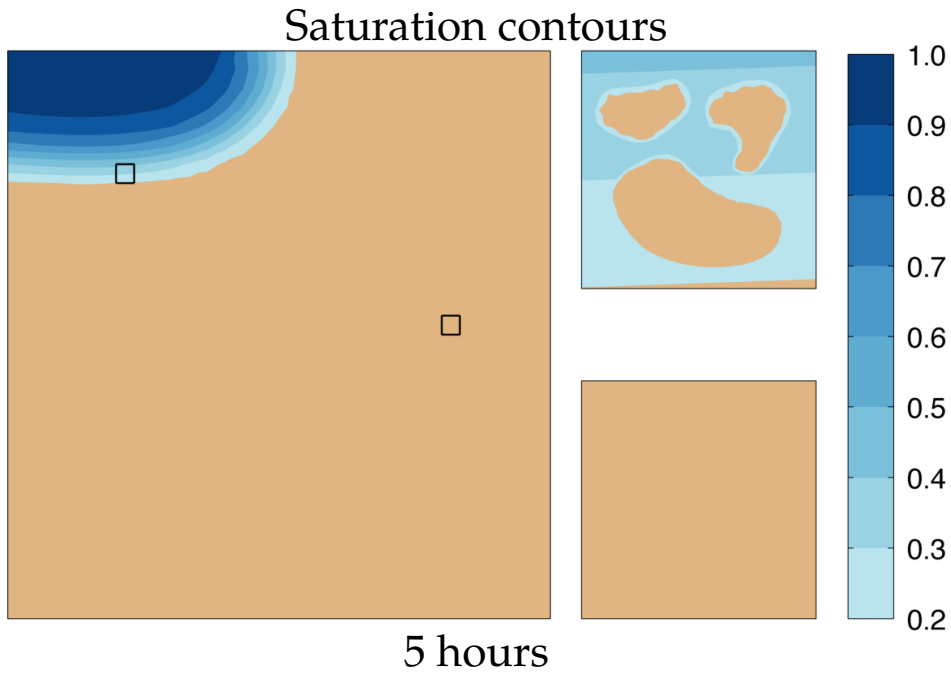
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



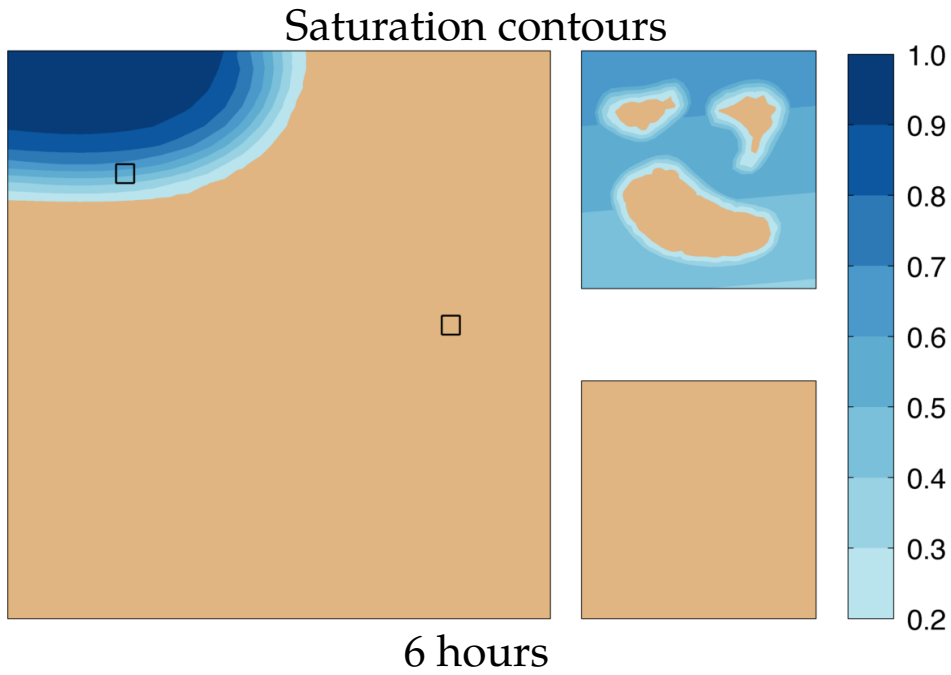
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



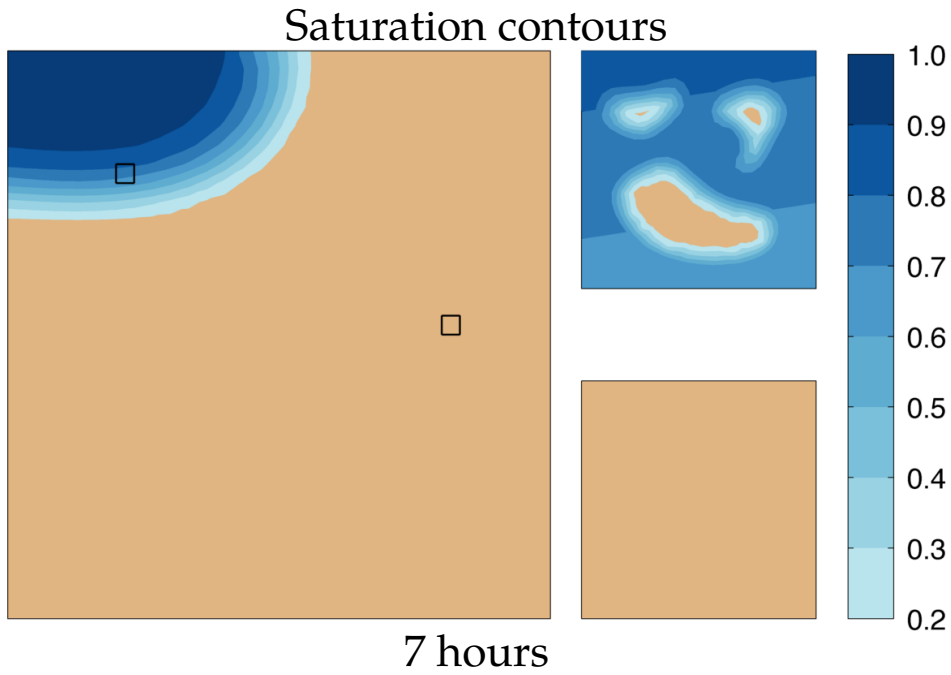
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



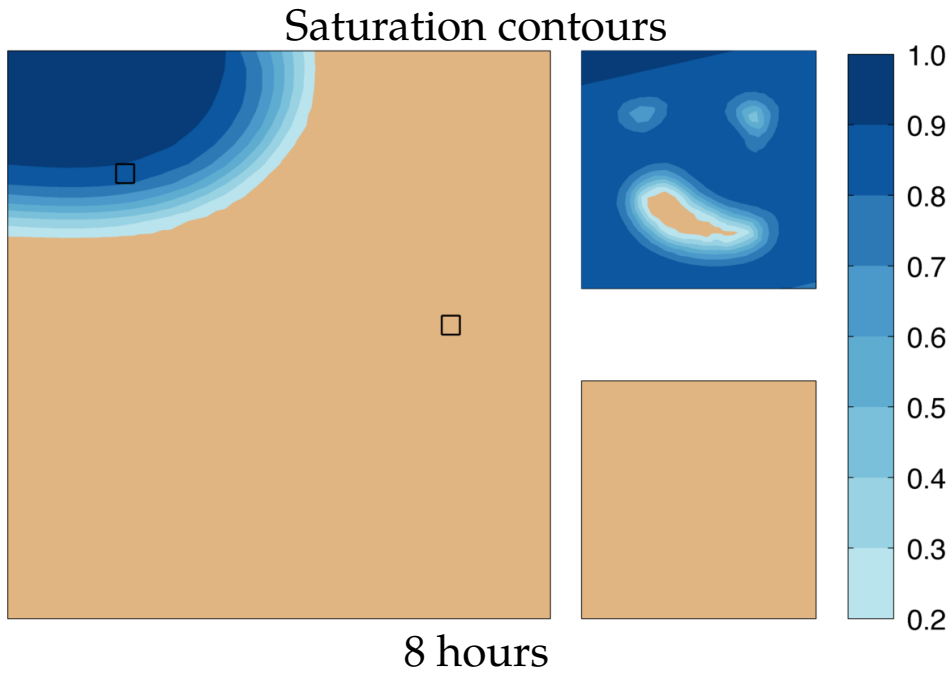
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



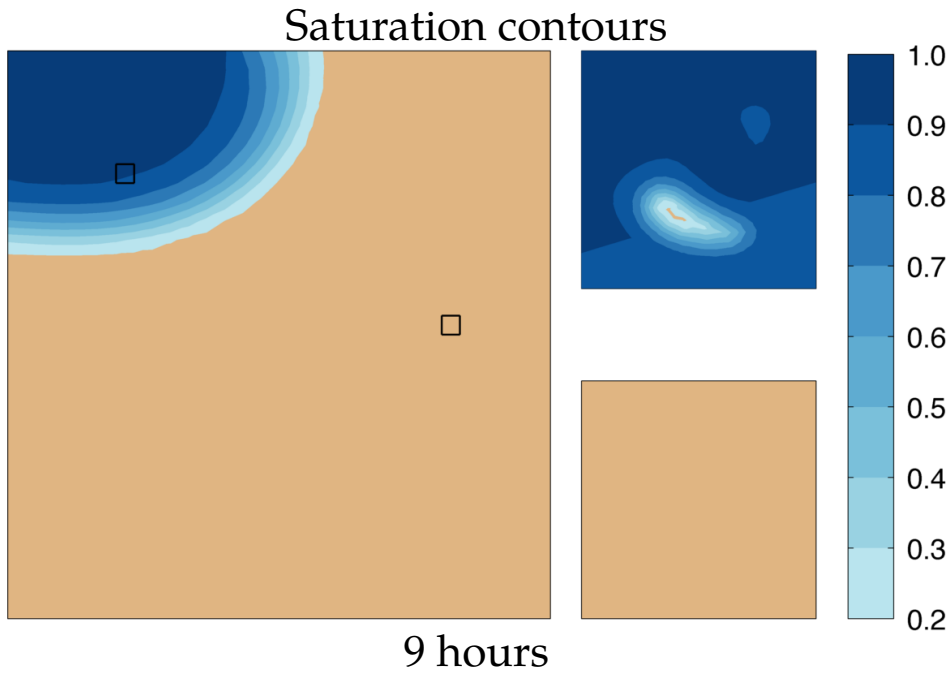
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



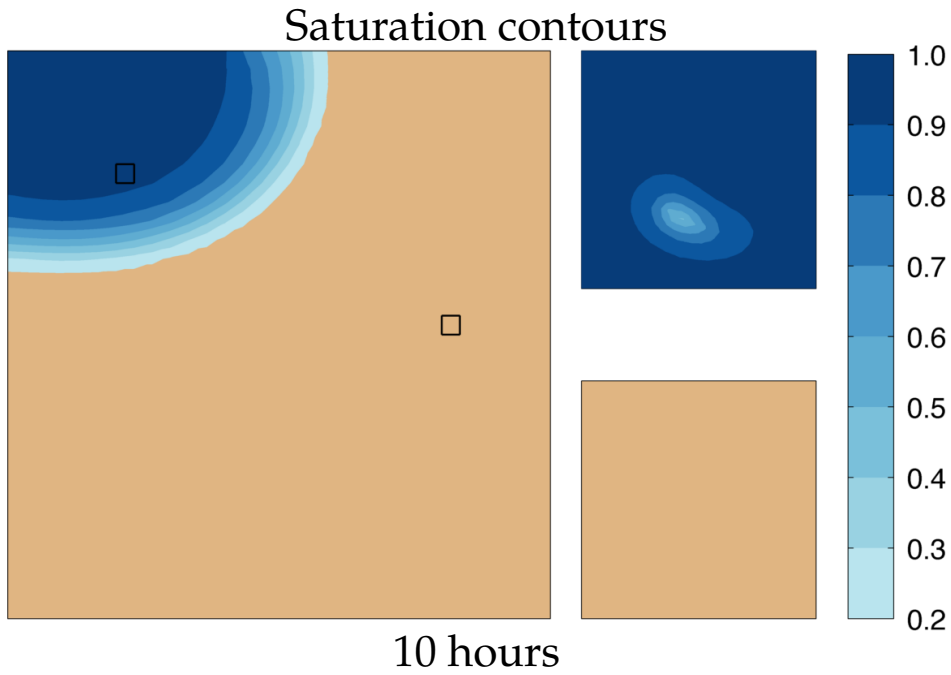
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



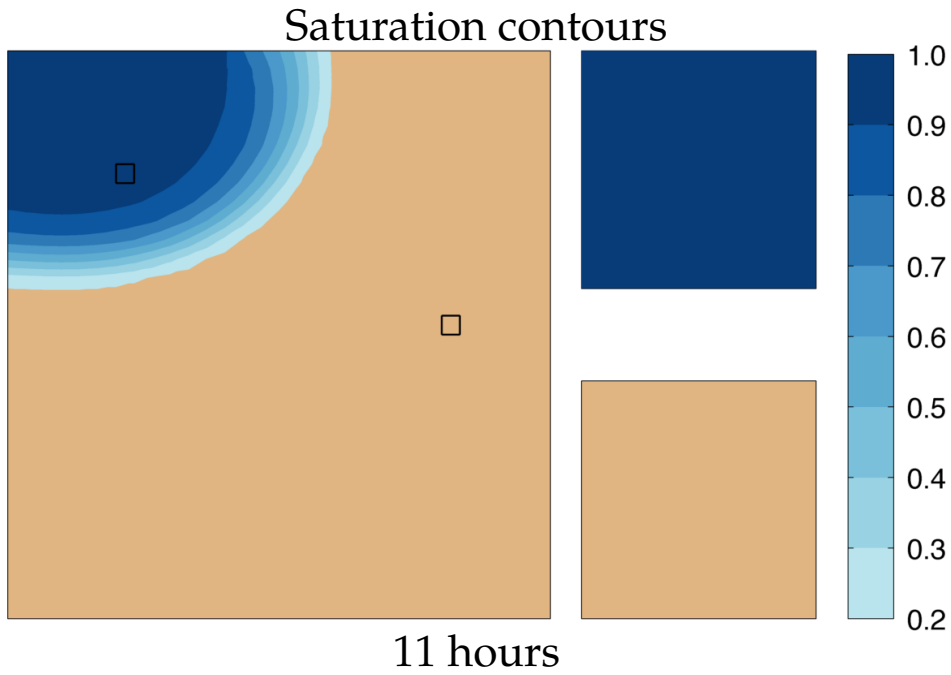
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



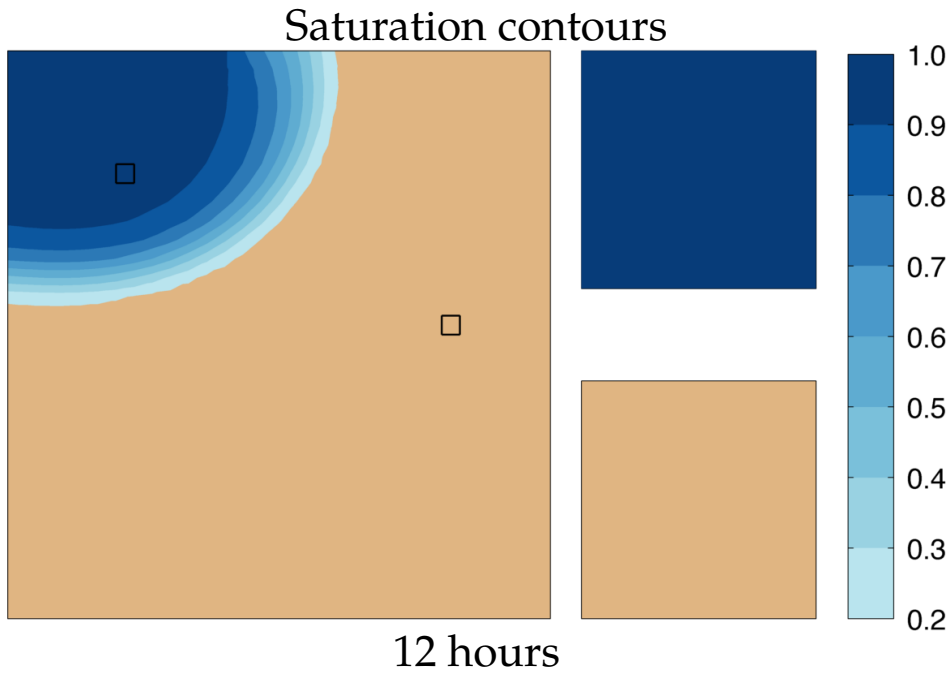
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



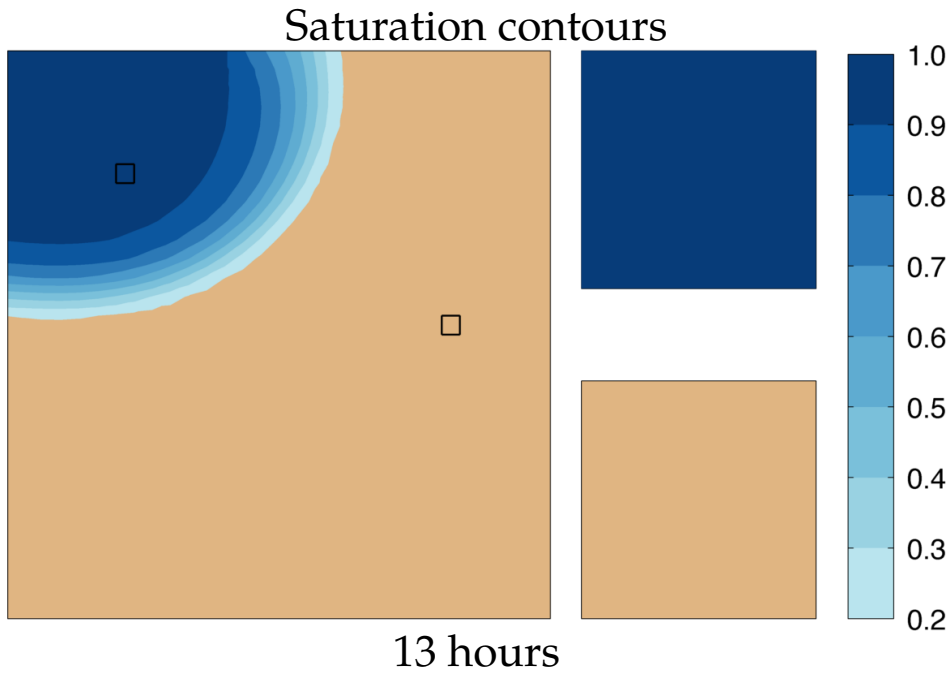
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



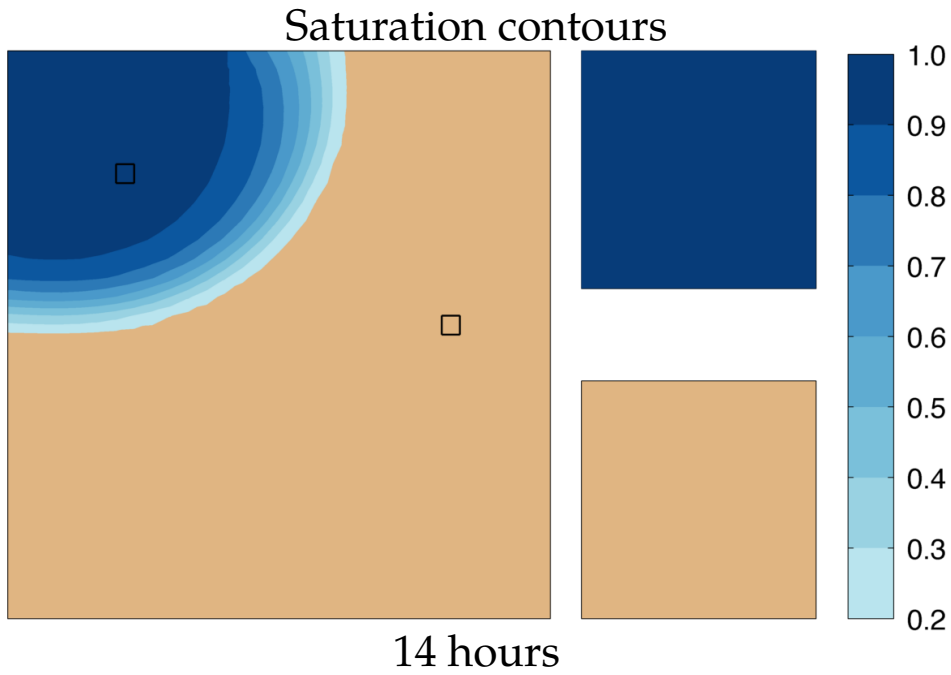
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



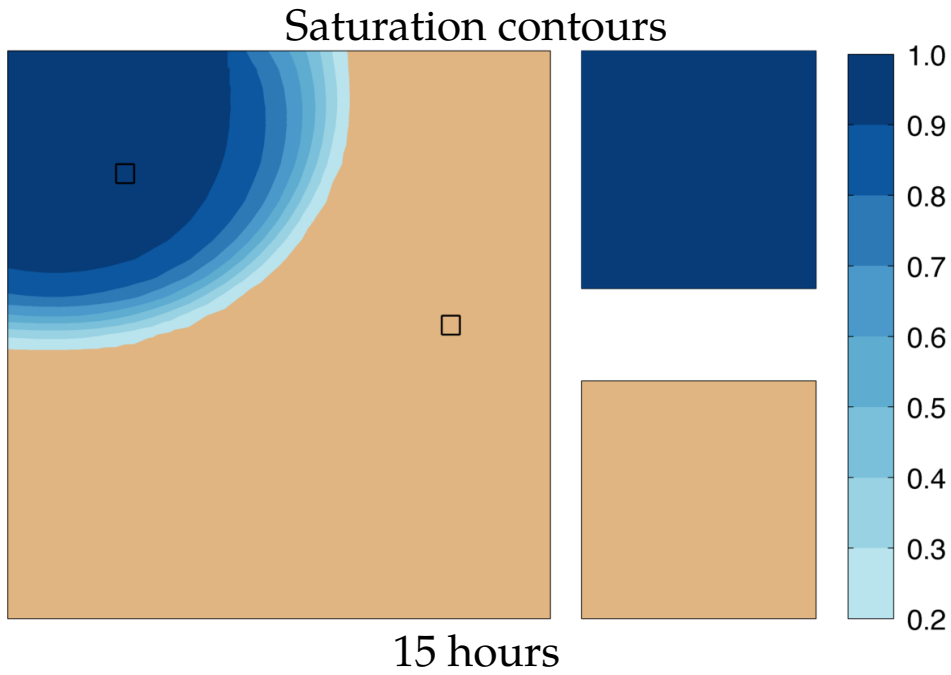
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



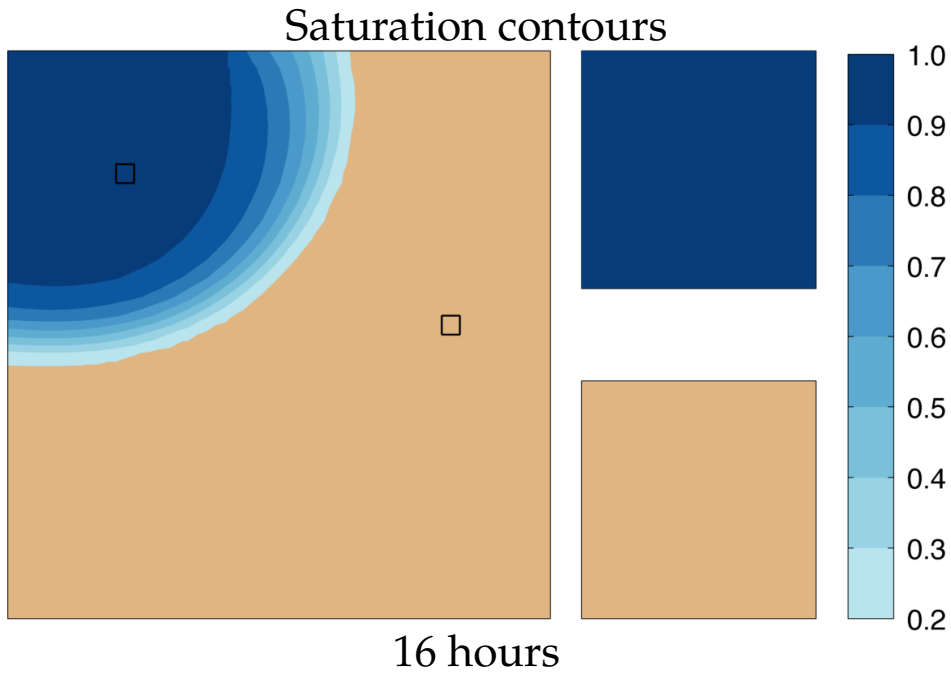
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



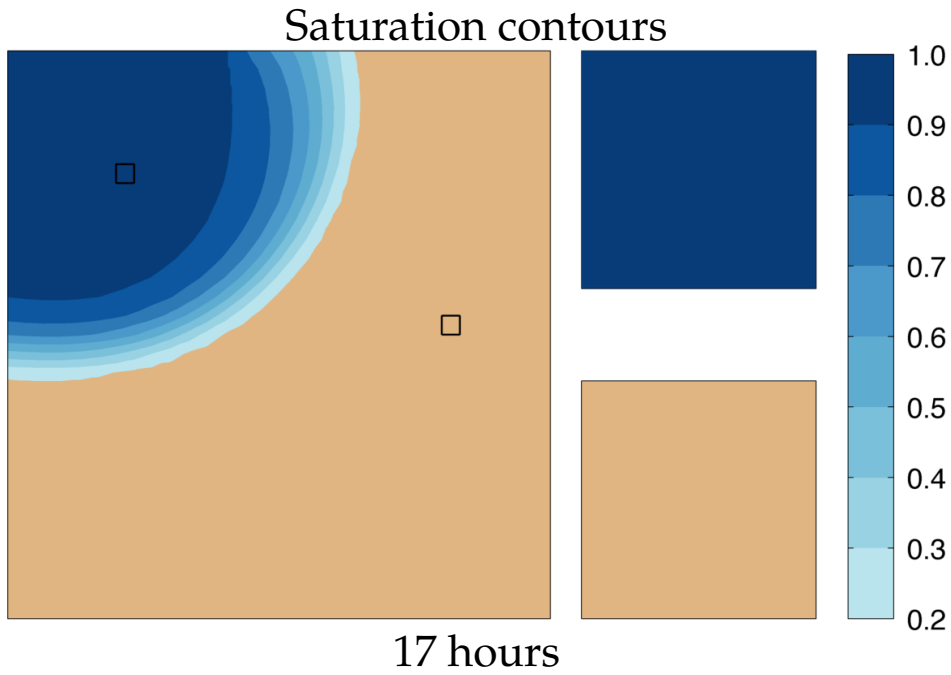
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



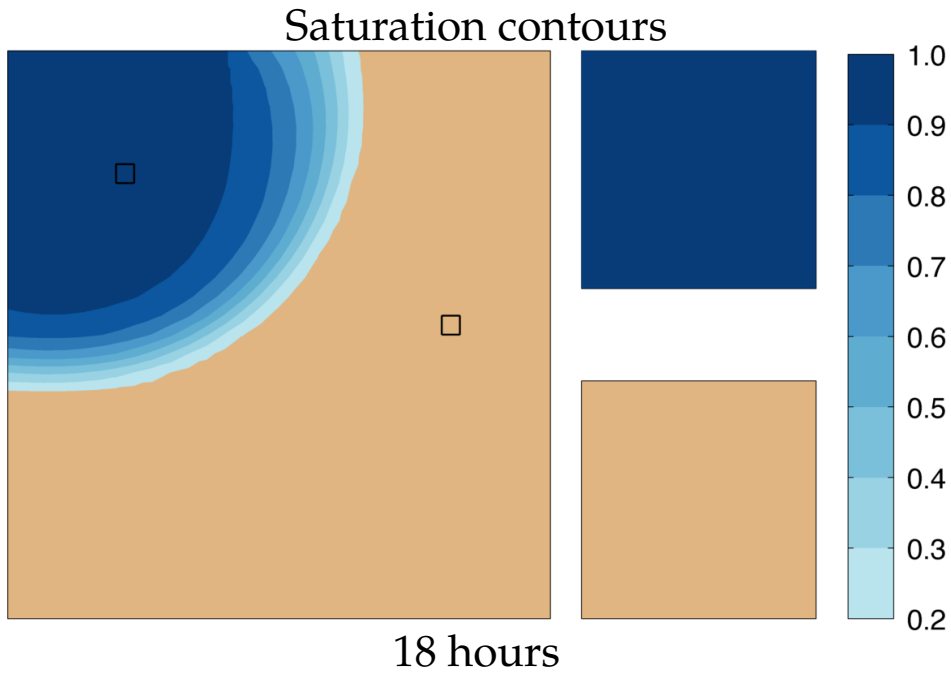
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



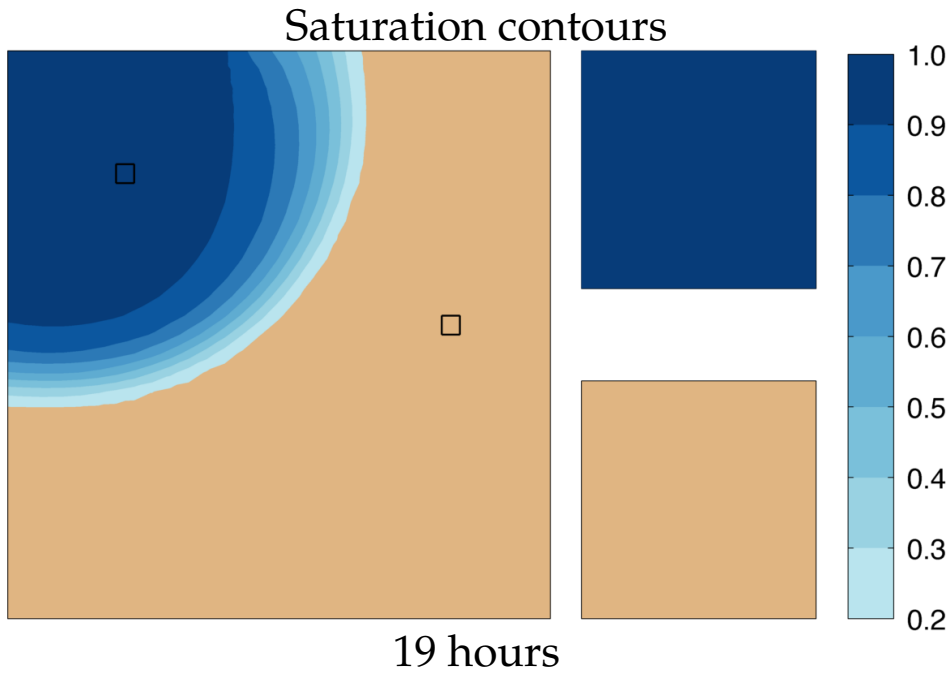
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



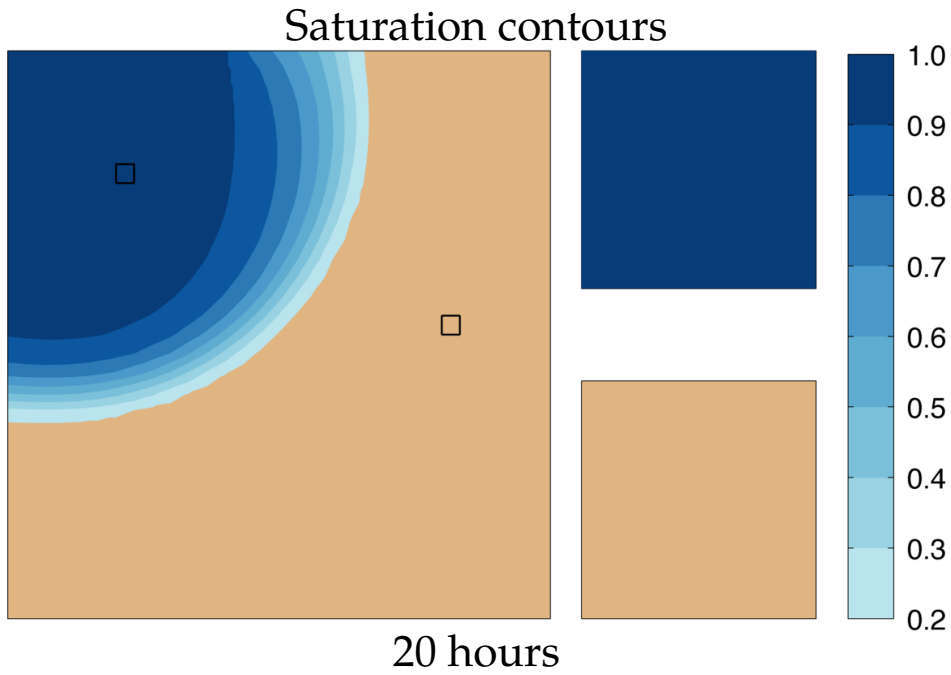
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



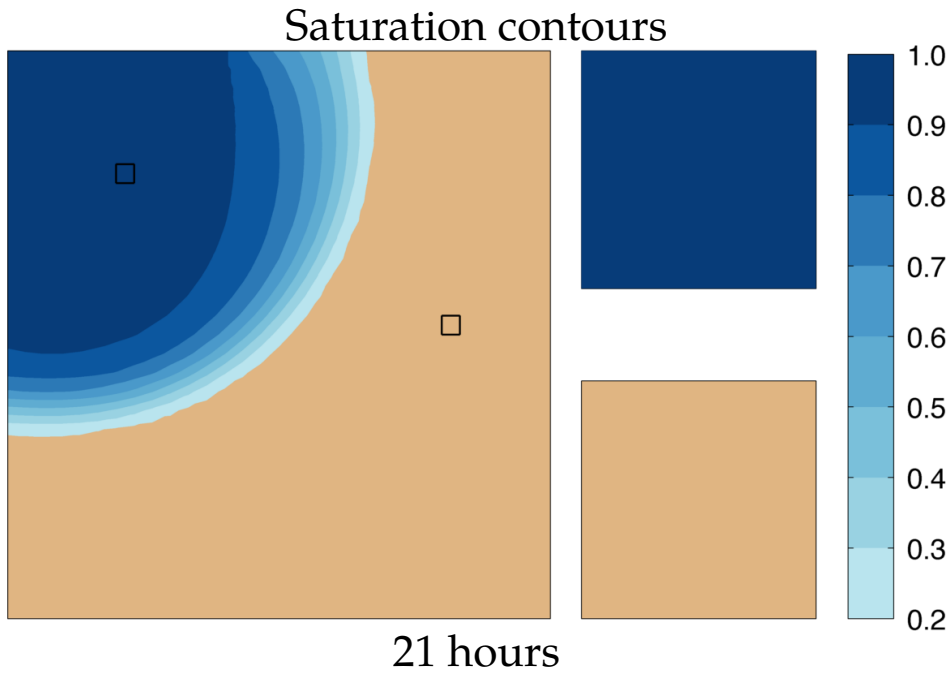
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



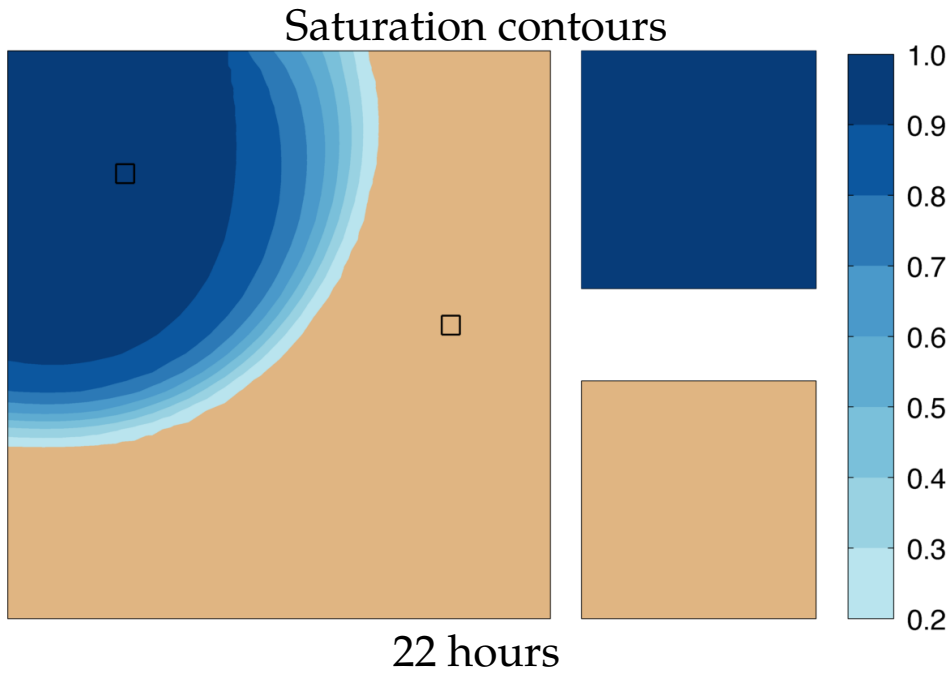
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



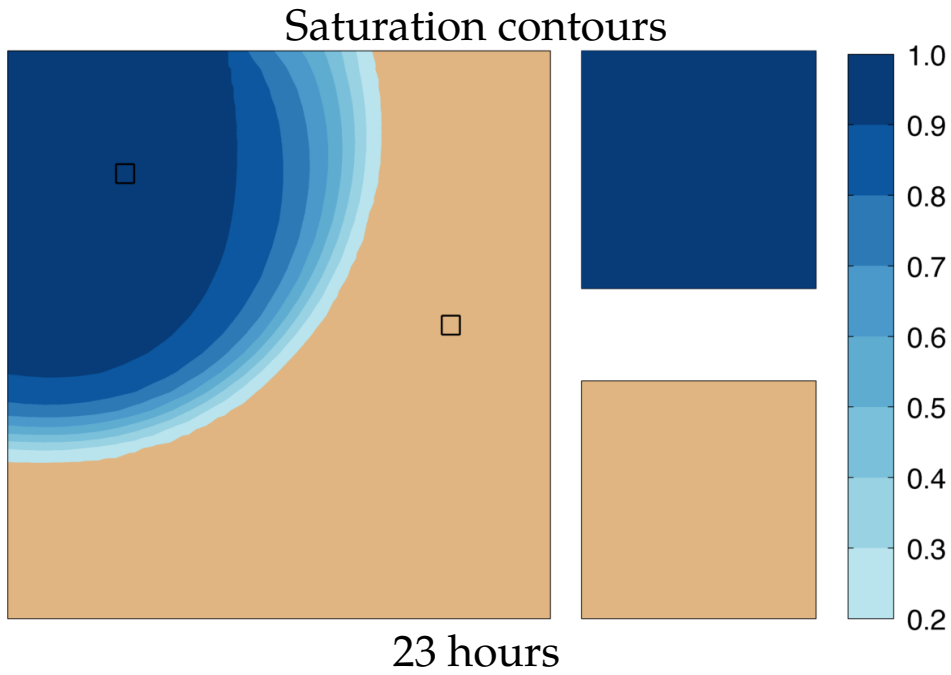
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



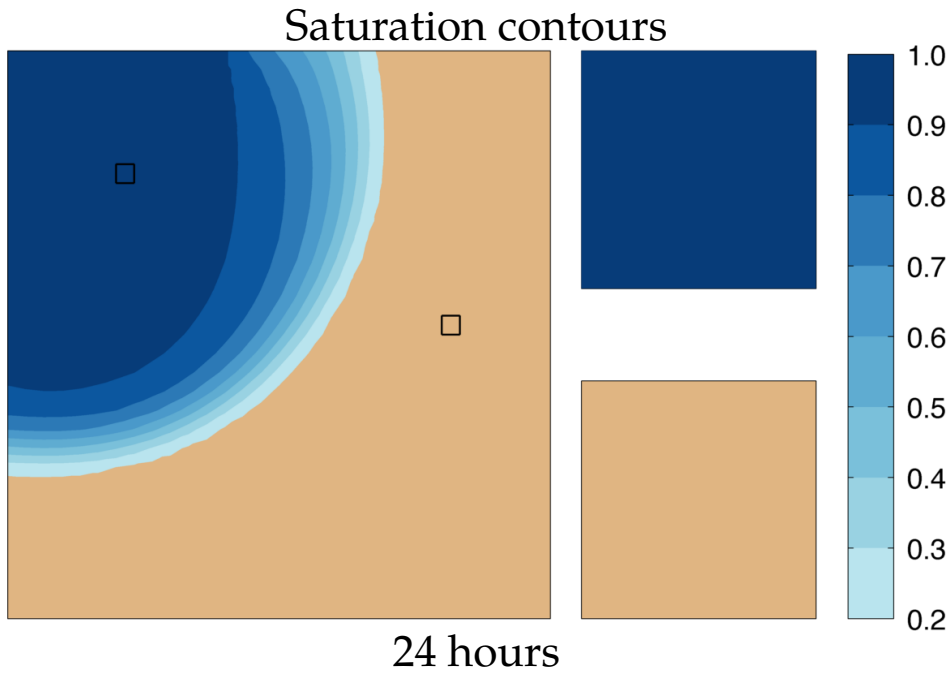
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



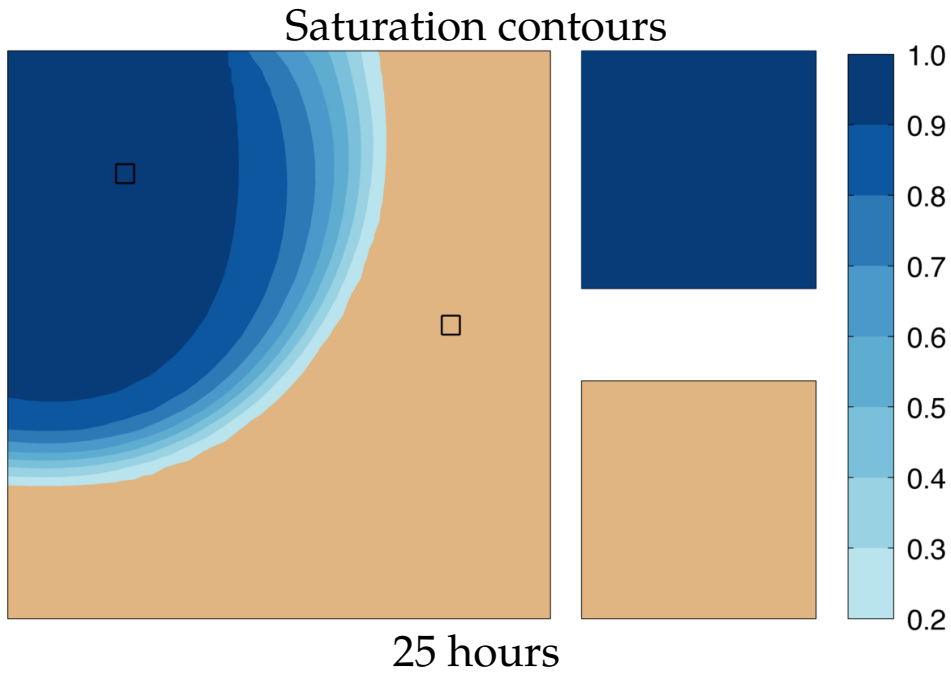
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



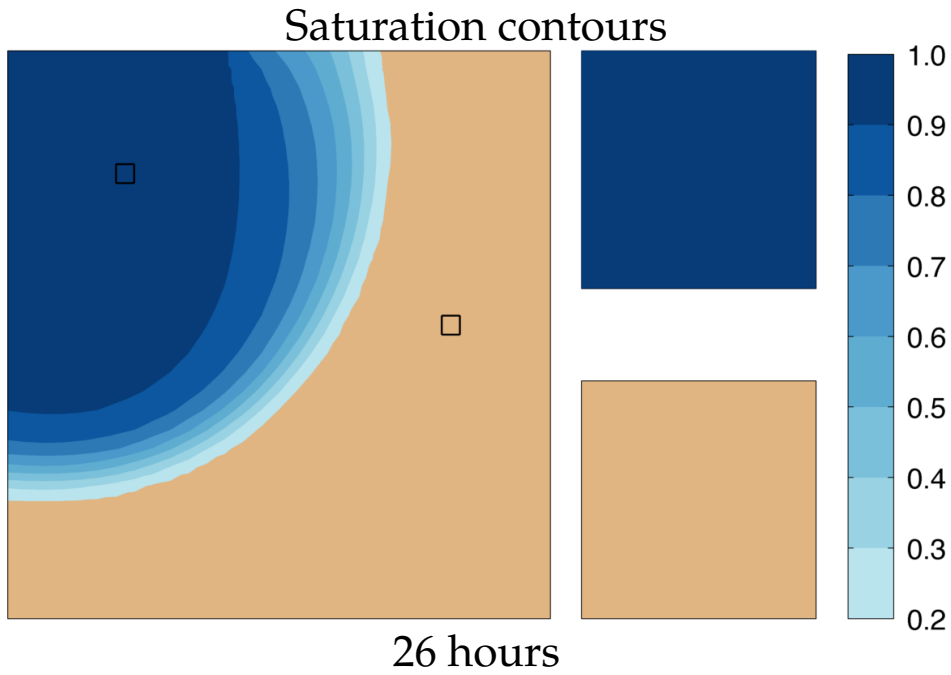
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



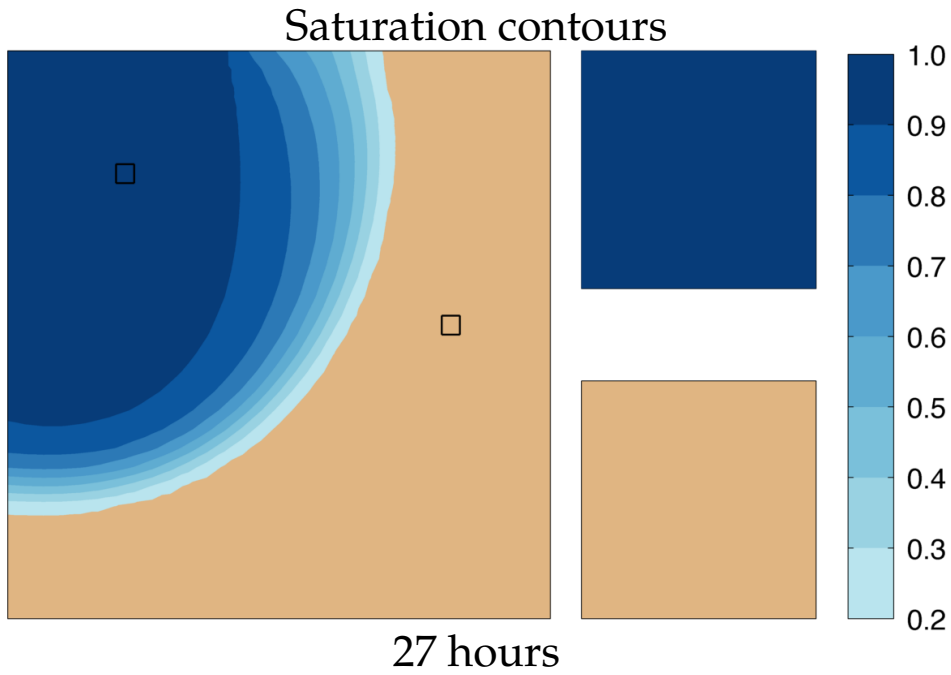
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



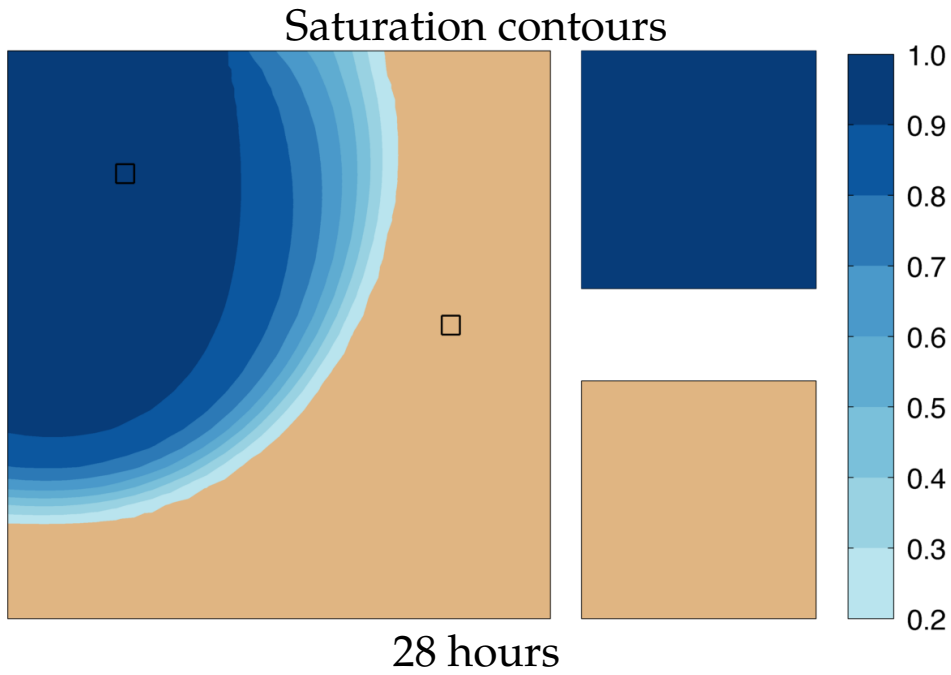
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



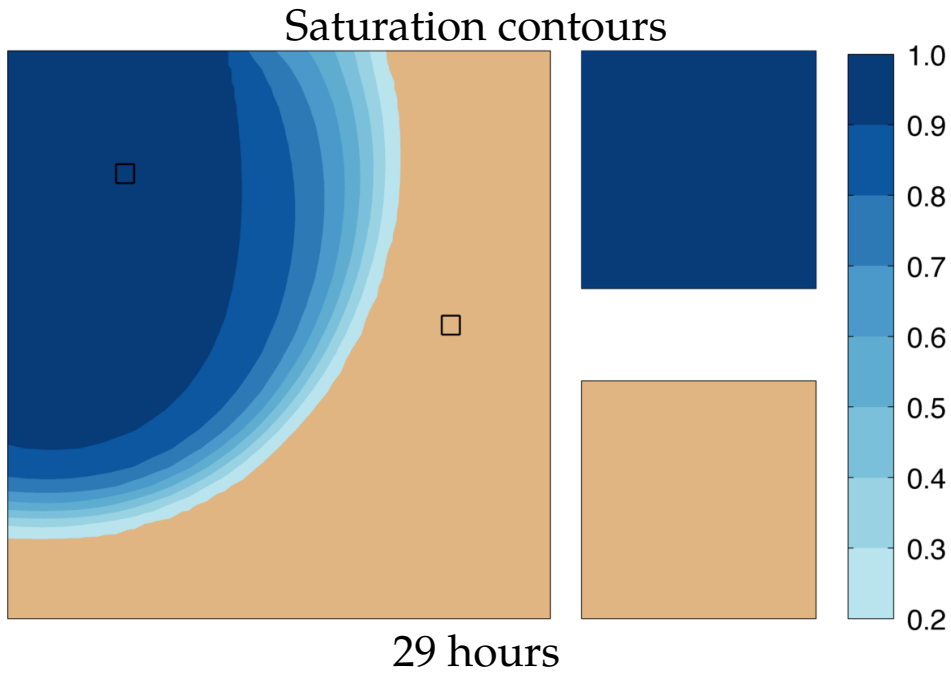
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



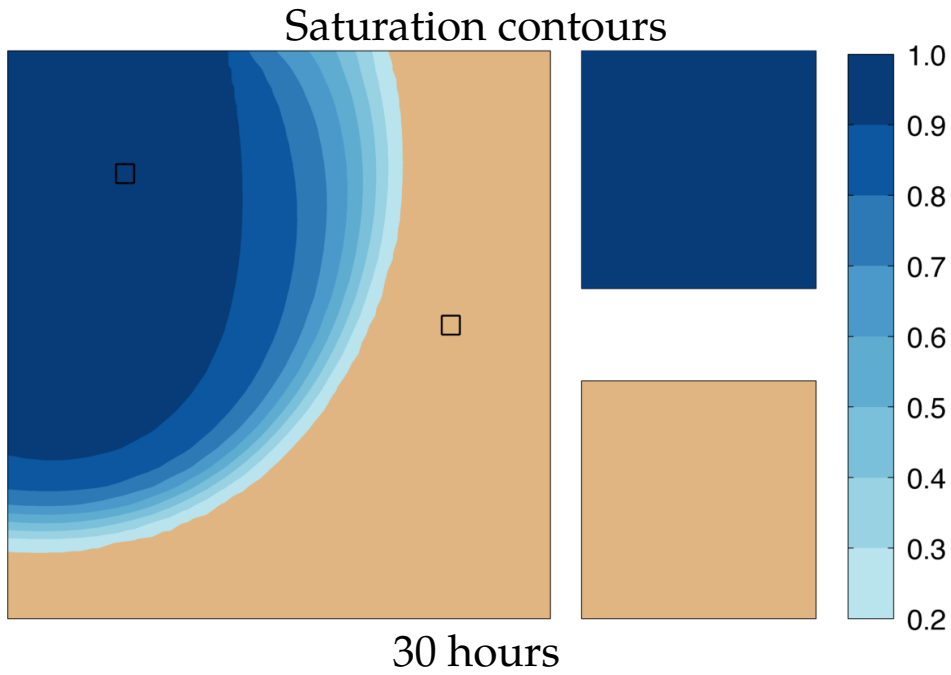
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



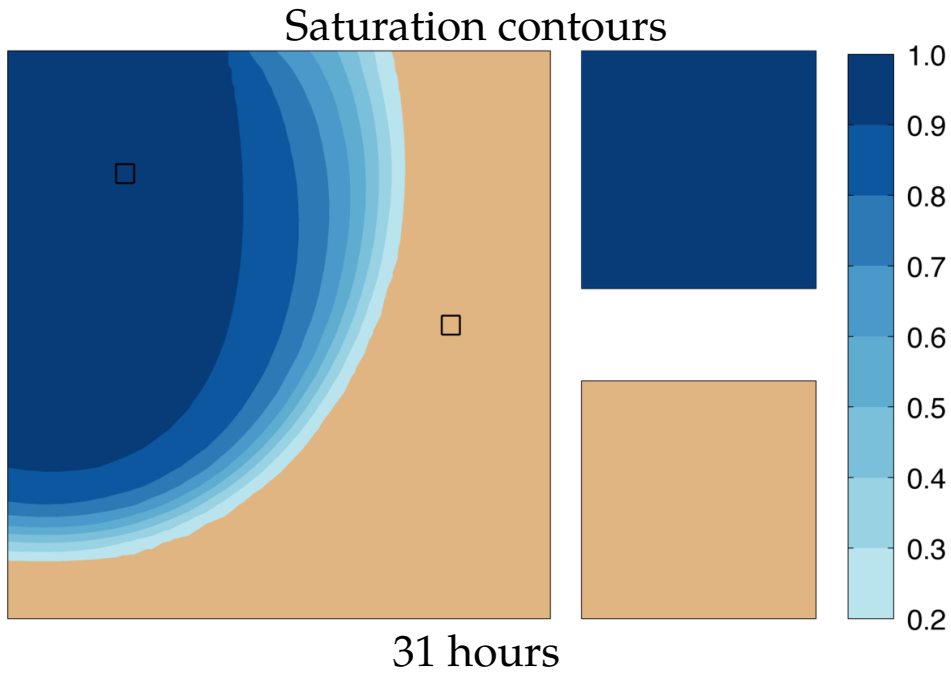
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



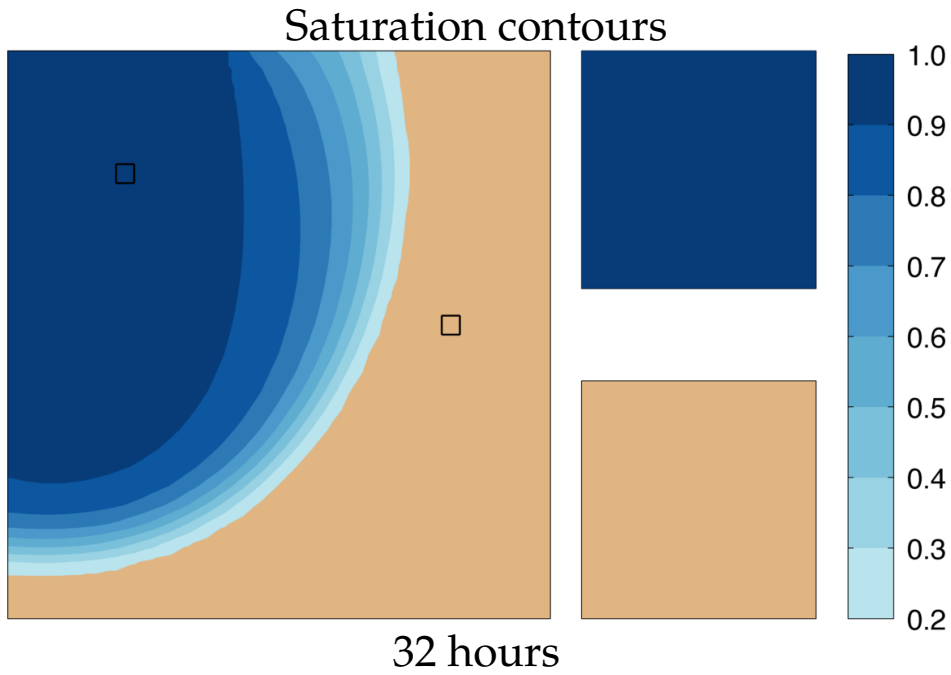
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



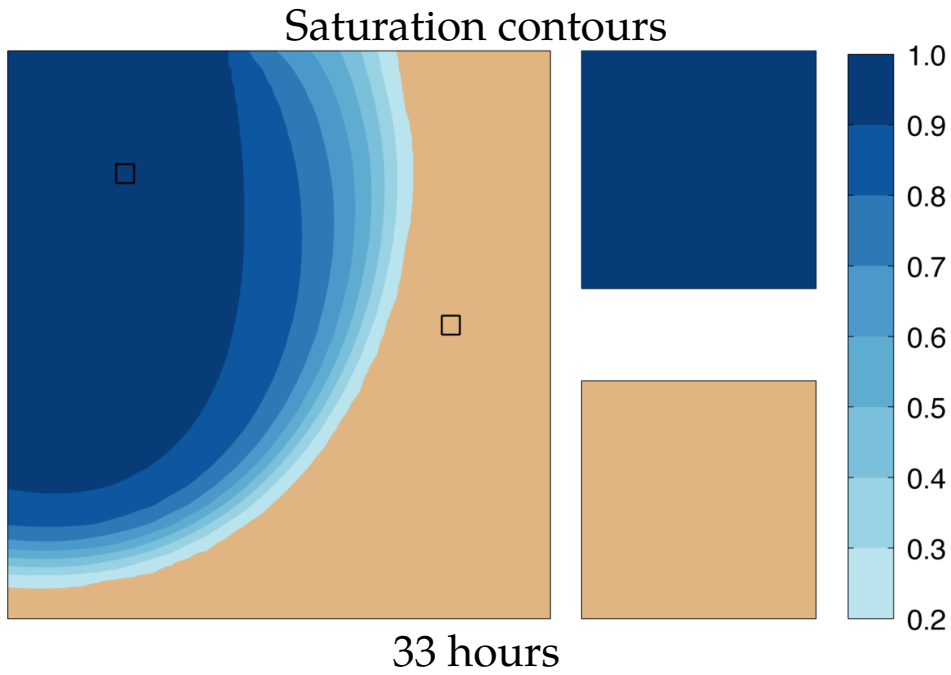
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



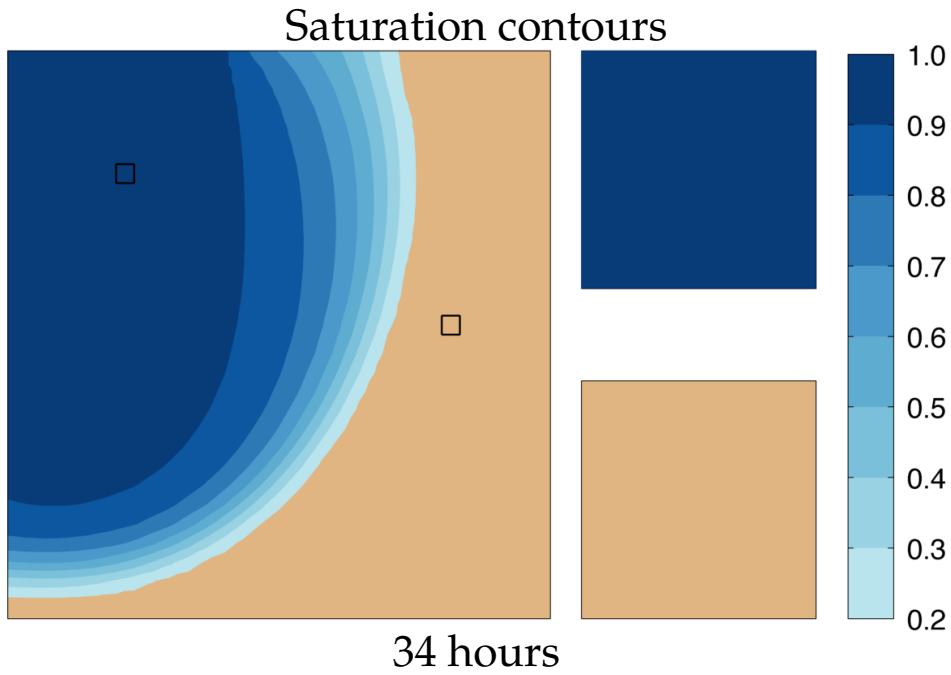
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



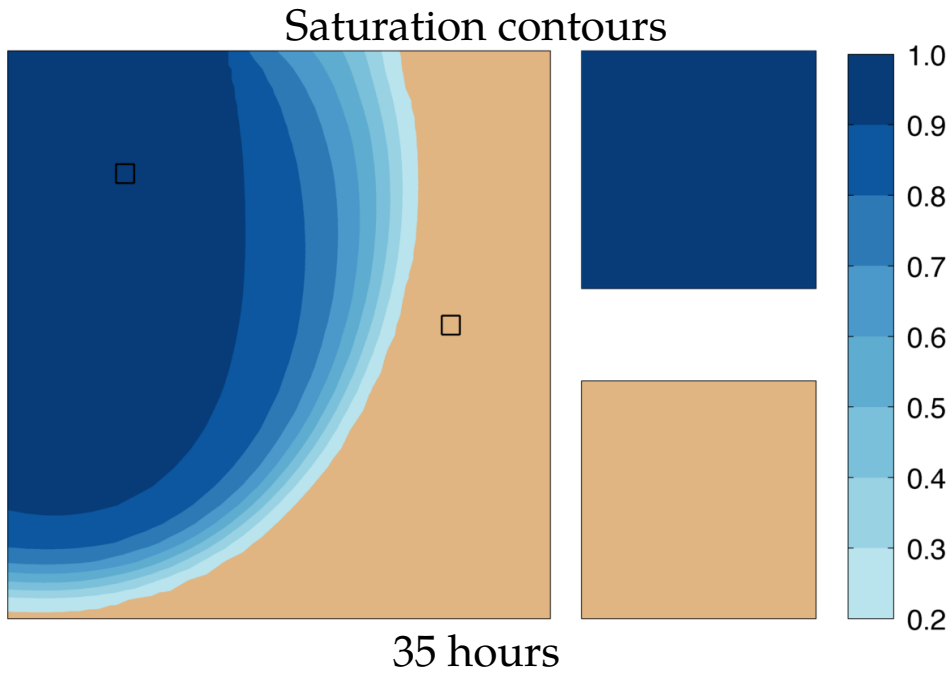
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



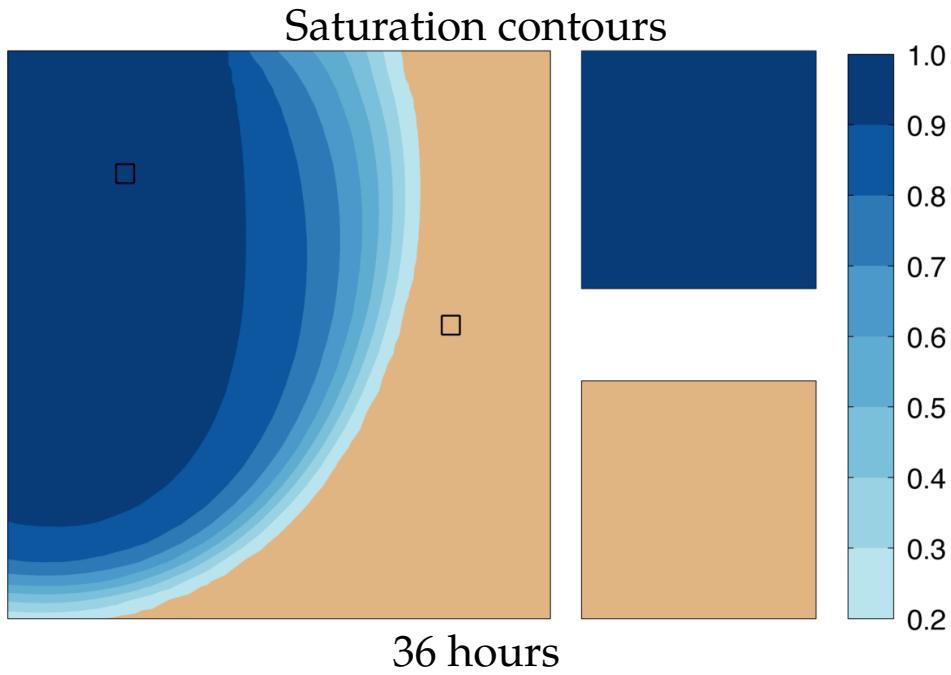
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



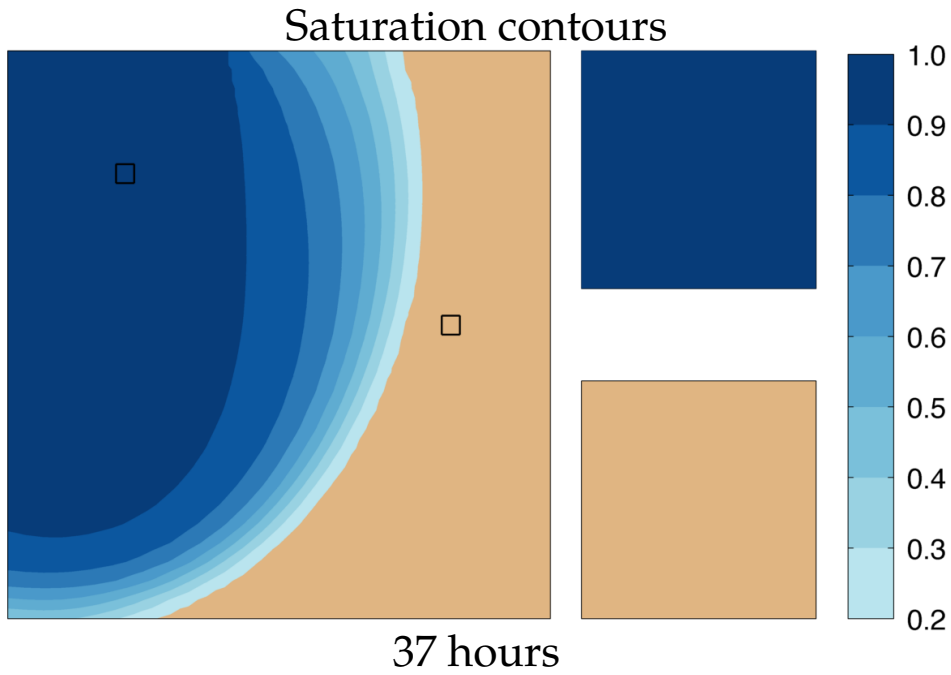
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



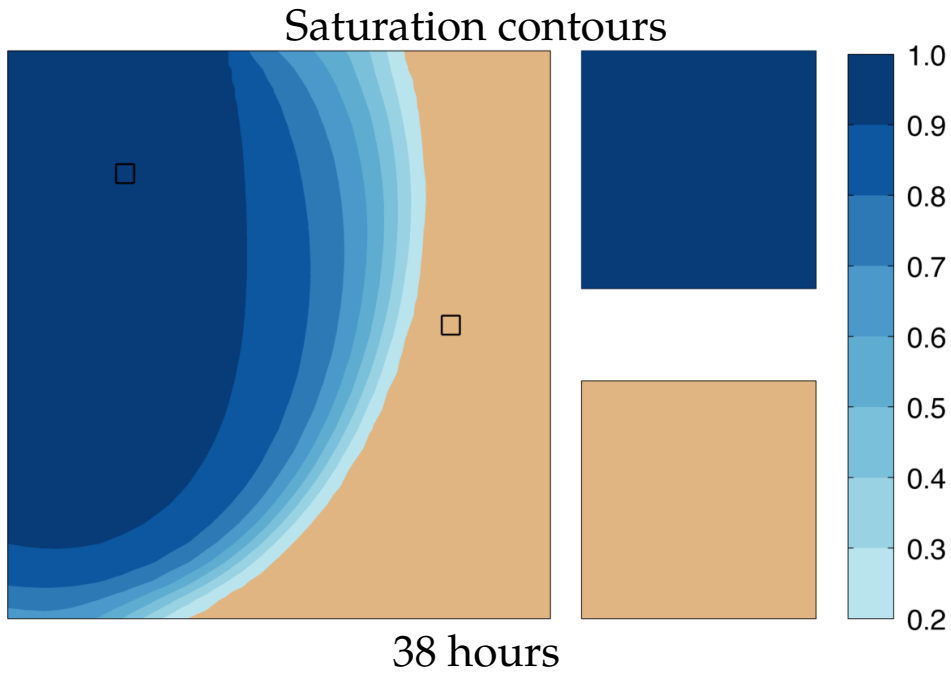
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



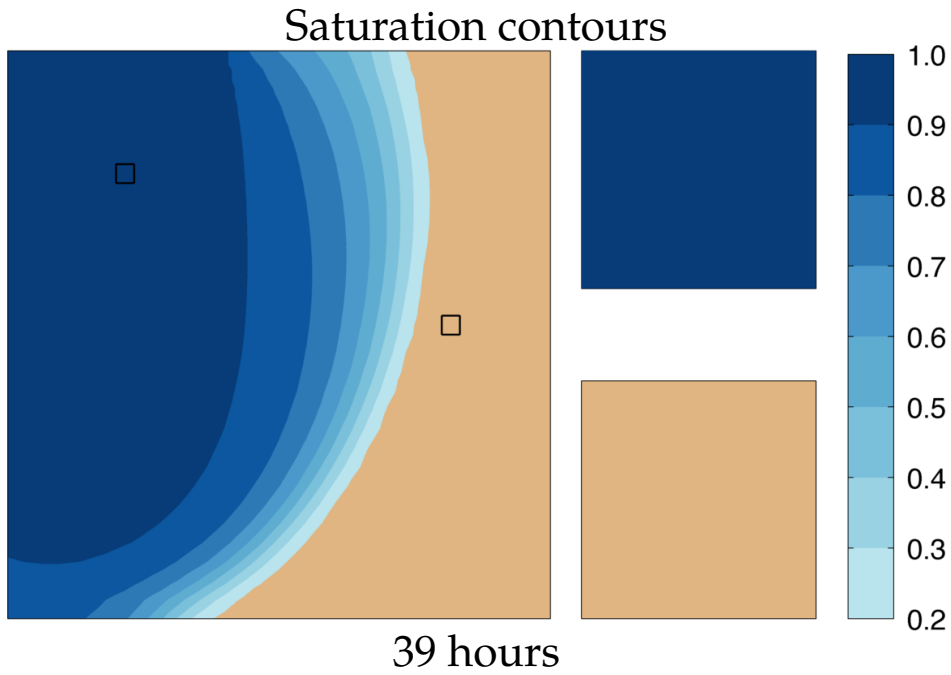
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



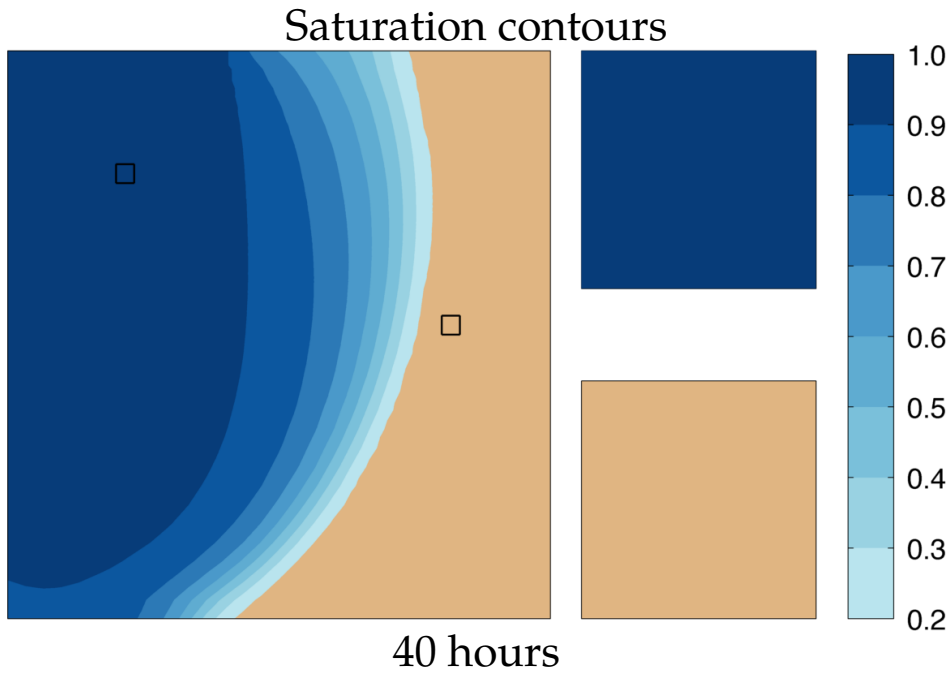
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



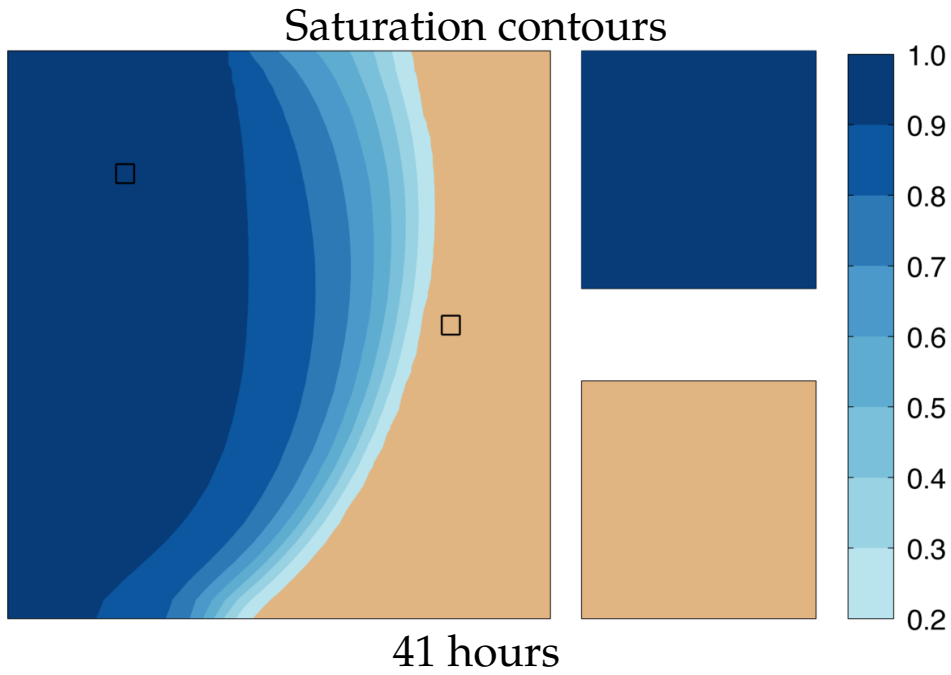
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



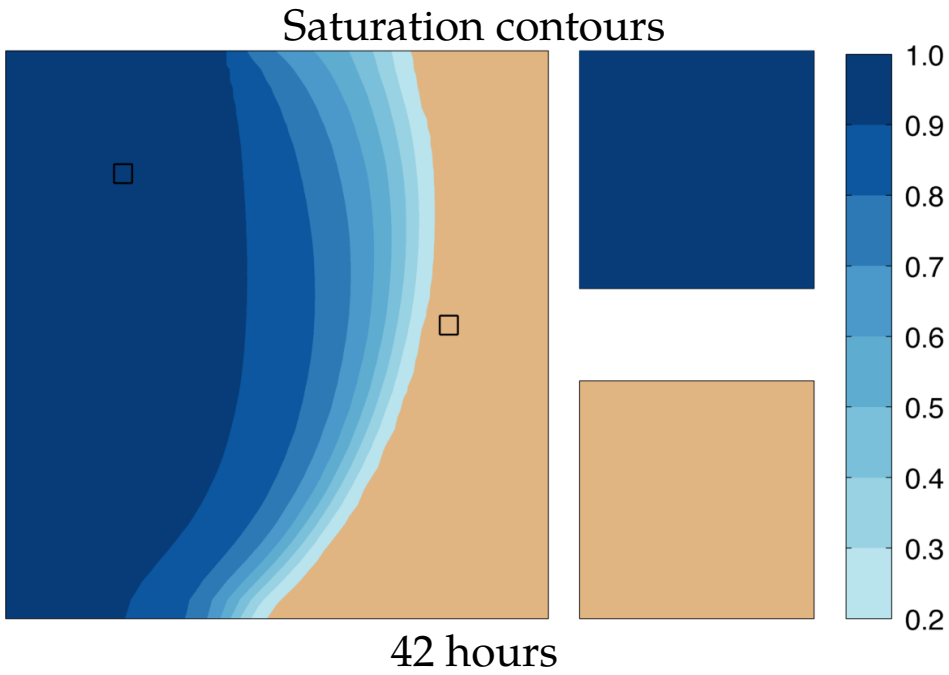
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



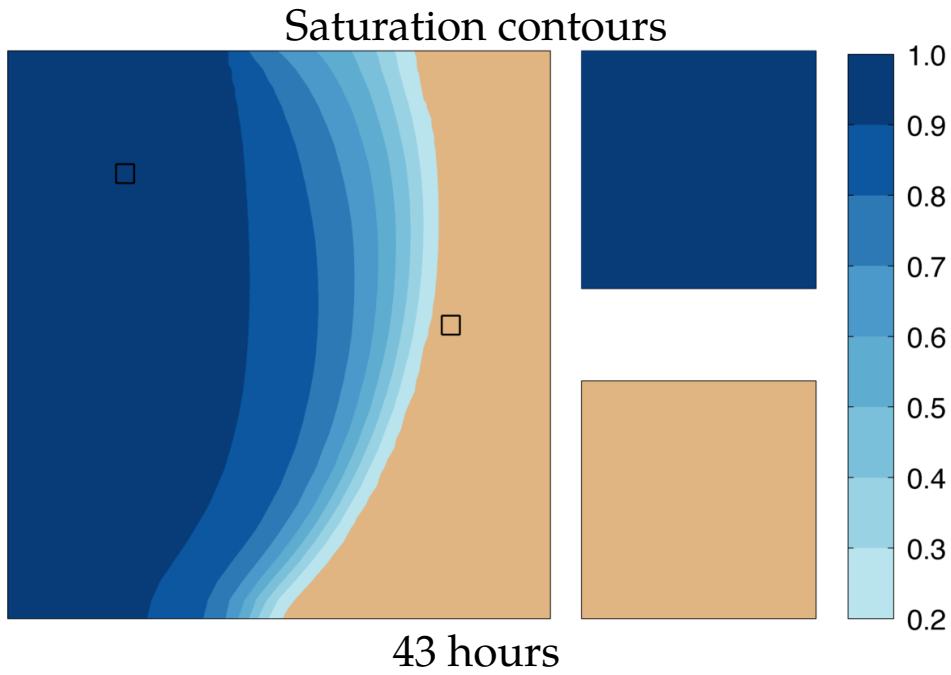
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



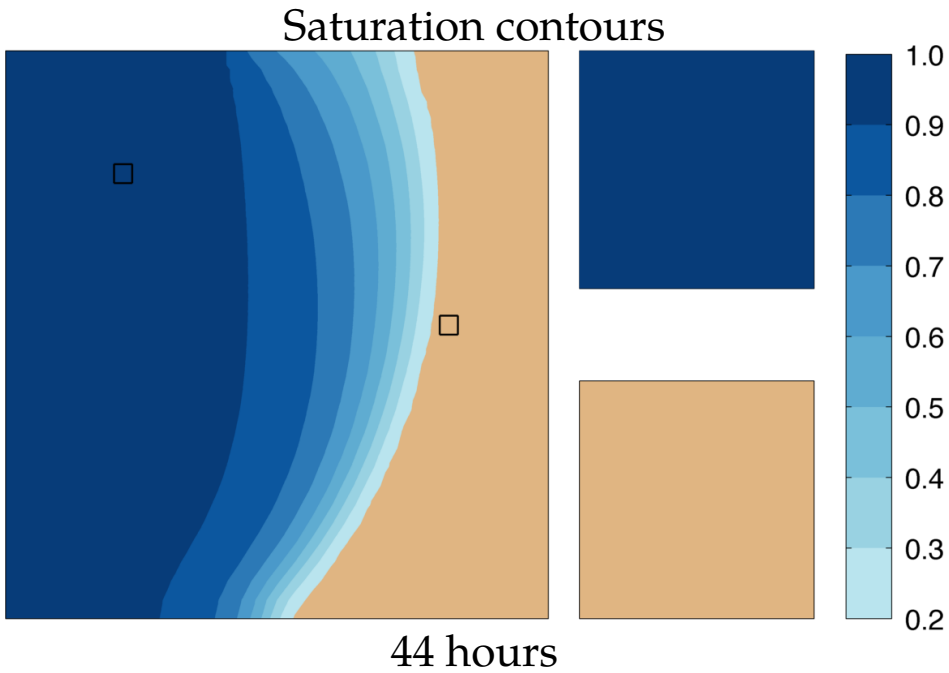
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



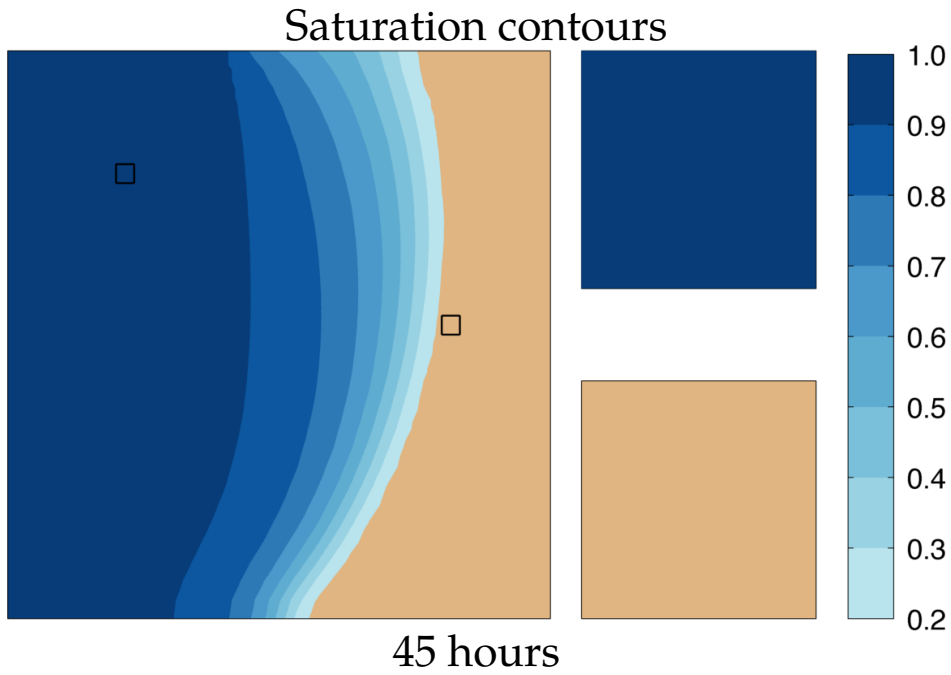
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



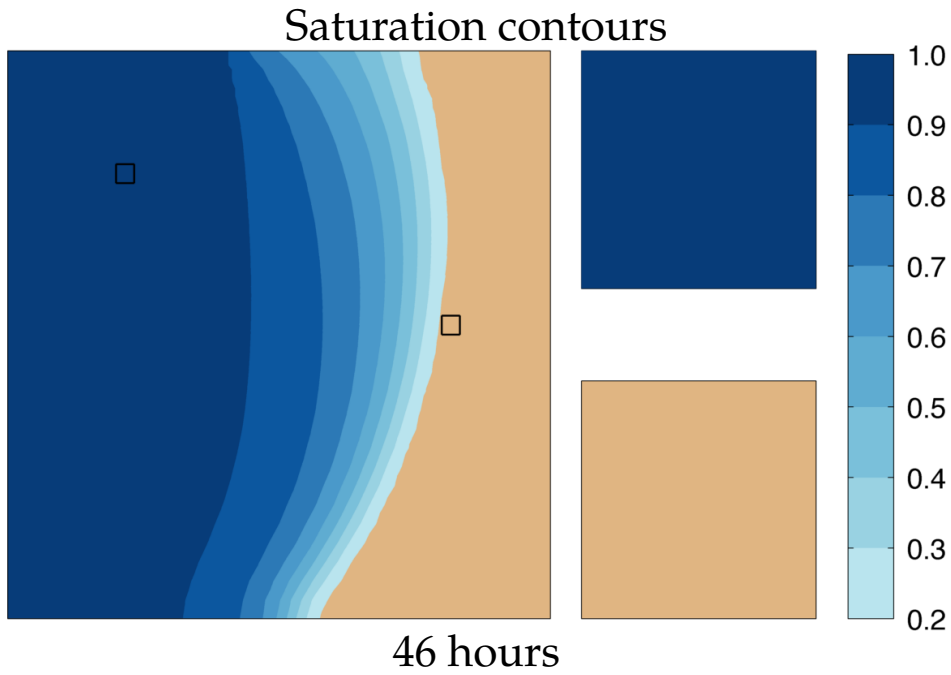
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



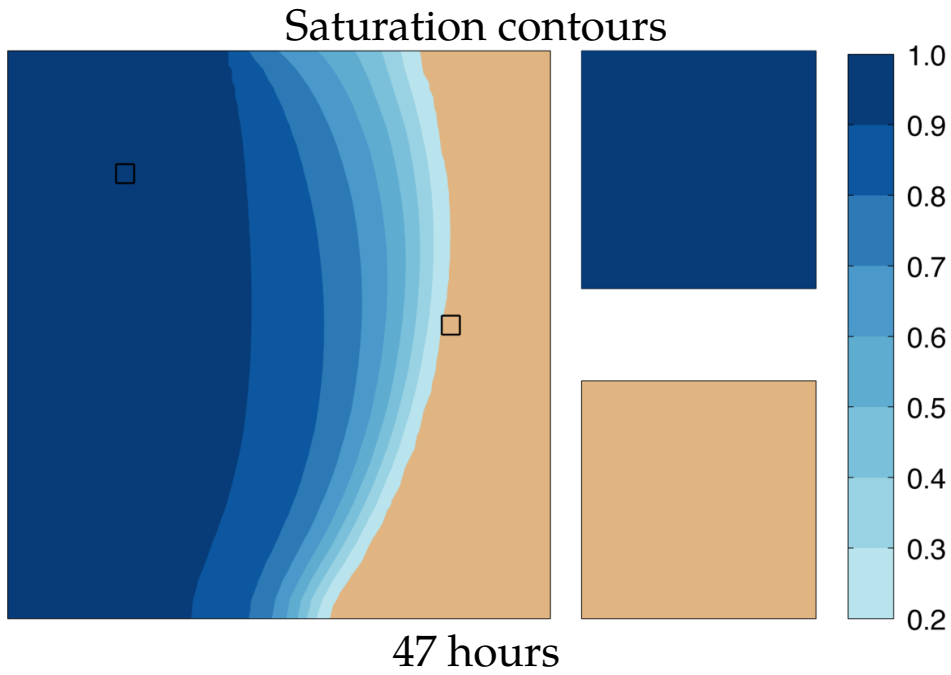
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



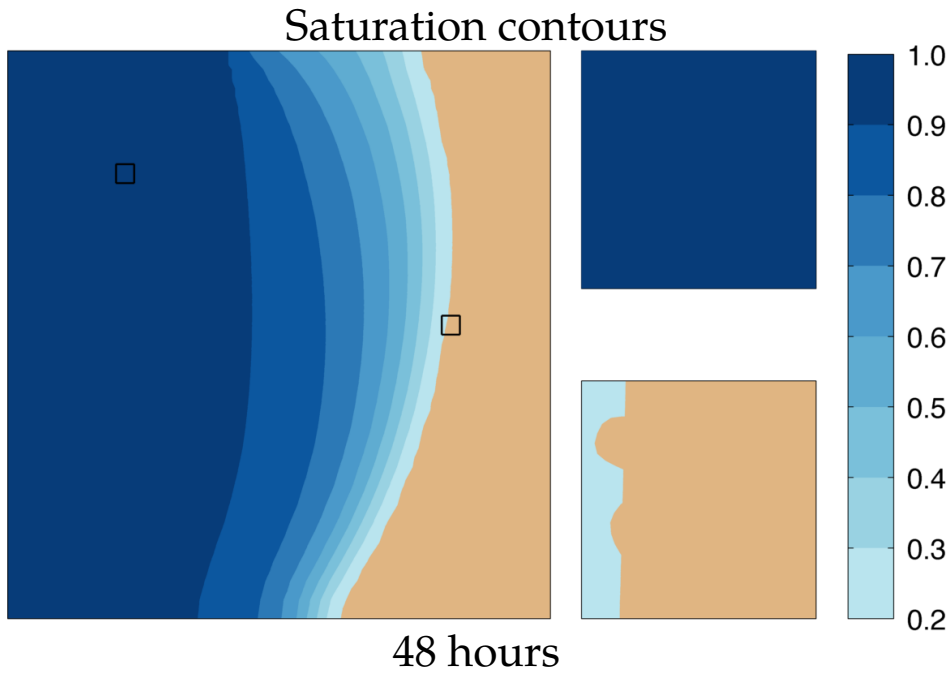
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



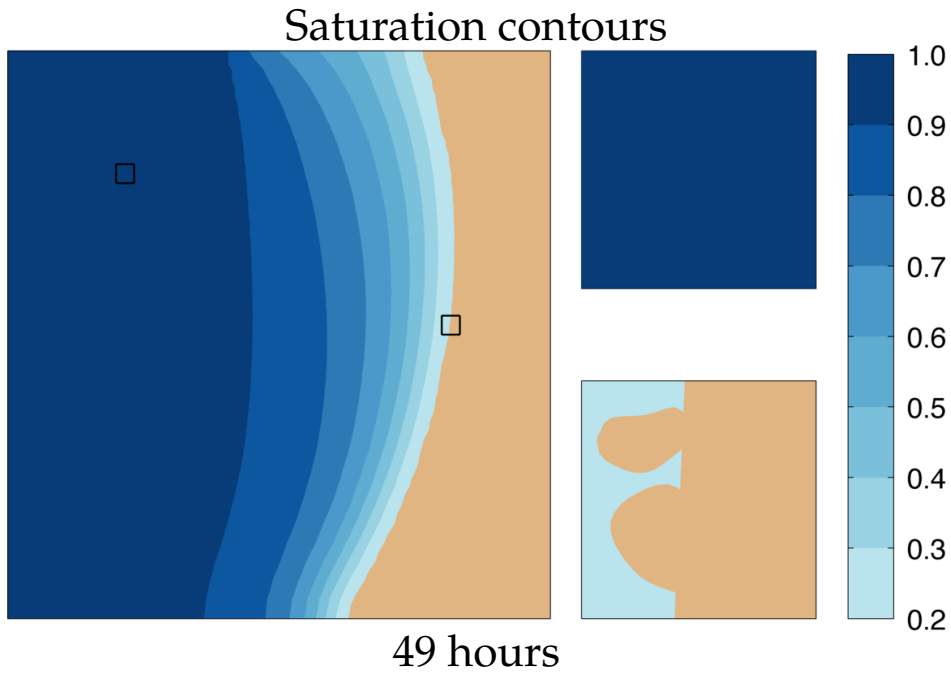
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



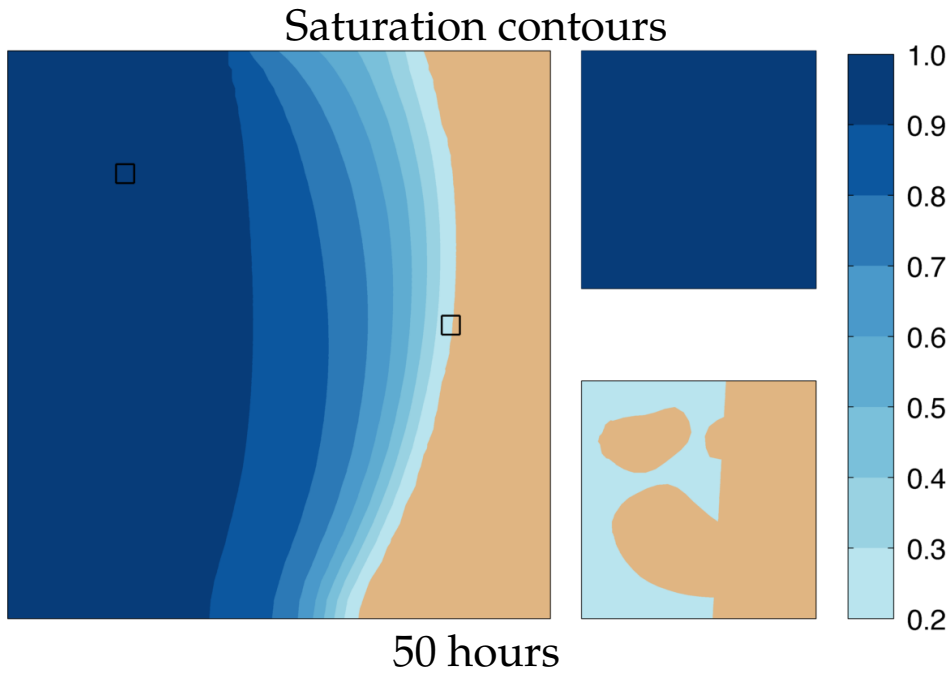
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



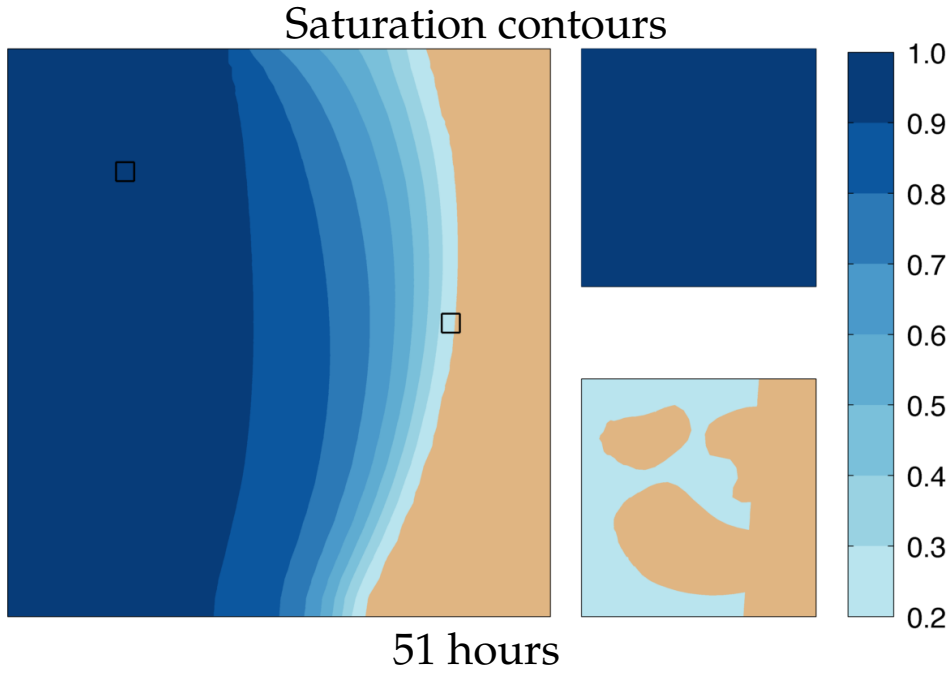
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



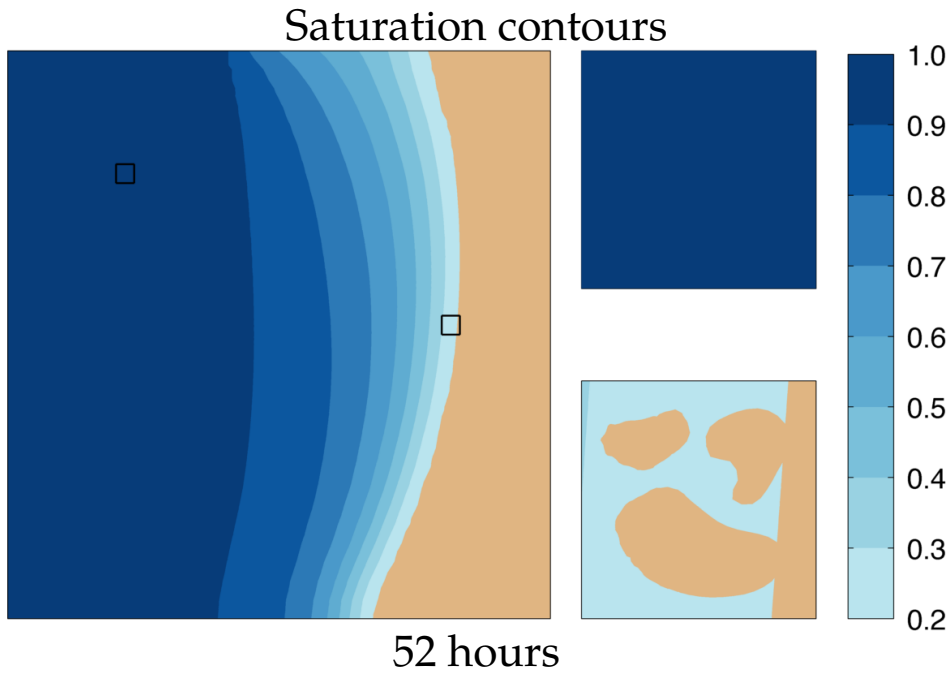
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



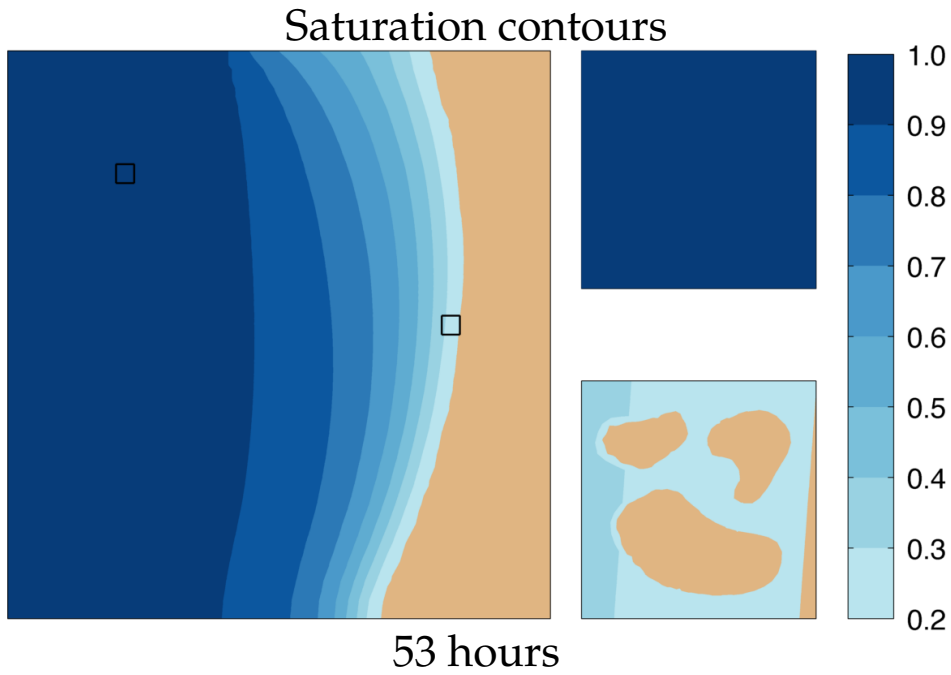
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



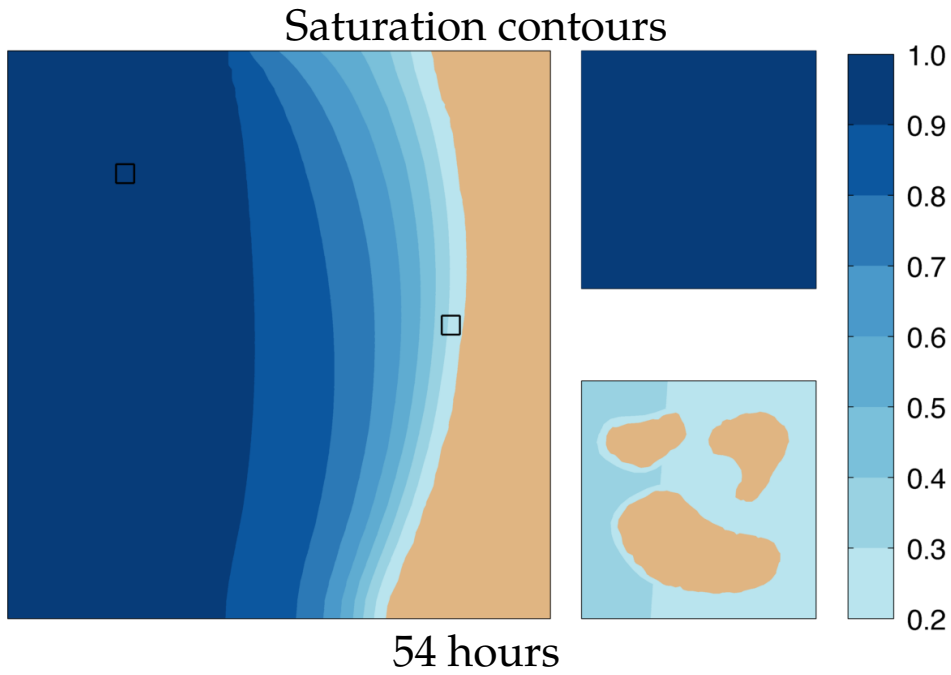
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



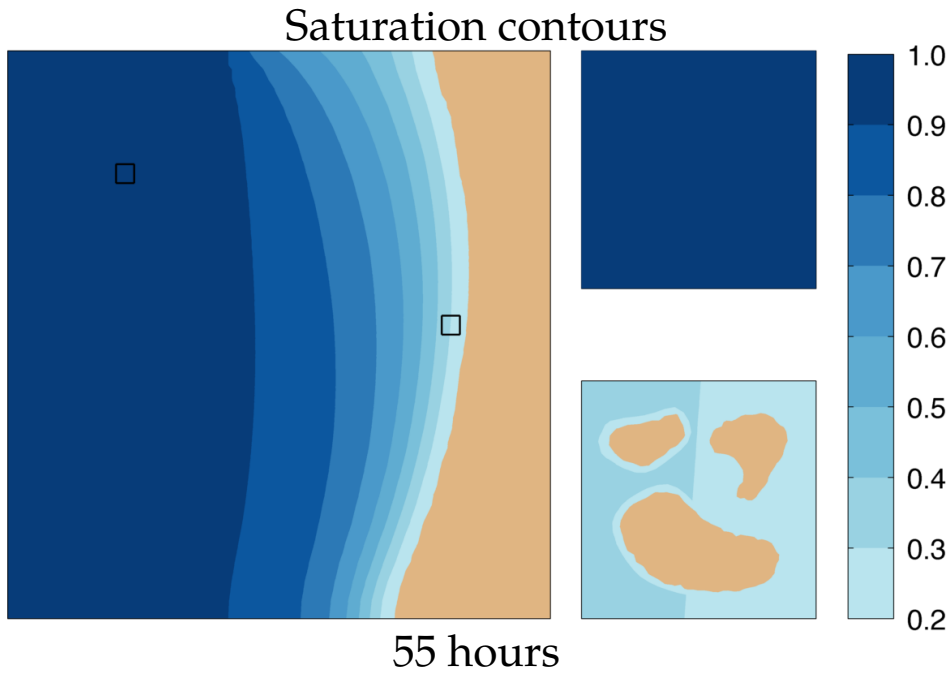
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



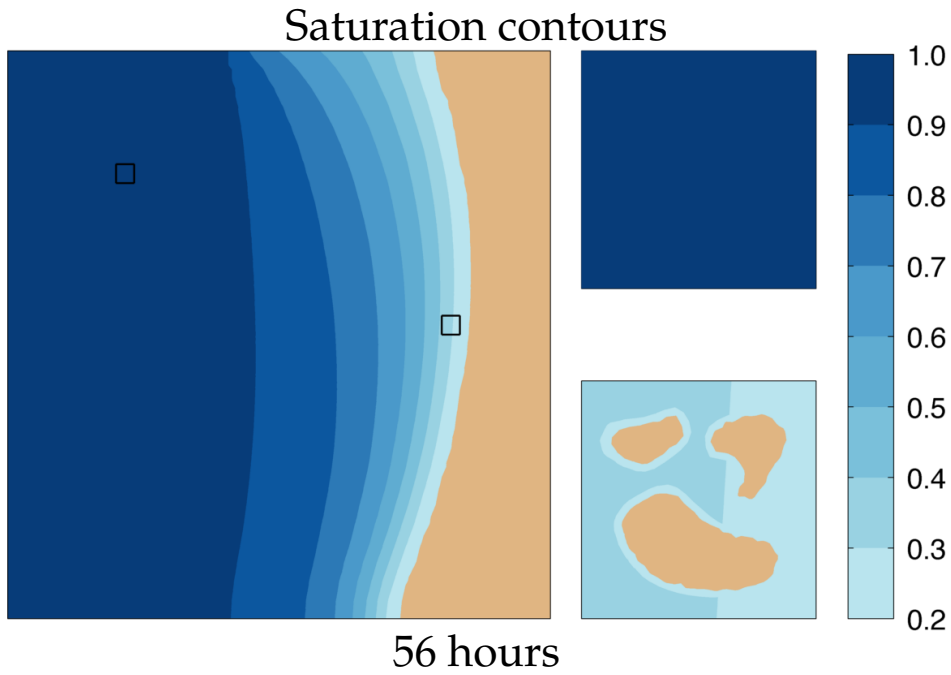
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



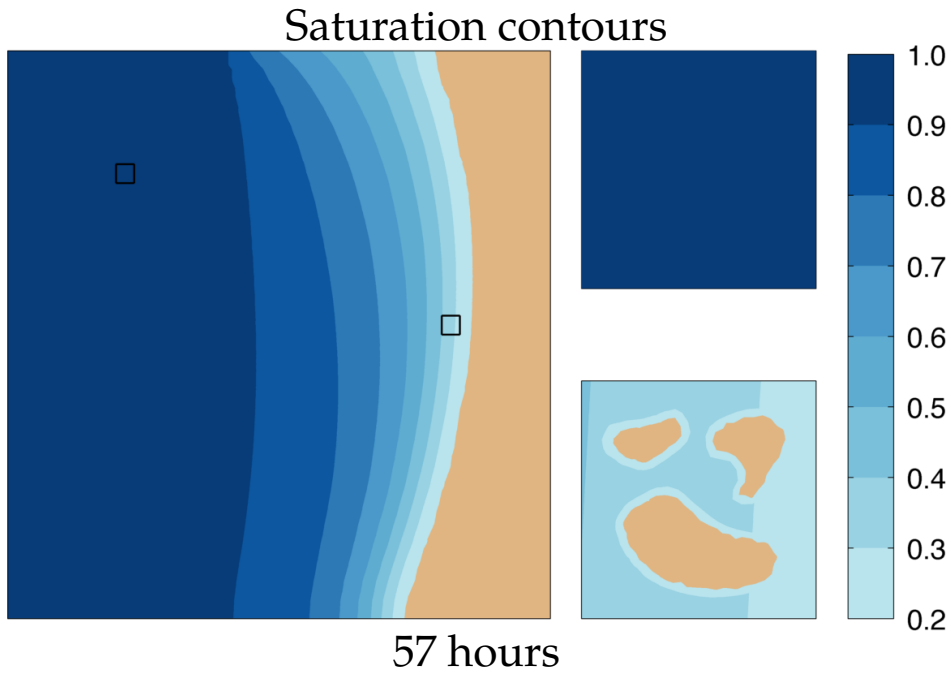
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



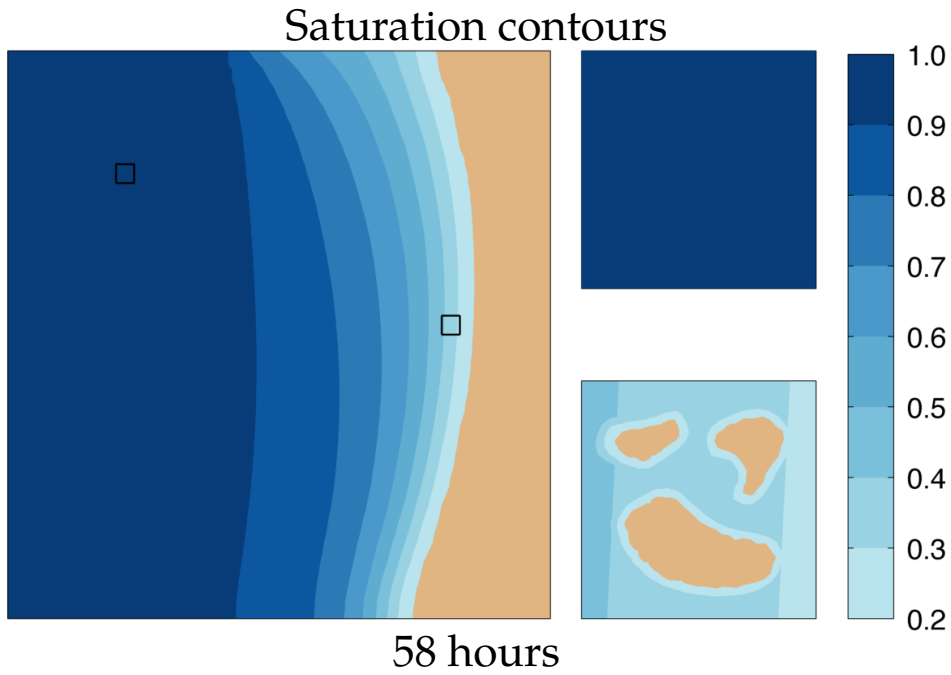
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



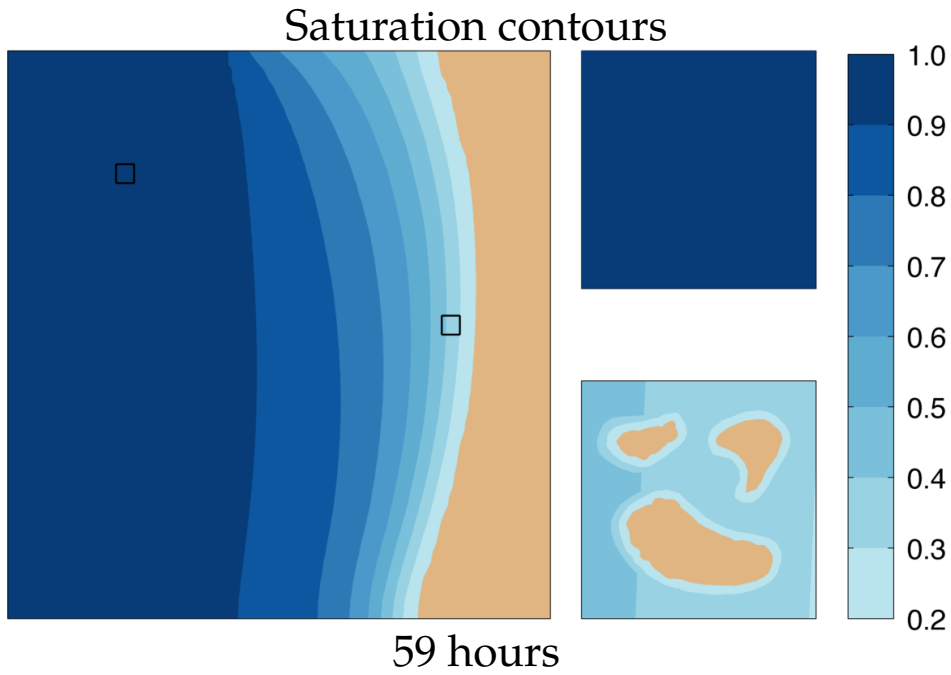
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



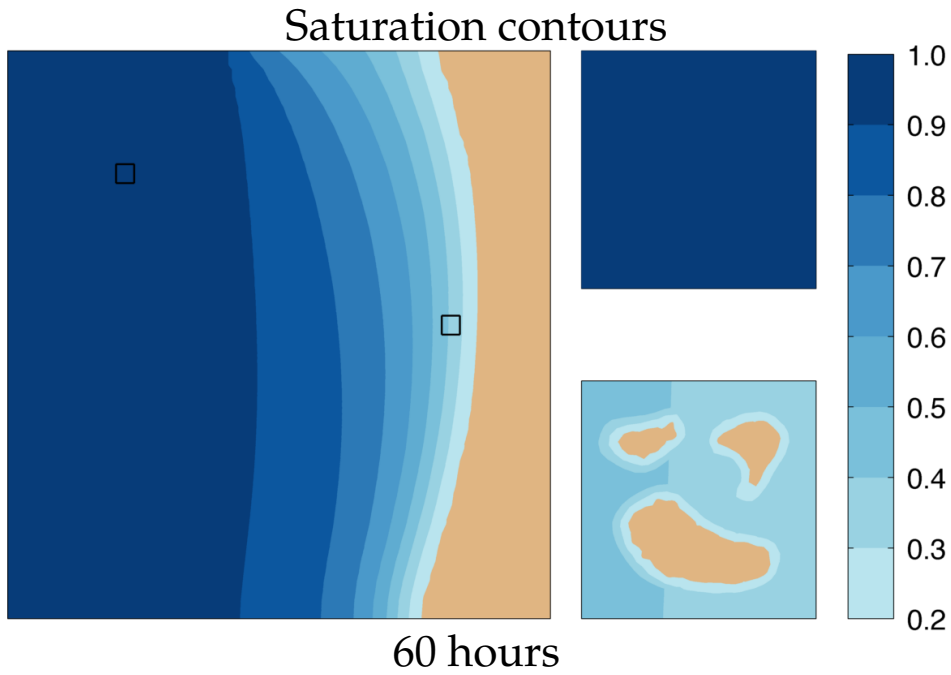
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



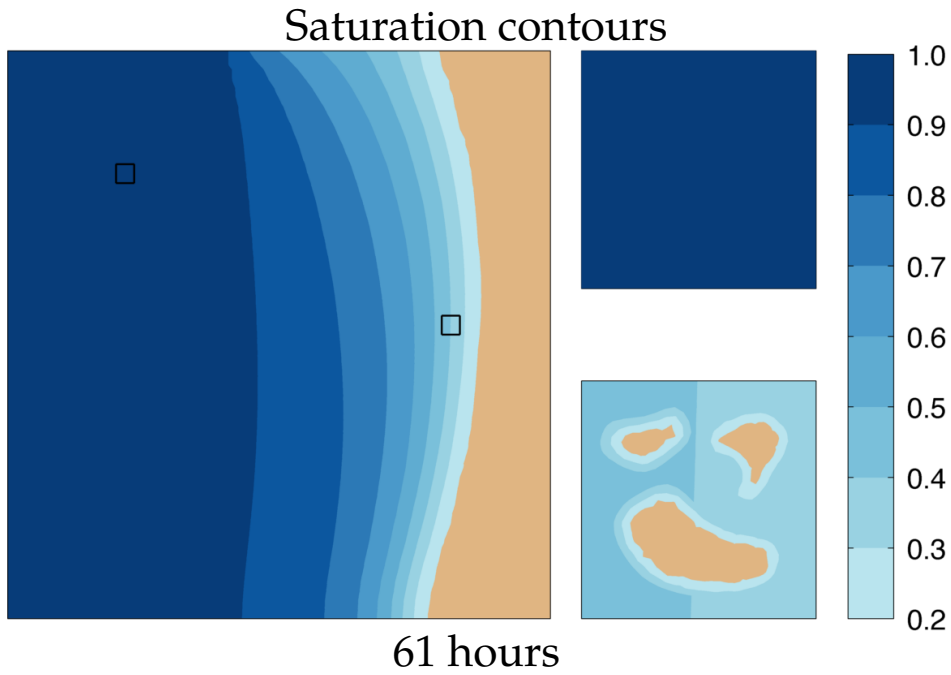
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



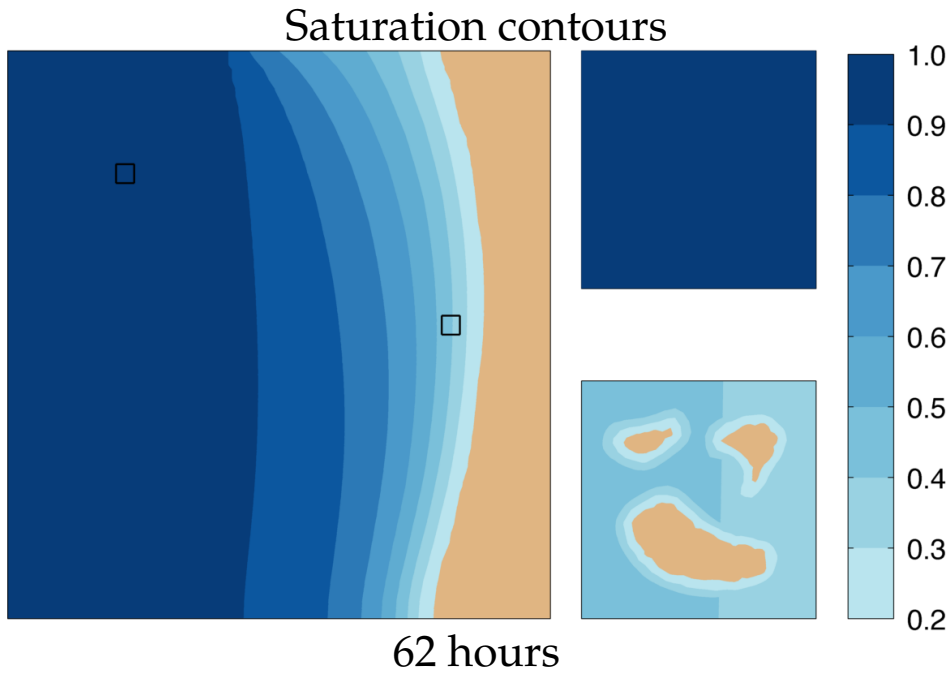
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



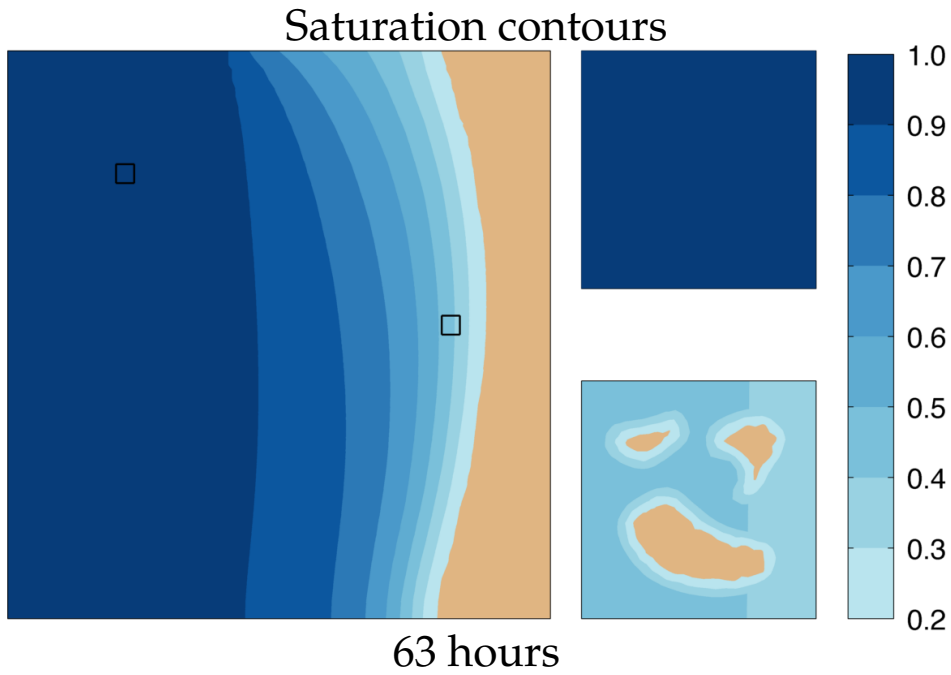
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



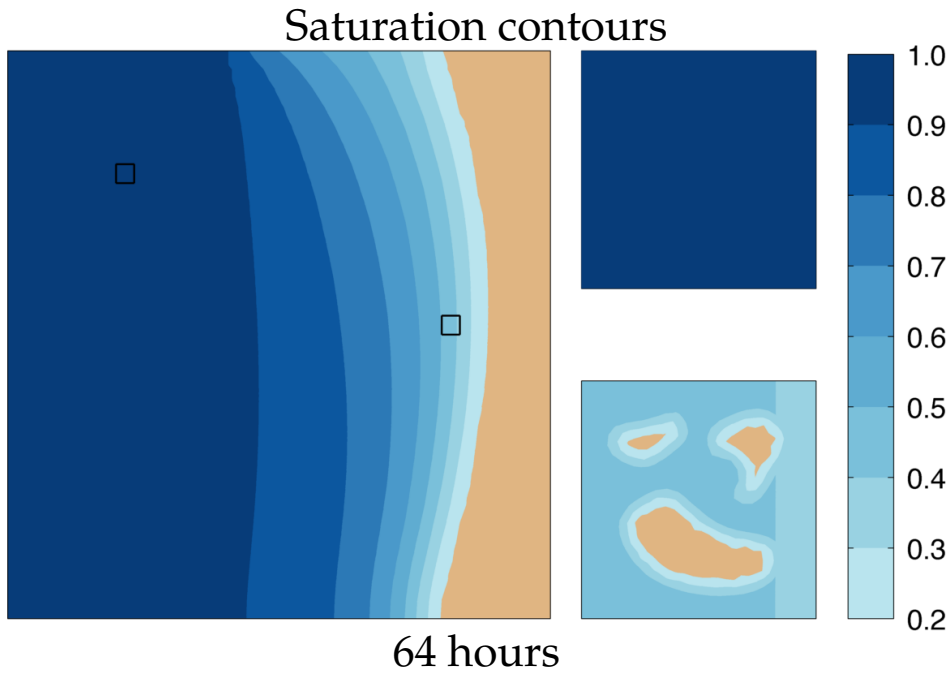
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



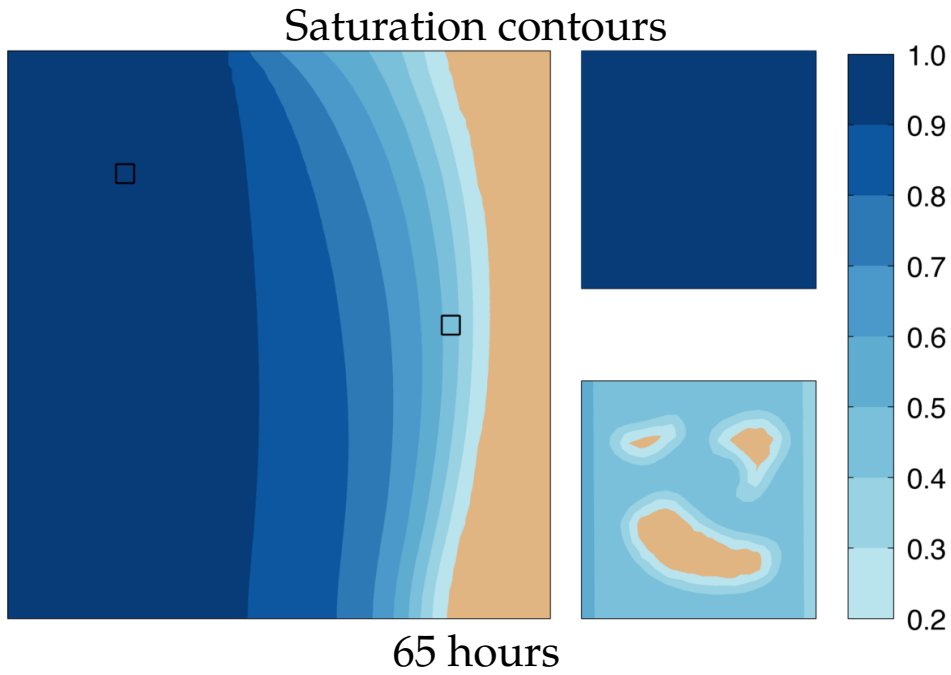
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



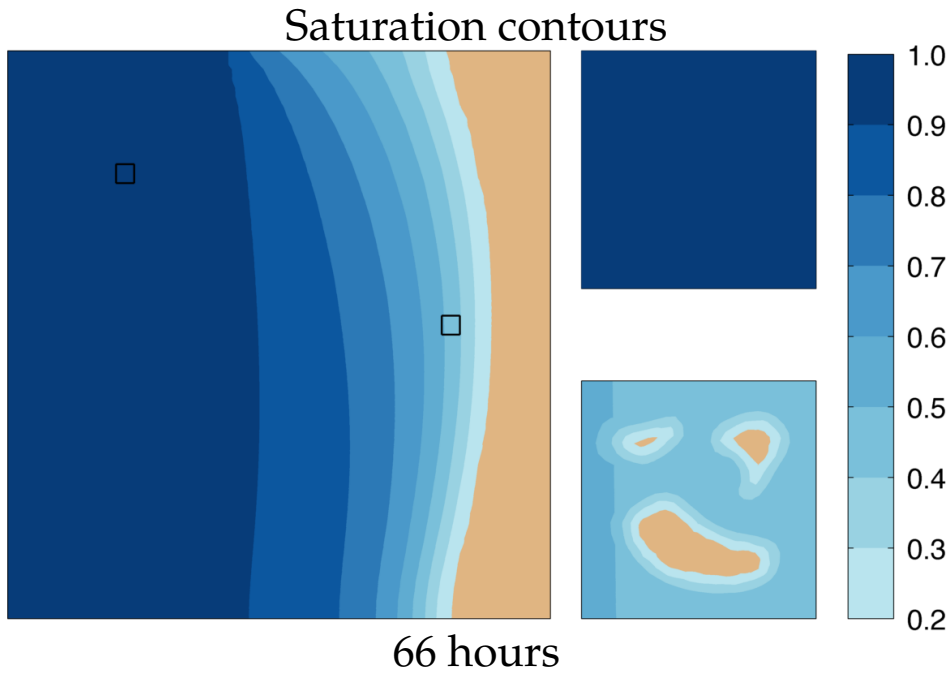
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



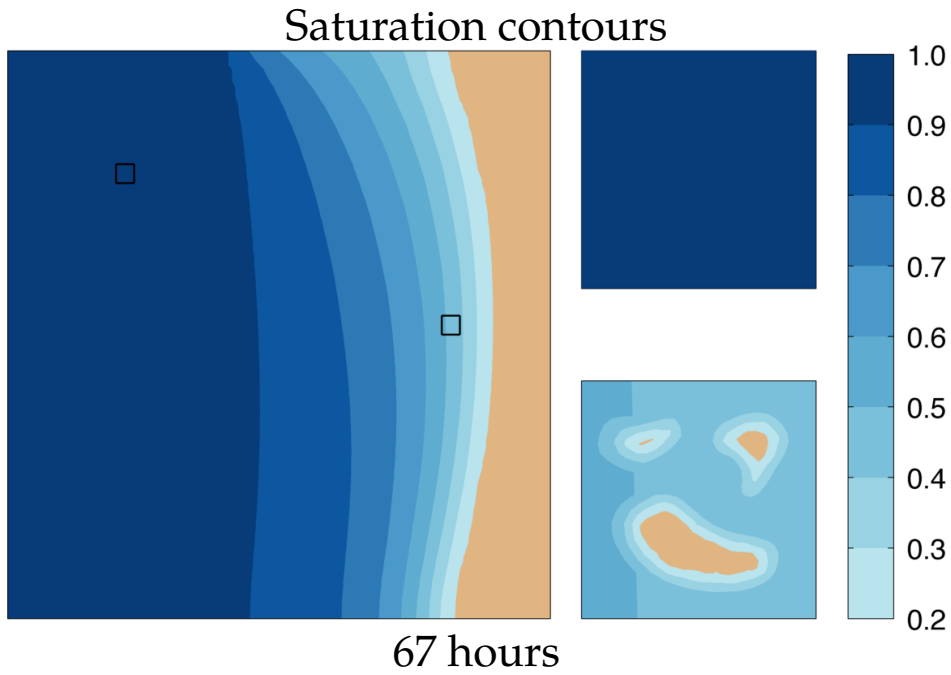
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



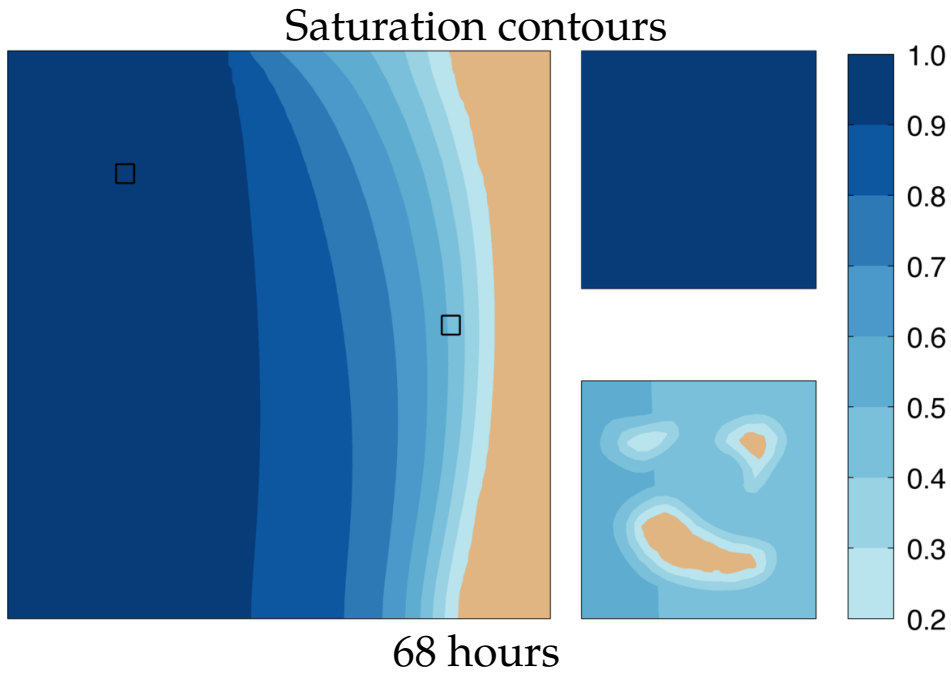
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



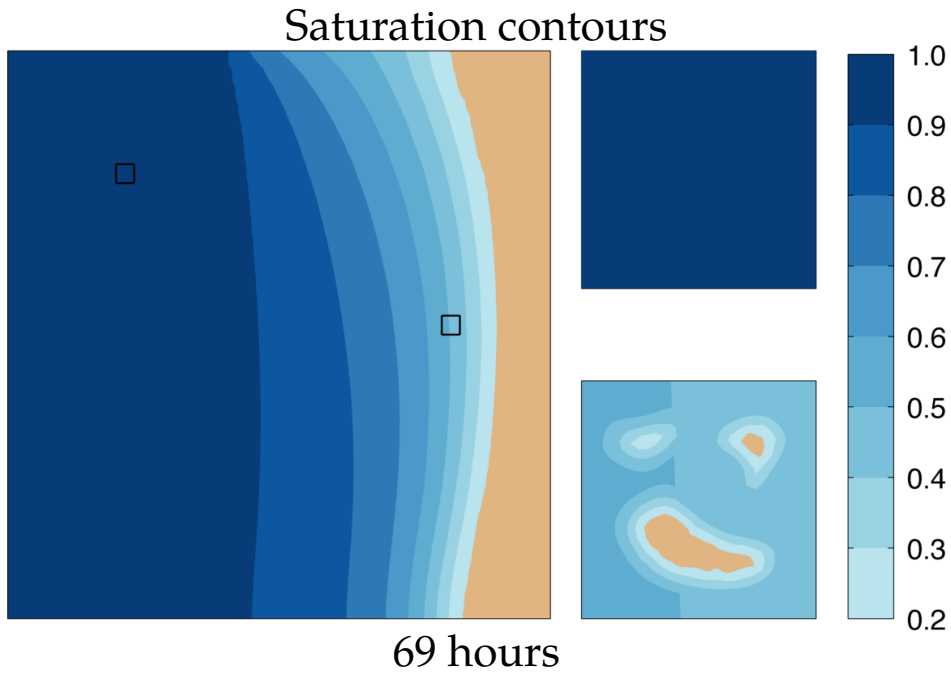
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



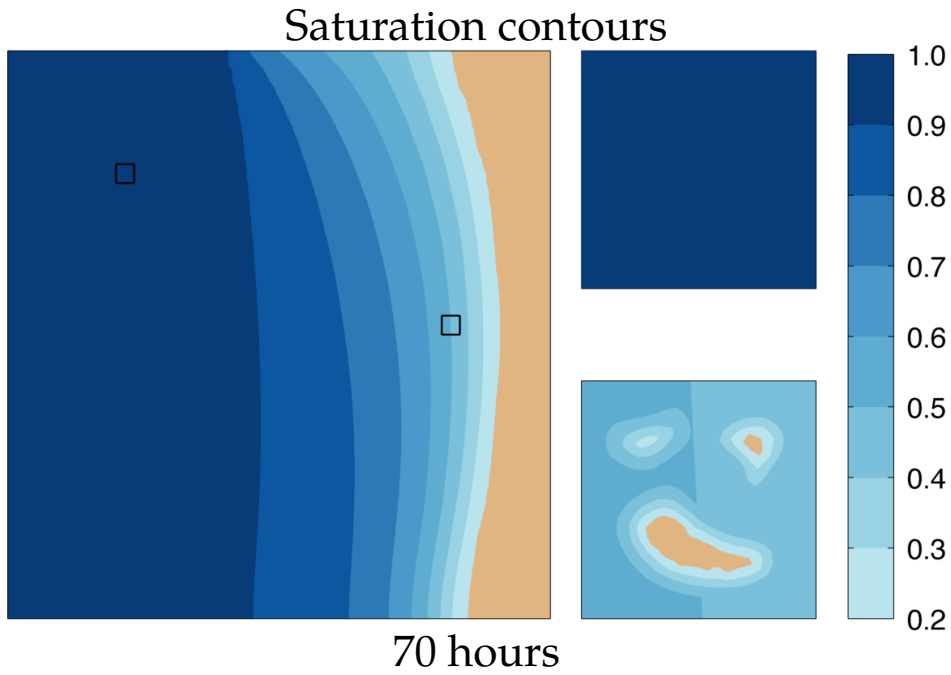
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



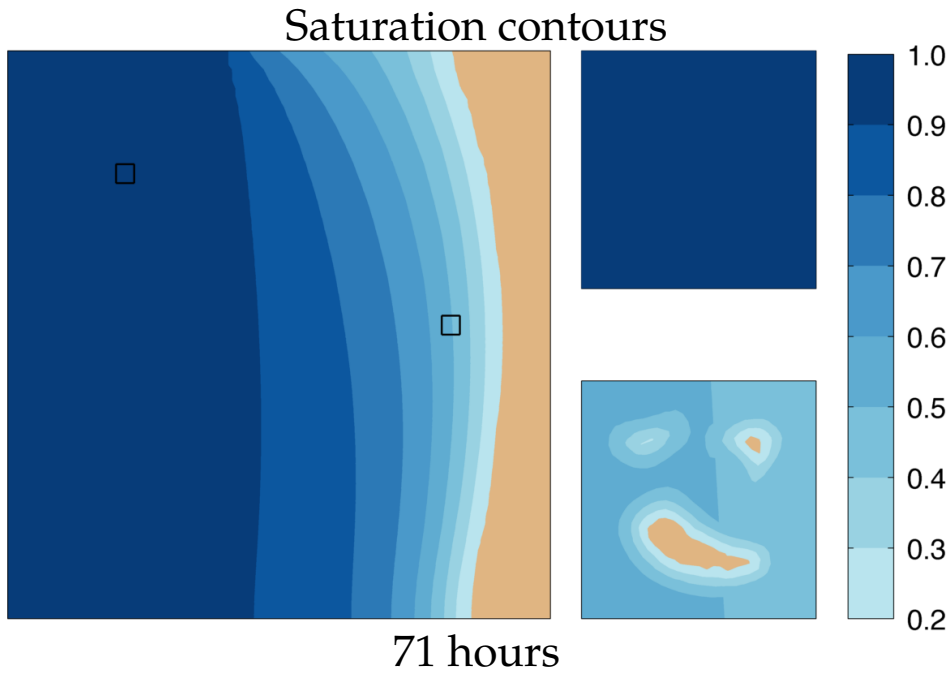
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



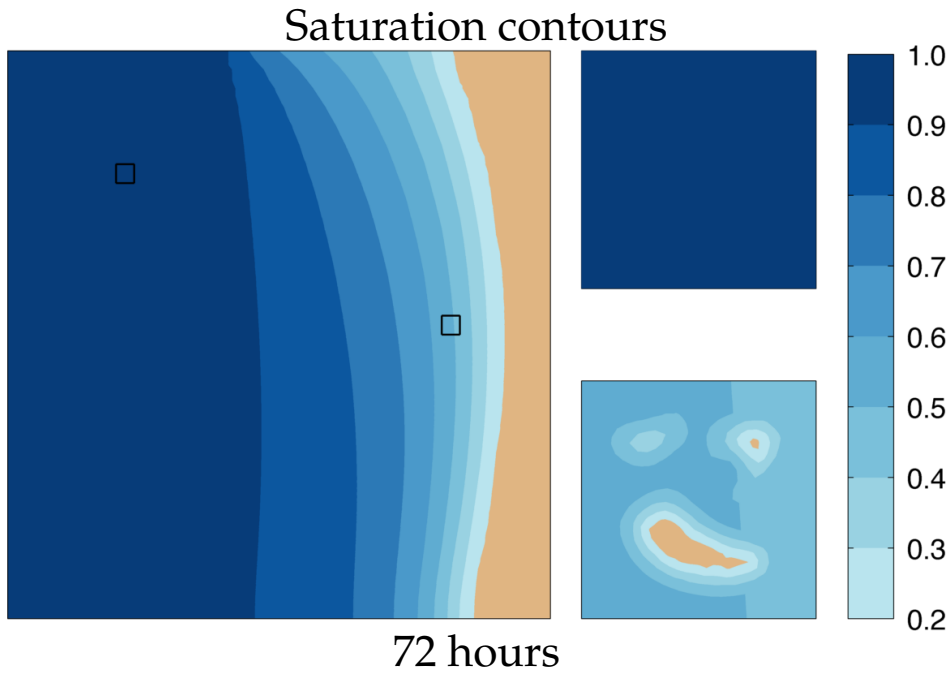
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



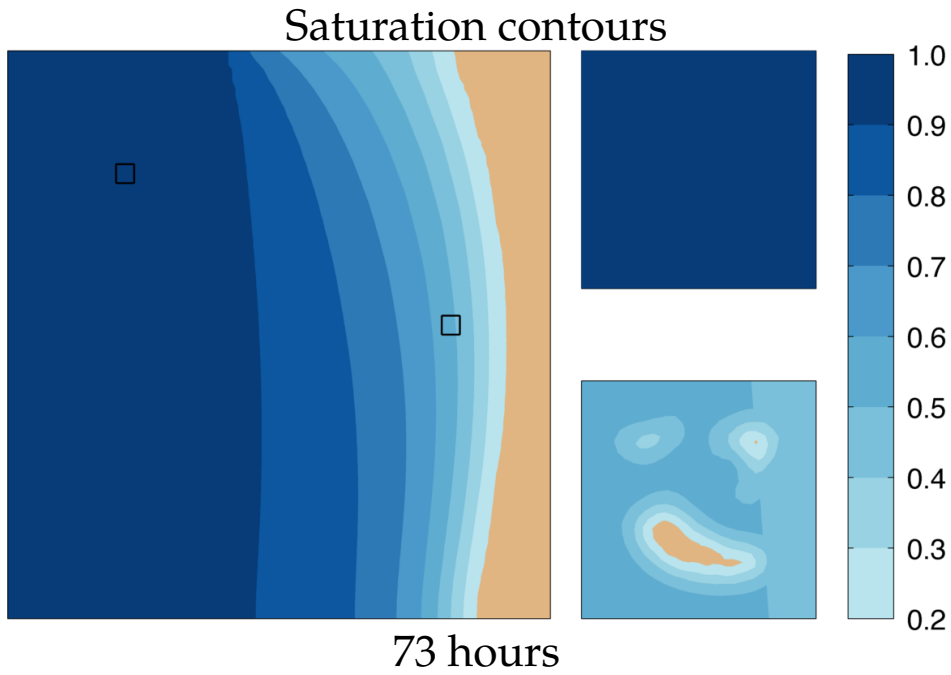
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



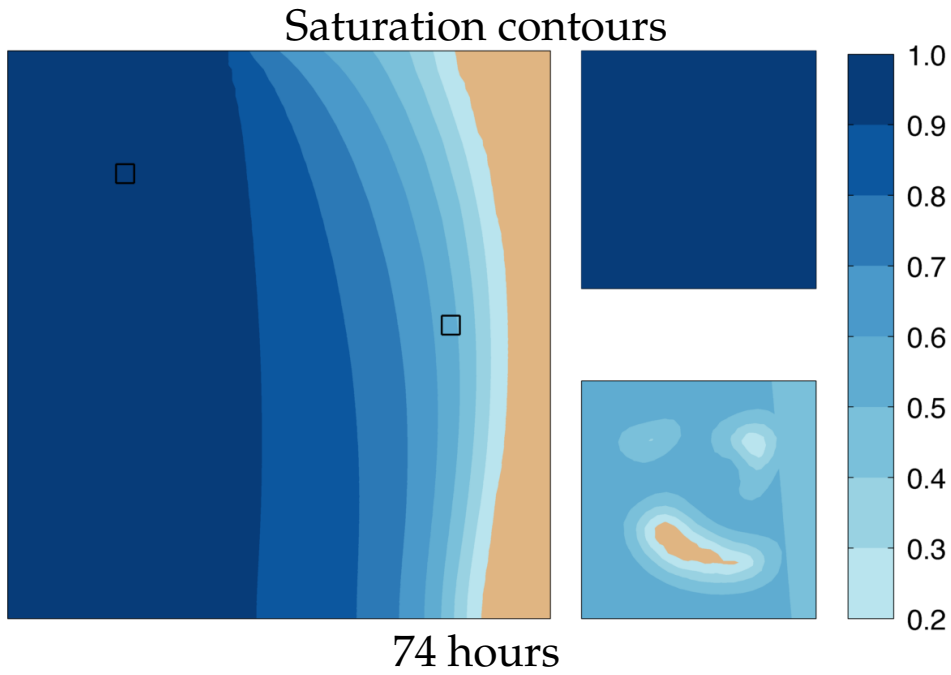
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



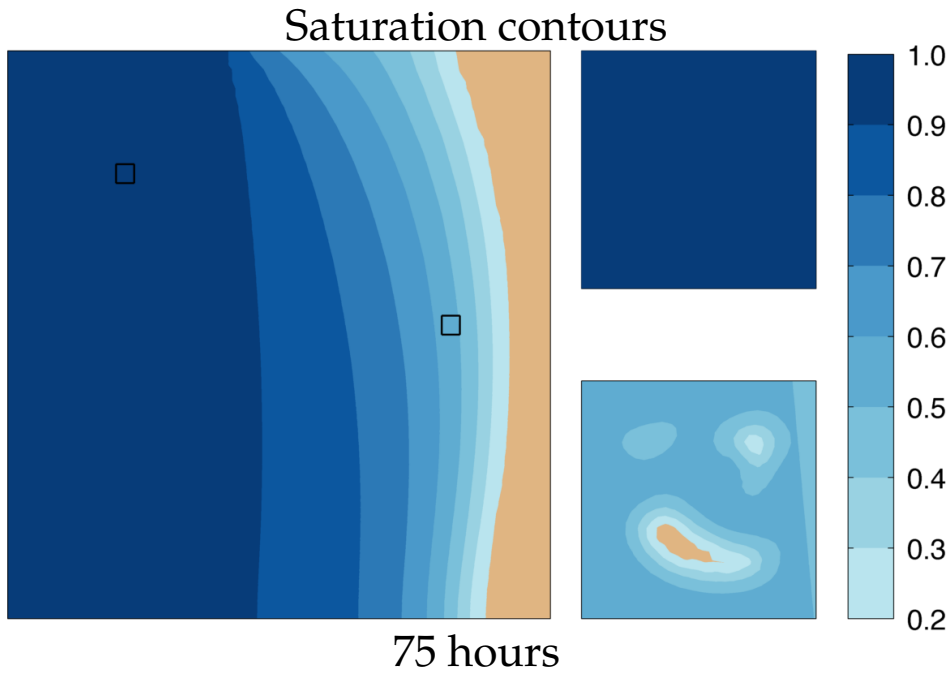
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



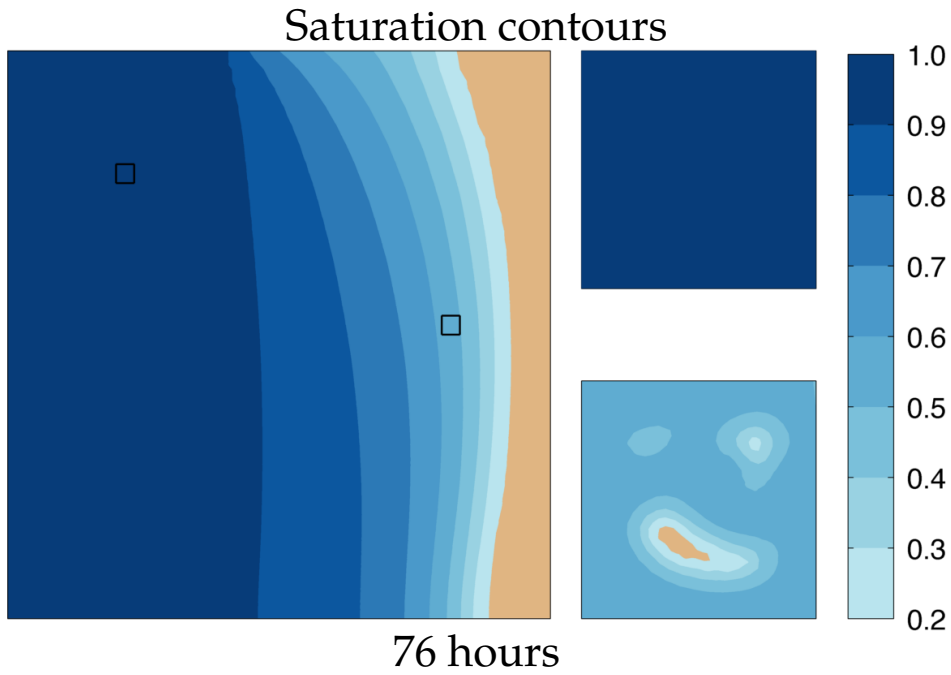
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



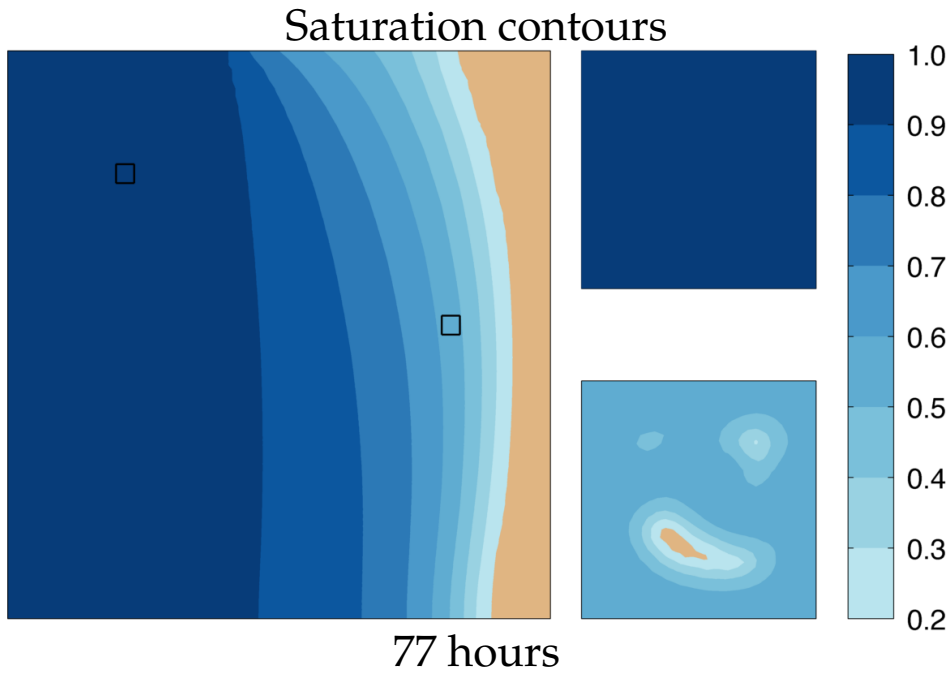
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



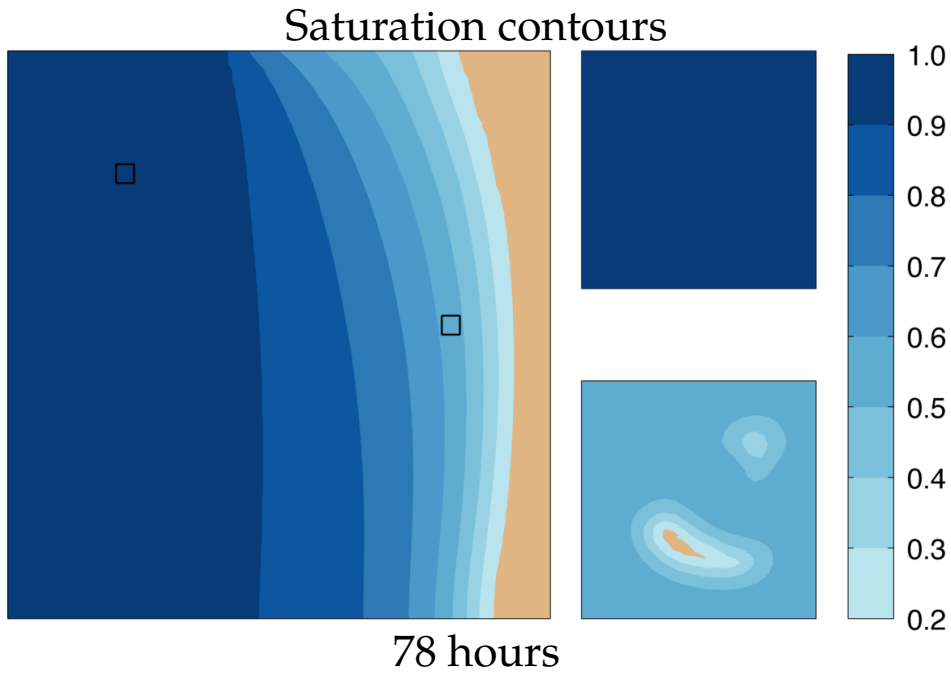
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



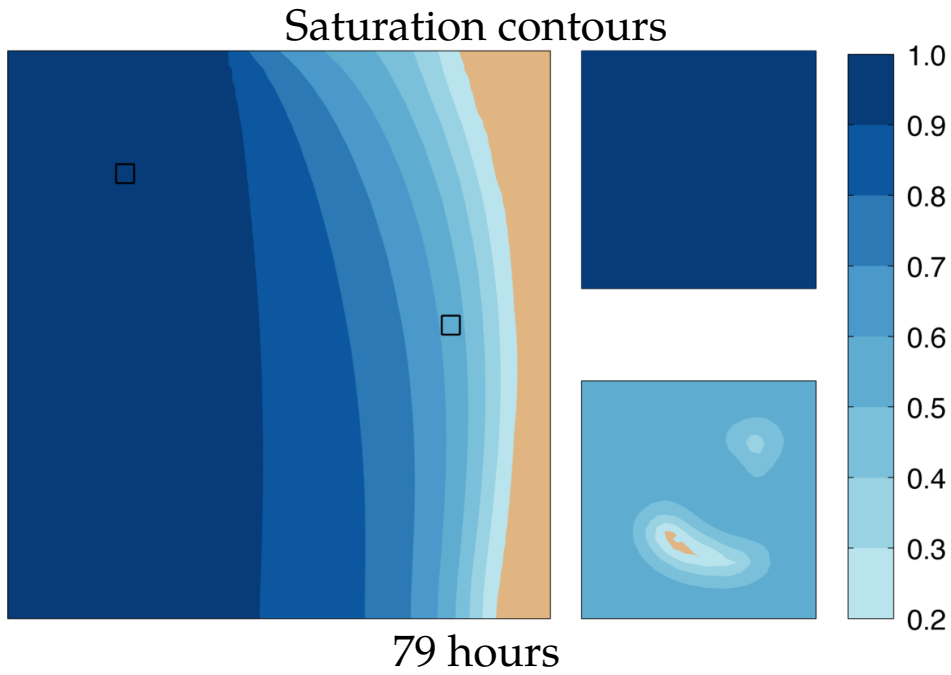
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



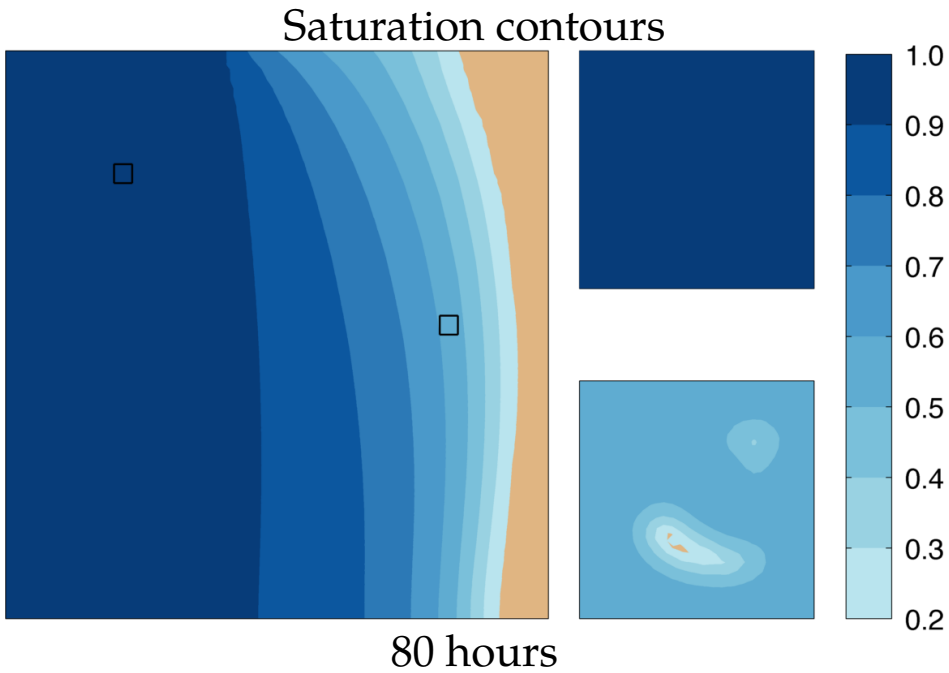
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



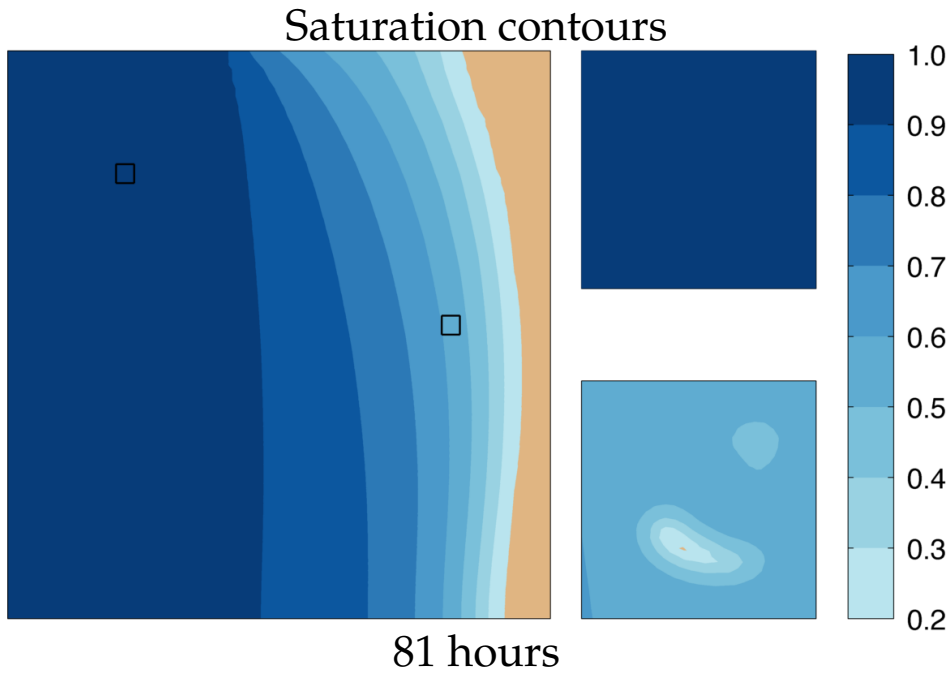
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



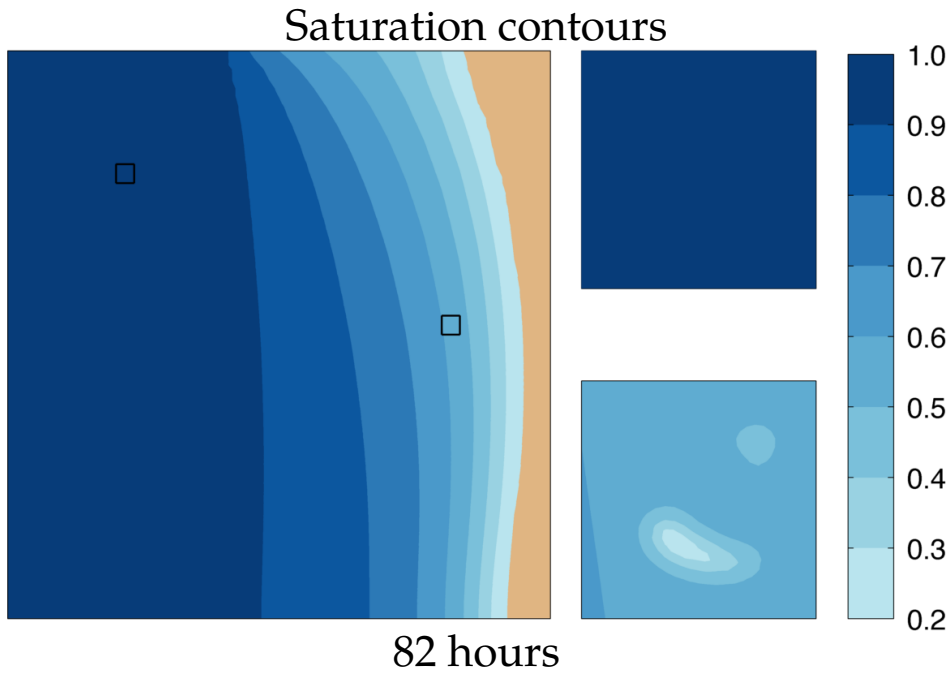
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



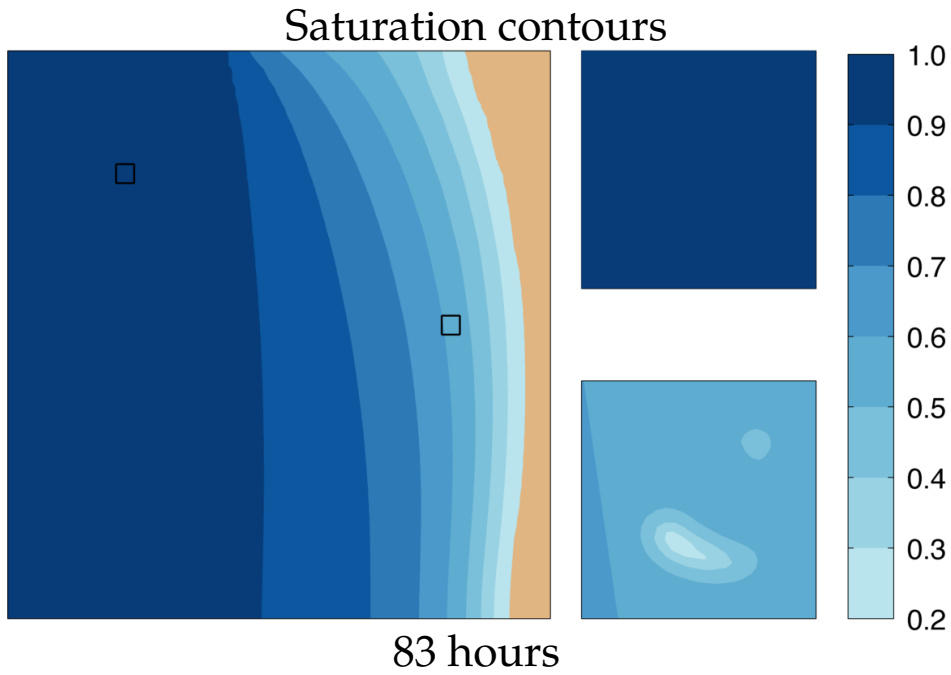
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



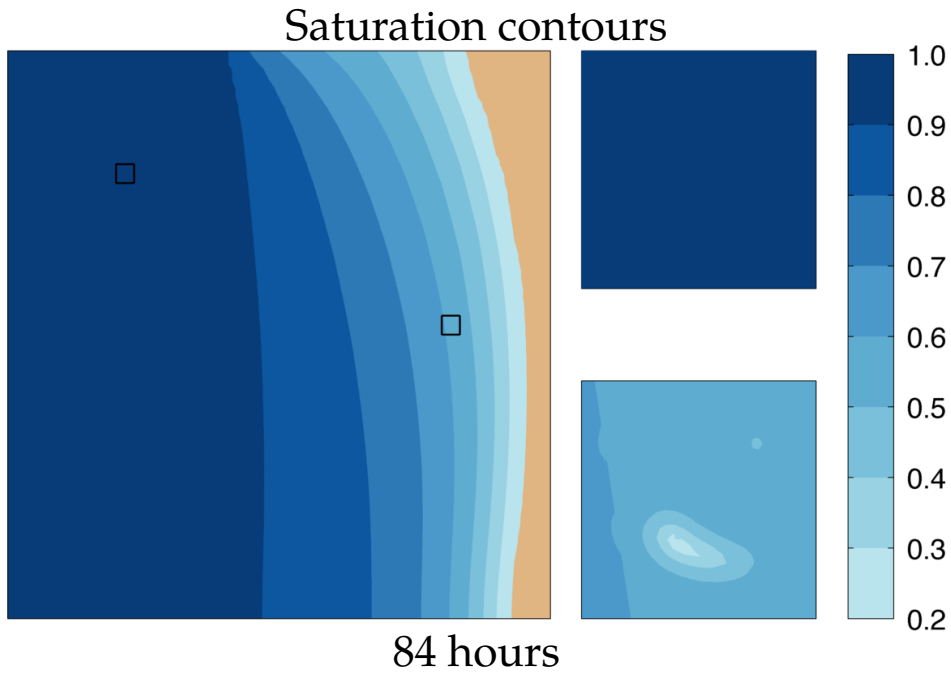
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



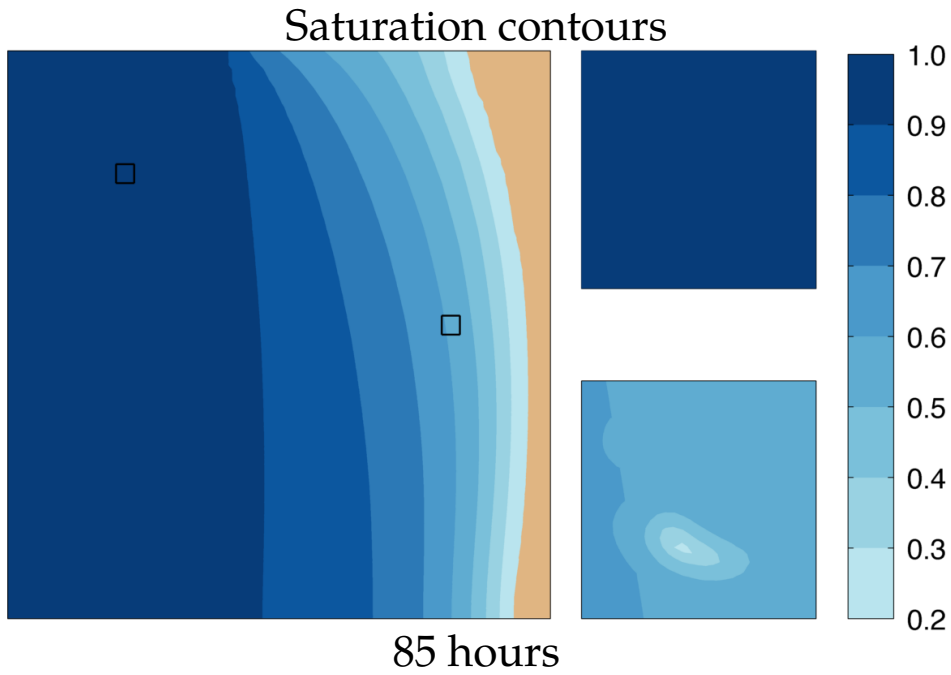
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



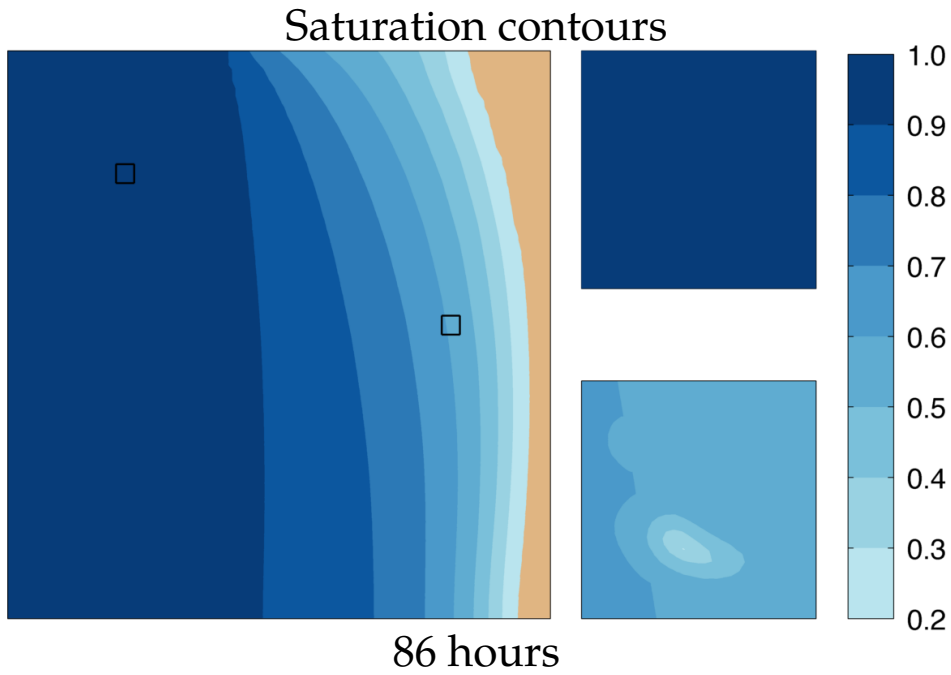
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



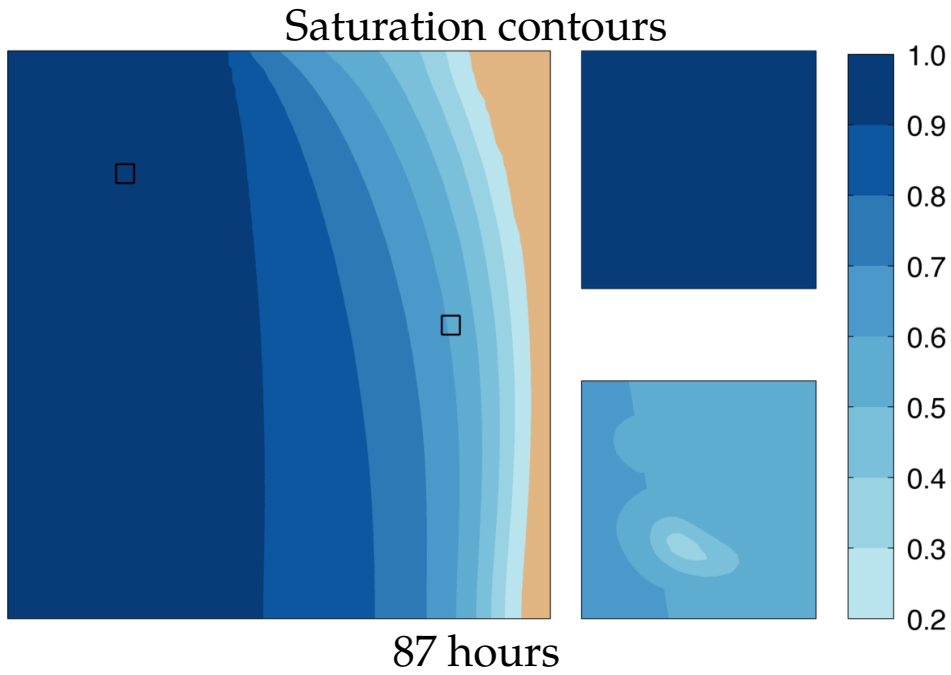
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



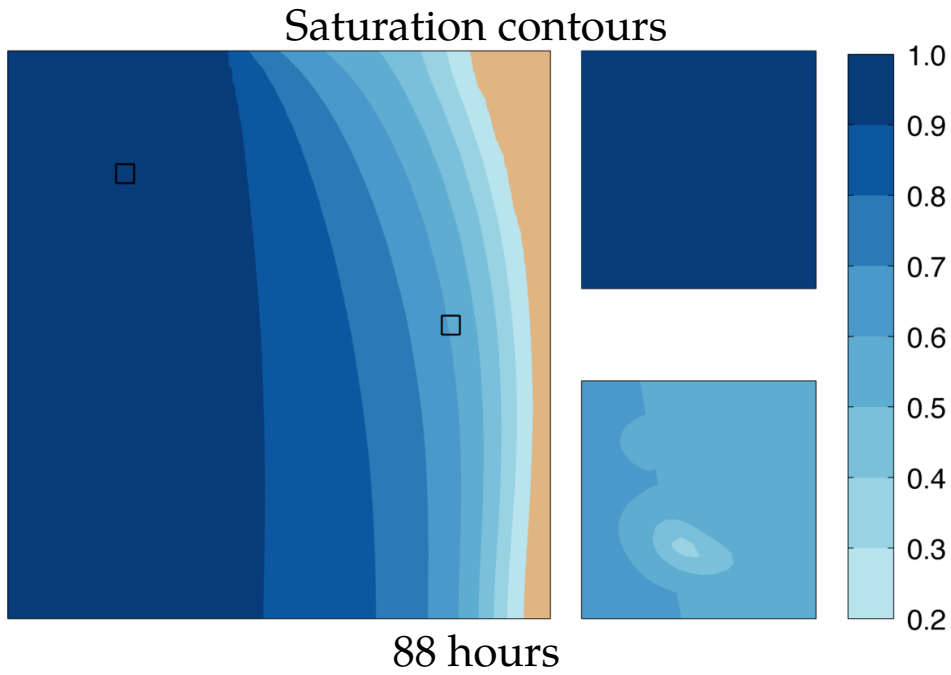
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



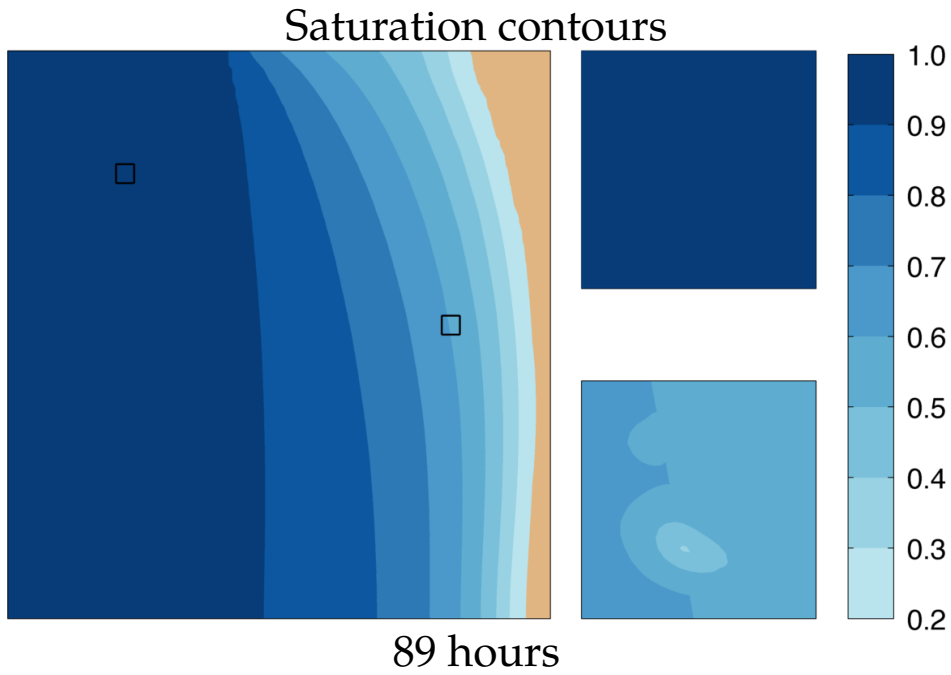
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



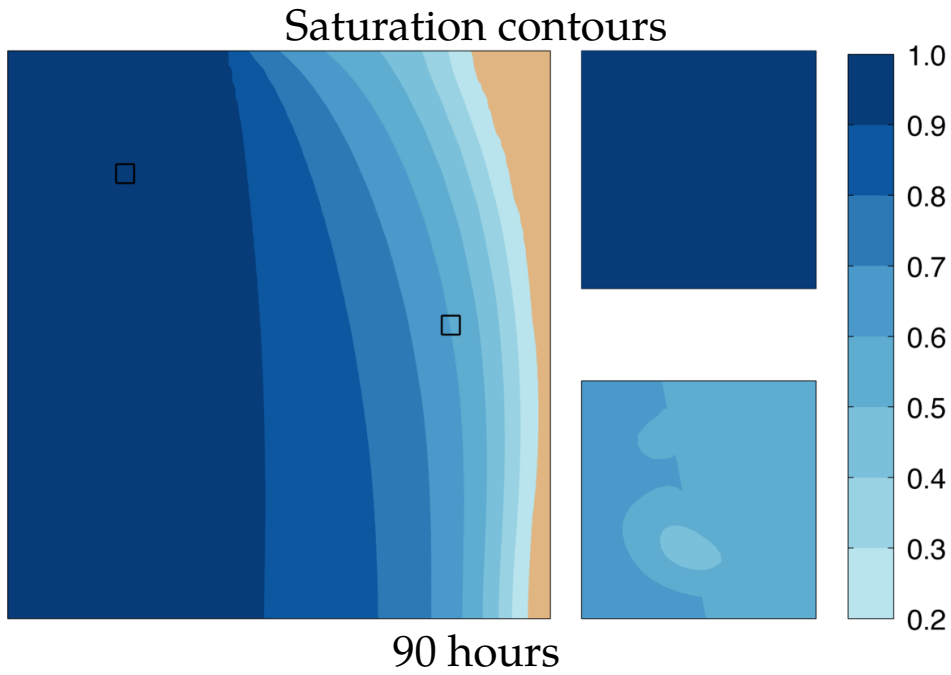
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



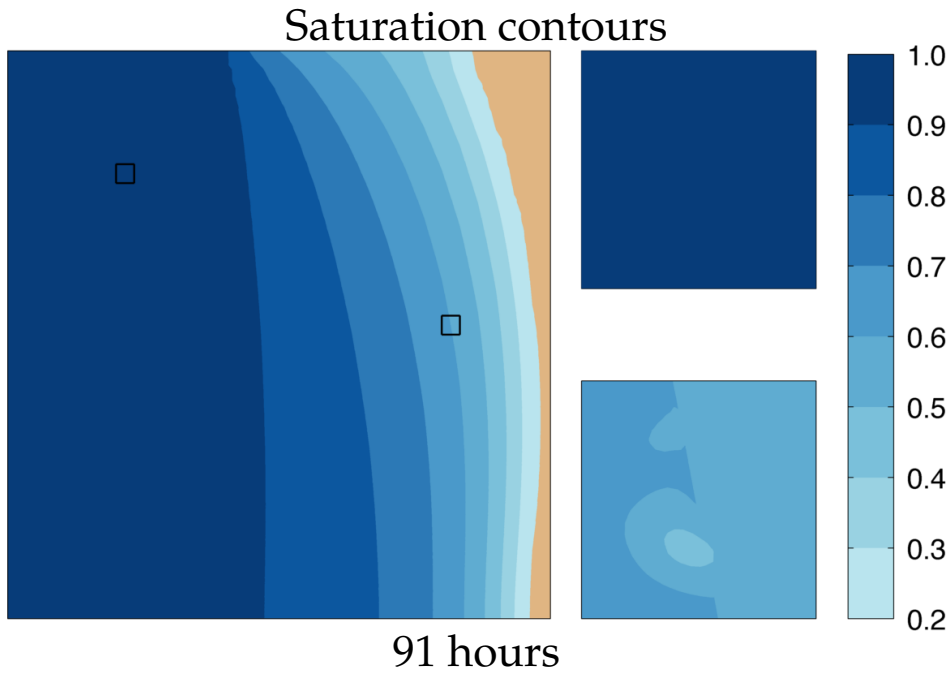
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



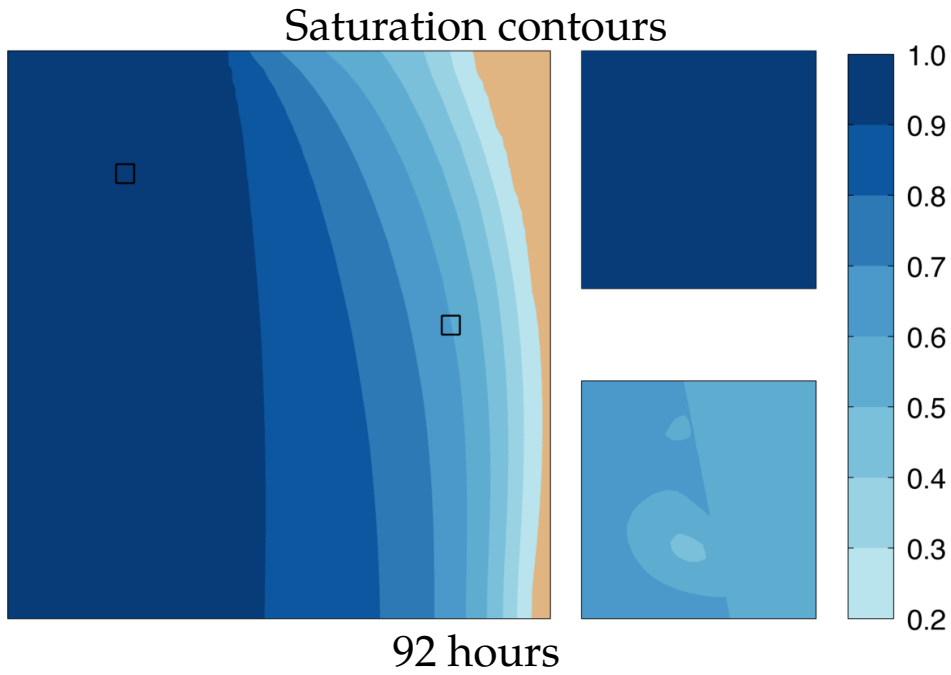
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



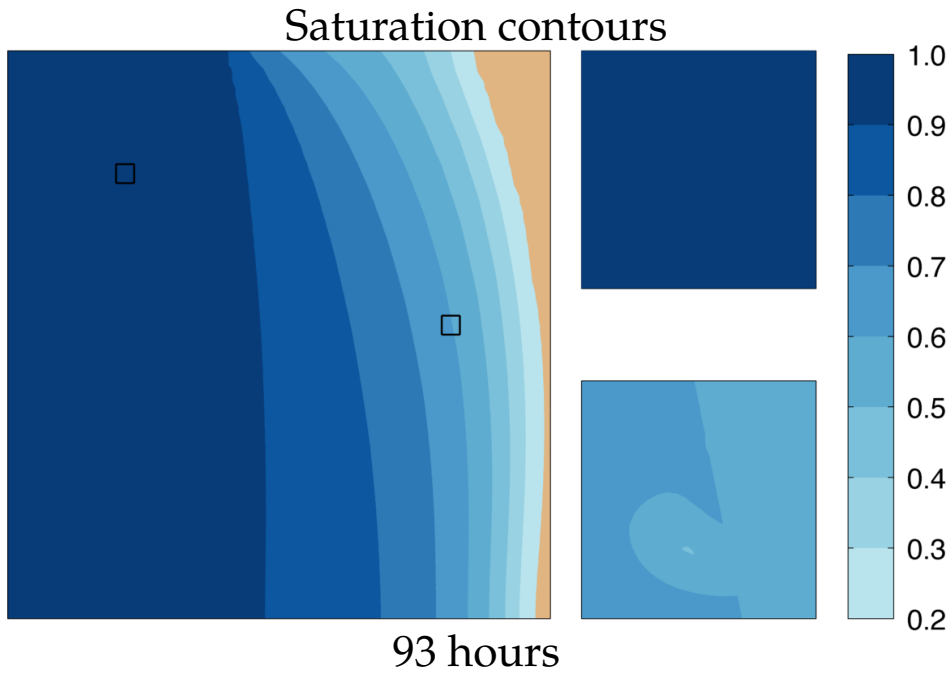
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



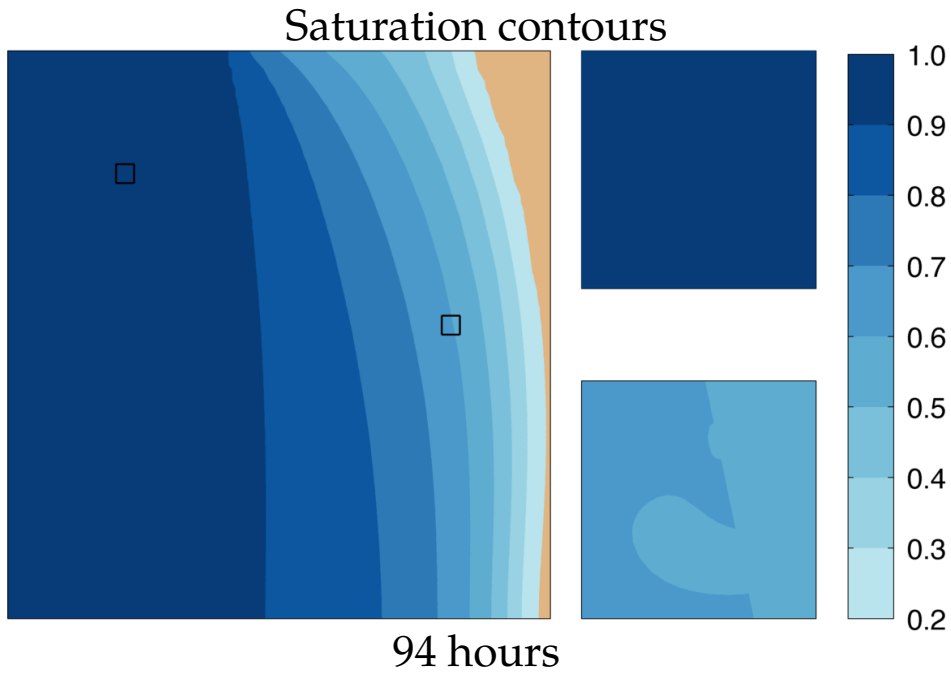
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



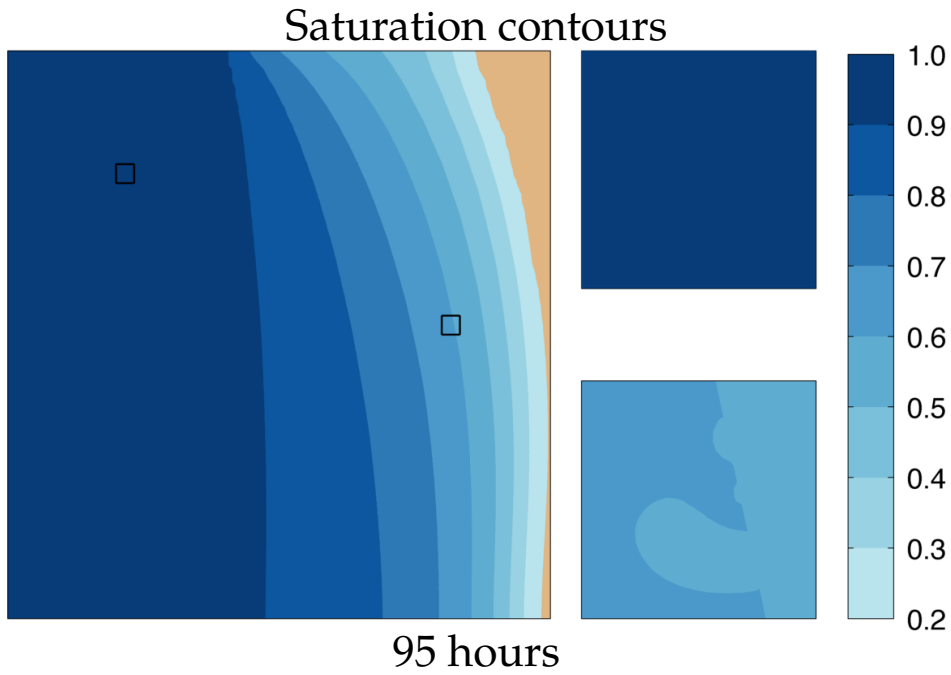
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



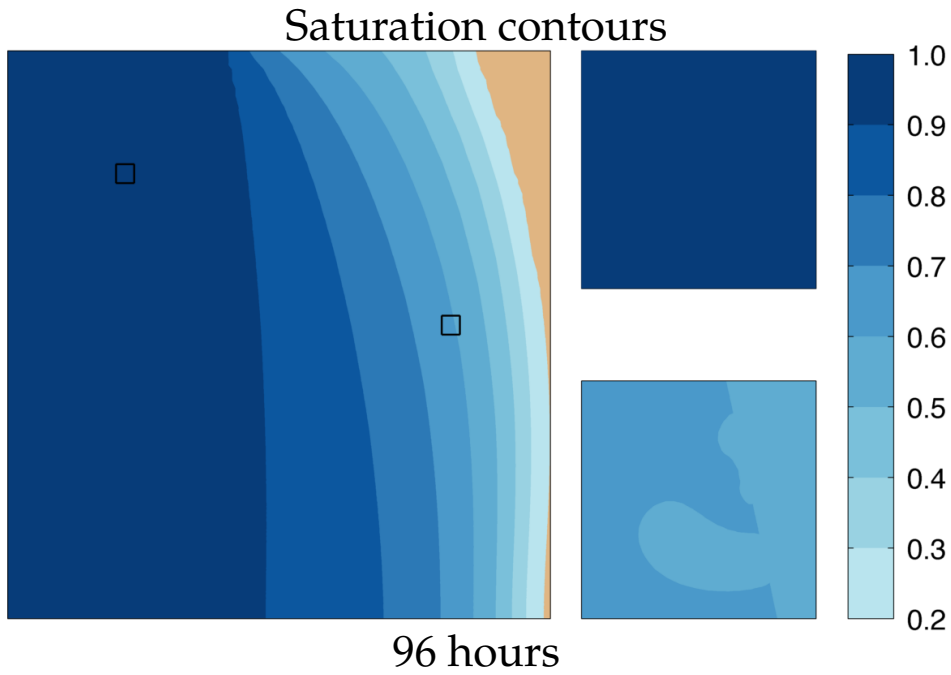
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



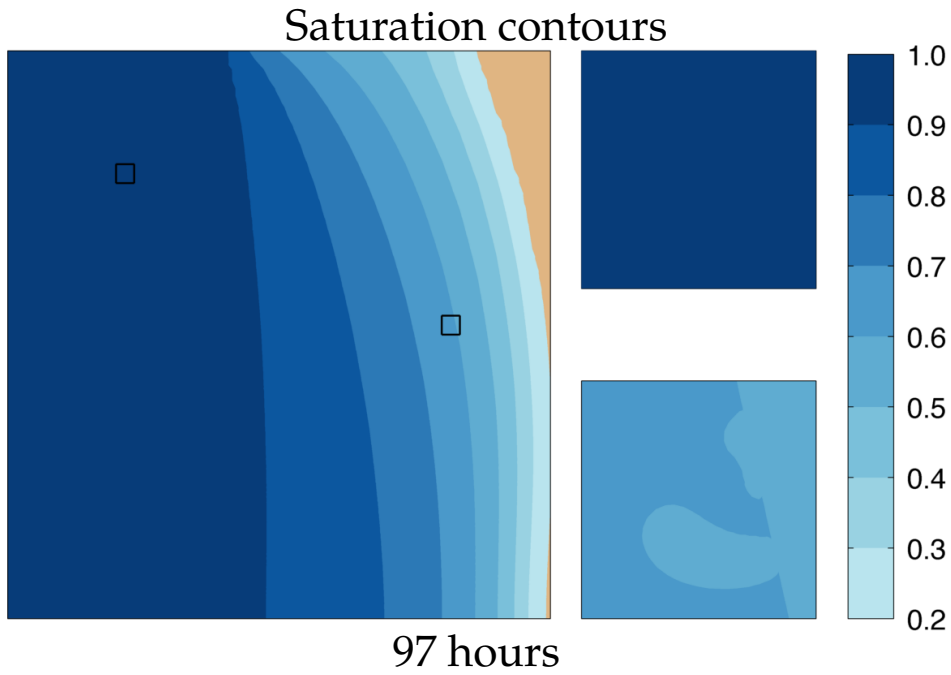
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



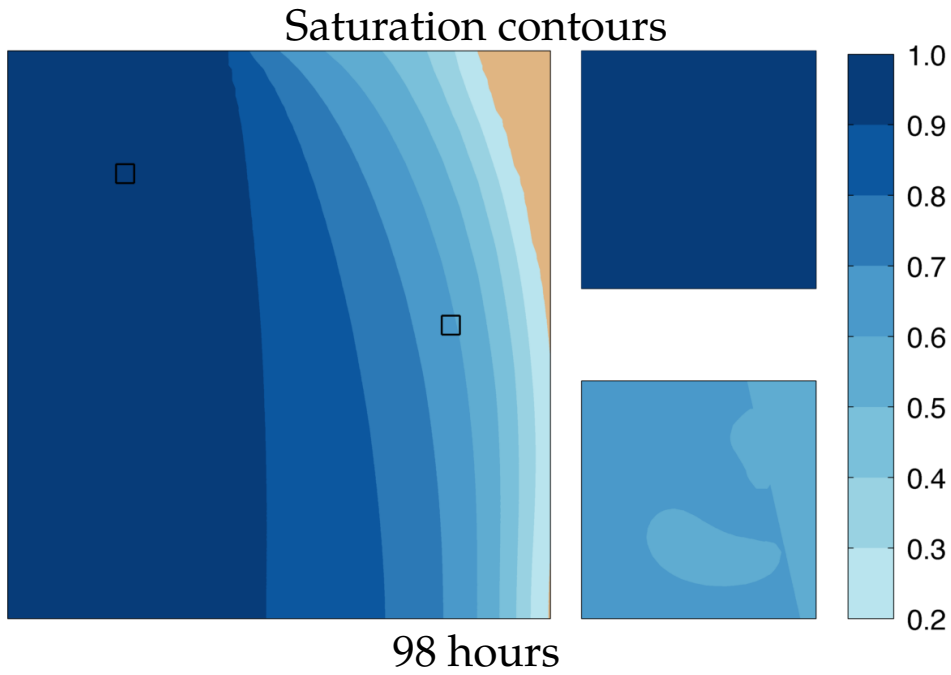
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



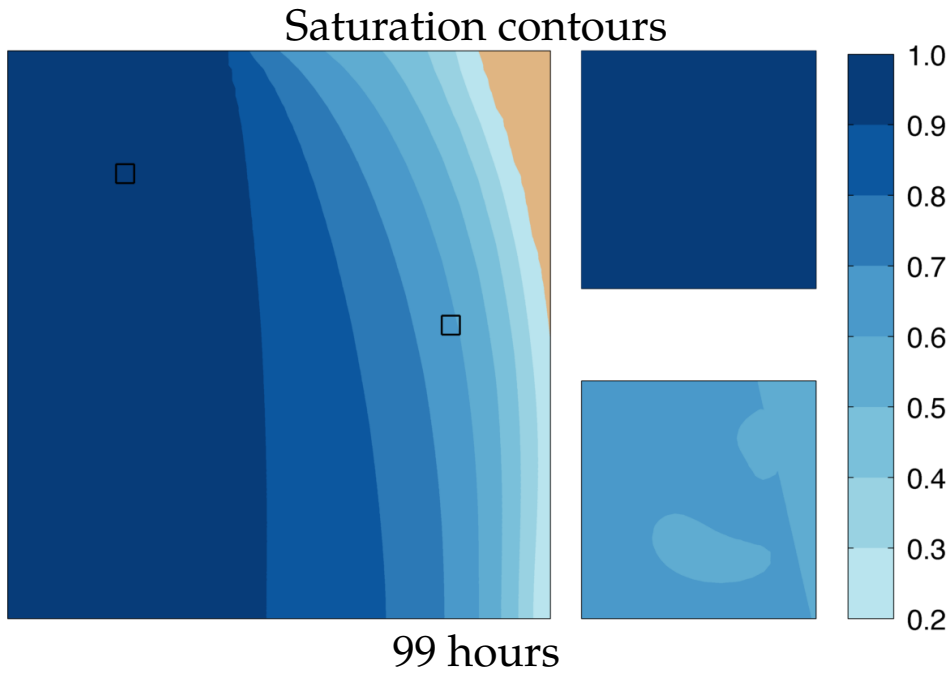
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



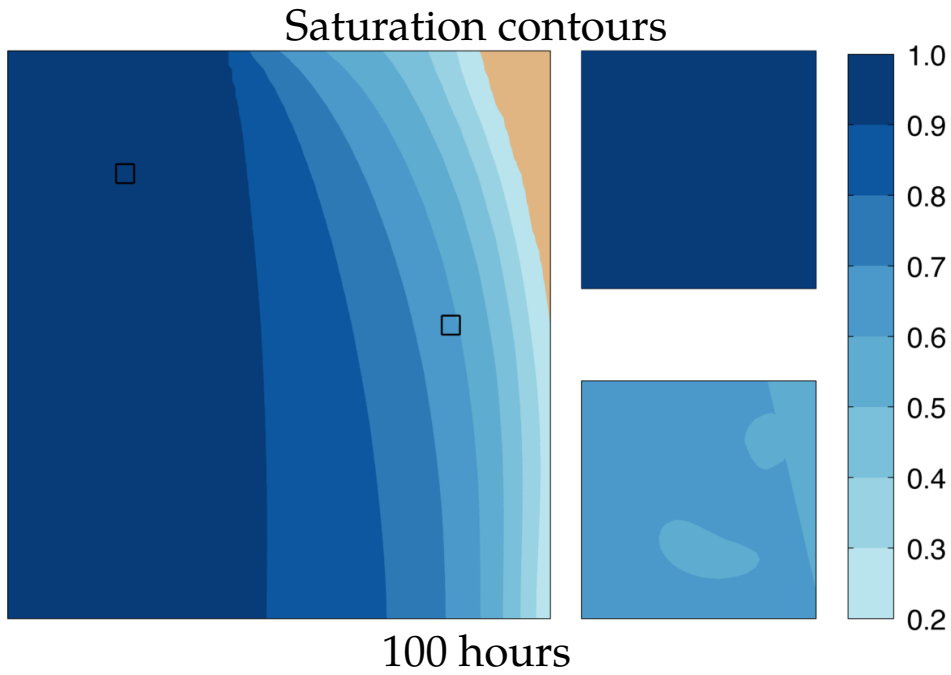
	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Simulation: Unsaturated water flow



	Serial	1 core	2 cores	4 cores	8 cores
Runtime (min)	179	182	94	51	31
Speedup	1.00	0.98	1.90	3.50	5.84

Summary and Conclusions

Pros and Cons of two-scale model:

- ✓ Ability to capture fine-scale detail in the full heterogeneous problem
- ✓ Produces good qualitative agreement with fine-scale model
- ✓ Significant reduction in simulation time (compared with fine-scale model)
- ✓ Numerically feasible for problems with very small-scale heterogeneities
- ✗ Restricted to heterogeneous domains comprised of two sub-domains where one is connected and the other forms disconnected/isolated inclusions

Summary and Conclusions

Pros and Cons of two-scale model:

- ✓ Ability to capture fine-scale detail in the full heterogeneous problem
- ✓ Produces good qualitative agreement with fine-scale model
- ✓ Significant reduction in simulation time (compared with fine-scale model)
- ✓ Numerically feasible for problems with very small-scale heterogeneities
- ✗ Restricted to heterogeneous domains comprised of two sub-domains where one is connected and the other forms disconnected/isolated inclusions

Questions?

References

E. J. Carr and I. W. Turner. Two-scale computational modelling of water flow in unsaturated soils containing irregular-shaped inclusions. *International Journal for Numerical Methods in Engineering*, 98(3): 157–173, 2014.

E. J. Carr, T. J. Moroney, and I. W. Turner. Efficient simulation of unsaturated flow using exponential time integration. *Appl. Math. Comput.*, 217(14):6587–6596, 2011.

E. J. Carr, T. J. Moroney, and I. W. Turner. Performance assessment of exponential rosenbrock methods for large systems ode. *ANZIAM Journal*, 54(CTAC2012):C102–C118, 2013.

C. Guezaine and J-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Schumaker, and C. S. Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software*, 31(3):363–396, 2005.

M. Hochbruck, C. Lubich, and H. Selhofer. Exponential integrators for large systems of differential equations. *SIAM. J. Sci. Comput.*, 19(5):1552–1574, 1998.

M. Hochbruck, A. Ostermann, and J. Schweitzer. Exponential Rosenbrock–type methods. *SIAM Journal on Numerical Analysis*, 47(1):786–803, 2009.

M. R. Kirkland, R. G. Hills, and P. J. Wierenga. Algorithms for solving Richards' equation for variably saturated soils. *Water Resources Research*, 28(8):2049–2058, 1992.

D. McBride, M. Cross, N. Croft, C. Bennett, and J. Gebhardt. Computational modelling of variably–saturated flow in porous media with complex three-dimensional geometries. *International Journal for Numerical Methods in Fluids*, 50(9):1085–1117, 2006.

B. V. Minchev and W. M. Wright. A review of exponential integrators for first order semi–linear problems. Technical Report Numerics No. 2, Norwegian University of Science and Technology, Trondheim, Norway, 2005.

A. Szymkiewicz and J. Lewandowska. Unified macroscopic model for unsaturated water flow in soils of bimodal porosity. *Hydrological Sciences Journal*, 51(6):1106–1124, 2006.

A. Szymkiewicz and J. Lewandowska. Micromechanical approach to unsaturated water flow in structured geomaterials by two-scale computations. *Acta Geotechnica*, 3:37–47, 2008.

M. Tokman. Efficient integration of large stiff systems of ODEs with exponential propagation iteration (EPI) methods. *J. Comput. Phys.*, 213:748–776, 2006.

M. T. van Genuchten. A closed–form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44(5):892–898, 1980.