Fully implicit higher-order schemes applied to polymer flooding

K.-A. Lie(SINTEF), T.S. Mykkeltvedt(IRIS), X. Raynaud (SINTEF)

Motivation

- Challenging and complex to simulate water-based EOR methods, unresolved simulation can lead to misleading predictions
- Polymer flooding enhances the water's ability to push the oil through the reservoir, and can reduce channeling through high flow zones
- Crucial to capture polymer fronts sharply to resolve the nonlinear displacement mechanisms correctly
- Polymer fronts will in the worst case be linear waves and generally have significantly less self-sharpening effects than water fronts

  challenge when using standard low-order methods

General flow equations for two-phase flow

Gathering the equations, we have

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla\cdot\left(\rho_\alpha\vec{u}_\alpha\right) = q_\alpha, \qquad \alpha = \{w, n\}$$

$$\vec{u}_\alpha = -\frac{\mathbf{K}k_{r\alpha}}{\mu_\alpha}\left(\nabla p_\alpha - \rho_\alpha g\nabla z\right)$$

$$p_c = p_n - p_w, \qquad S_w + S_n = 1$$

Model for polymer flooding

Simple model: introduce extra immiscible component and mixture law

Polymer transported in water:

$$\vec{u}_w = -\frac{k_{rw}(S)}{\mu_{w,\mathsf{eff}}(c)\, R_k(c)}\mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

$$\vec{u}_p = -\frac{k_{rw}(S)}{\mu_{p,\mathsf{eff}}(c) R_k(c)}\mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

Conservation of polymer component:

$$\partial_t\left[\phi(1-S_{ipv})cb_w S + \rho_r\, C^a(1-\phi)\right] + \nabla\cdot\left(cb_w\vec{u}_p\right) = q_p$$

Model for polymer flooding

Simple model: introduce extra immiscible component and mixture law

Polymer transported in water:

$$\vec{u}_w = -\frac{k_{rw}(S)}{\mu_{w,\text{eff}}(c)\, R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

viscosity enhancement

$$\vec{u}_p = -\frac{k_{rw}(S)}{\mu_{p,\text{eff}}(c) R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

Todd–Longstaff mixing:

$$\frac{1}{\mu_{w,\text{eff}}} = \frac{1 - c/c_m}{\mu_m(c)^\omega \mu_w^{1-\omega}} + \frac{c/c_m}{\mu_m(c)^\omega \mu_p^{1-\omega}}$$

Conservation of polymer component:

$$\partial_t \left[\phi(1 - S_{ipv}) c b_w S + \rho_r\, C^a (1 - \phi)\right] + \nabla \cdot \left(c b_w \vec{u}_p\right) = q_p$$

Model for polymer flooding

Simple model: introduce extra immiscible component and mixture law

Polymer transported in water:

$$\vec{u}_w = -\frac{k_{rw}(S)}{\mu_{w,\text{eff}}(c)\, R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

viscosity enhancement

permeability reduction

$$\vec{u}_p = -\frac{k_{rw}(S)}{\mu_{p,\text{eff}}(c) R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

Todd–Longstaff mixing:

$$\frac{1}{\mu_{w,\text{eff}}} = \frac{1-c/c_m}{\mu_m(c)^\omega \mu_w^{1-\omega}} + \frac{c/c_m}{\mu_m(c)^\omega \mu_p^{1-\omega}}$$

Conservation of polymer component:

$$\partial_t \left[\phi(1-S_{ipv})cb_w S + \rho_r\, C^a(1-\phi)\right] + \nabla \cdot \left(cb_w \vec{u}_p\right) = q_p$$

Model for polymer flooding

Simple model: introduce extra immiscible component and mixture law

Polymer transported in water:

$$\vec{u}_w = -\frac{k_{rw}(S)}{\mu_{w,\text{eff}}(c)\, R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

viscosity enhancement

permeability reduction

Todd–Longstaff mixing:

$$\frac{1}{\mu_{w,\text{eff}}} = \frac{1 - c/c_m}{\mu_m(c)^\omega \mu_w^{1-\omega}} + \frac{c/c_m}{\mu_m(c)^\omega \mu_p^{1-\omega}}$$

$$\vec{u}_p = -\frac{k_{rw}(S)}{\mu_{p,\text{eff}}(c) R_k(c)} \mathbf{K}\big(\nabla p_w - \rho_w g \nabla z\big)$$

Conservation of polymer component:

$$\partial_t \left[\phi(1 - S_{ipv}) c b_w S + \rho_r\, C^a (1 - \phi)\right] + \nabla \cdot \left(c b_w \vec{u}_p\right) = q_p$$

inaccessible pore space

Model for polymer flooding

Simple model: introduce extra immiscible component and mixture law

Polymer transported in water:

$$\vec{u}_w = -\frac{k_{rw}(S)}{\mu_{w,\text{eff}}(c)\, R_k(c)} \mathbf{K}\left(\nabla p_w - \rho_w g \nabla z\right)$$

viscosity enhancement

permeability reduction

Todd–Longstaff mixing:

$$\frac{1}{\mu_{w,\text{eff}}} = \frac{1 - c/c_m}{\mu_m(c)^\omega \mu_w^{1-\omega}} + \frac{c/c_m}{\mu_m(c)^\omega \mu_p^{1-\omega}}$$

$$\vec{u}_p = -\frac{k_{rw}(S)}{\mu_{p,\text{eff}}(c) R_k(c)} \mathbf{K}\left(\nabla p_w - \rho_w g \nabla z\right)$$

Conservation of polymer component:

$$\partial_t \left[\phi(1 - S_{ipv}) c b_w S + \rho_r\, C^a (1 - \phi)\right] + \nabla \cdot \left(c b_w \vec{u}_p\right) = q_p$$

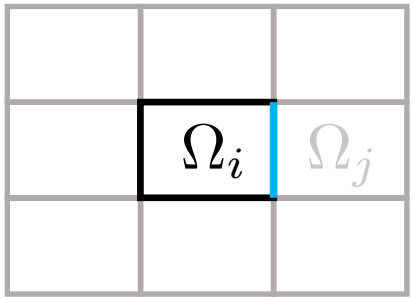
inaccessible pore space

adsorption

Numerical framework

Dicrete cell-averages ($q = p, s, c$)

$$q_i(t) = \frac{1}{|\Omega_i|} \iint_{\Omega_i} q(x, y, t) \mathrm{d}\mathbf{x}$$



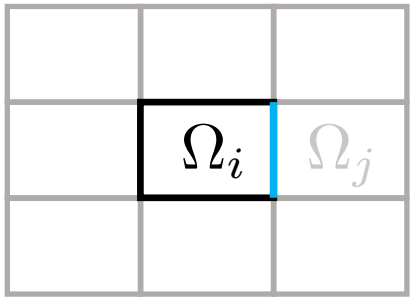
Discretized eq. for water:

$$[\rho_w(p_i)\phi(p_i)s_i]^{n+1} = [\rho_w(p_i)\phi(p_i)s_i]^n - \frac{\Delta t}{|\Omega_i|} \sum_{|i-j=1|} \int_{\Gamma_{ij}} [\rho_w(p)\lambda_w(s,c)]_{ij}^m \left(\mathbf{v}_w^m \cdot \mathbf{n}\right)_{ij} \mathrm{d}s$$

Numerical framework

Dicrete cell-averages ($q = p, s, c$)

$$q_i(t) = \frac{1}{|\Omega_i|} \iint_{\Omega_i} q(x, y, t) \mathrm{d}\mathbf{x}$$



Discretized eq. for water:

$$[\rho_w(p_i)\phi(p_i)s_i]^{n+1} = [\rho_w(p_i)\phi(p_i)s_i]^n - \frac{\Delta t}{|\Omega_i|} \sum_{|i-j=1|} \int_{\Gamma_{ij}} [\rho_w(p)\lambda_w(s,c)]_{ij}^m \left(\mathbf{v}_w^m \cdot \mathbf{n}\right)_{ij} \mathrm{d}s$$

How to compute the mass flux evaluated <u>at the interface?</u>

$$\int_{\Gamma_{ij}} \rho_{ij}(p)\lambda_{ij}(s,c) \left(\mathbf{v} \cdot \mathbf{n}\right)_{ij} \mathrm{d}s \approx \frac{1}{2} \left[\rho(p_i) + \rho(p_j)\right] v_{ij} \int_{\Gamma_{ij}} \lambda(s,c) \mathrm{d}s$$

- Which quadrature rule to use for the integral ?

- How to reconstruct the necessary one-sided point values from the cell-averages ?

- How to approximate the mobility given point values that generally are different on each side of Γ_{ij} ?

- Which quadrature rule to use for the integral ?
  - Midpoint, Simpsons rule, 4th order Gauss

- How to reconstruct the necessary one-sided point values from the cell-averages ?

- How to approximate the mobility given point values that generally are different on each side of Γ_{ij} ?

$$\lambda_{ij}(s,c) = \begin{cases} \lambda(s_{ij}^-, c_{ij}^-), & \text{if } v_{ij} \geq 0, \\ \lambda(s_{ij}^+, c_{ij}^+), & \text{otherwise.} \end{cases}$$

- **First-order scheme:** constant reconstruction

  $\tilde{q}(\mathbf{x}) = q_i$

- **Second-order slope-limiter scheme:** linear reconstruction

  $\tilde{q}(\mathbf{x}) = q_i + \sigma_i^x (x - x_i) + \sigma_i^y (y - y_i)$

  with slopes estimated from cell-averages

  $\Delta x \sigma_i^x = \Phi \left(q_i - q_{i-(1,0)}, q_{i+(1,0)} - q_i \right)$

- **First-order scheme:** constant reconstruction

  $\tilde{q}(\mathbf{x}) = q_i$

- **Second-order slope-limiter scheme:** linear reconstruction

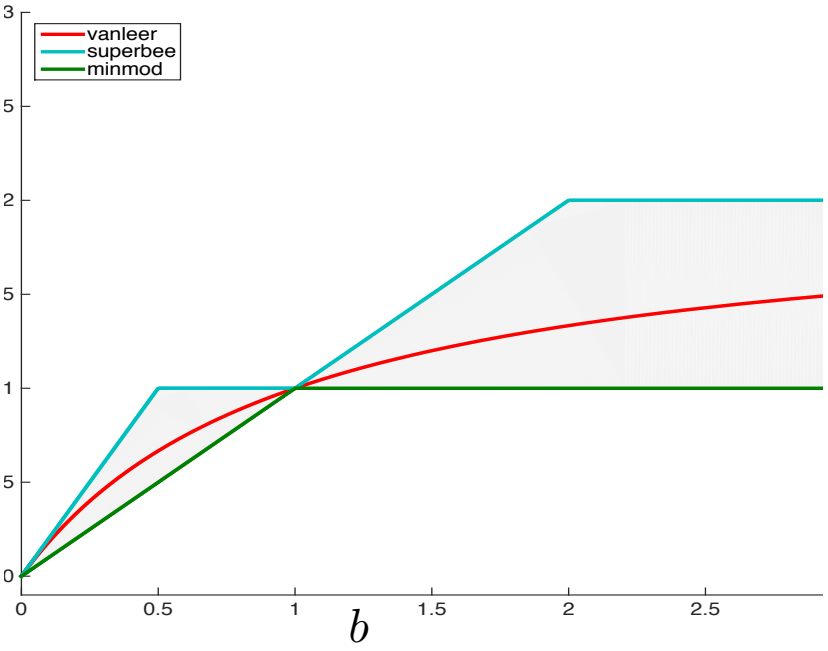
  $\tilde{q}(\mathbf{x}) = q_i + \sigma_i^x (x - x_i) + \sigma_i^y (y - y_i)$

  with slopes estimated from cell-averages

  $\Delta x \sigma_i^x = \Phi \left(q_i - q_{i-(1,0)}, q_{i+(1,0)} - q_i \right)$

$\Phi(1, b)$

$\theta = 2$



WENO schemes

WENO builds on ENO, which picks the least oscillatory stencil, uses a convex combination of local stencils

Here: simplified version, with four linear reconstructions

Polynomial:

$$q^{NE}(\mathbf{x}) = q_i + \sigma_i^E (x - x_i) + \sigma_i^N (y - y_i)$$

Smoothness indicator:

$$\beta_i^{NE} = \frac{1}{4}\left[\left(\Delta x \sigma_i^E\right)^2 + \left(\Delta y \sigma_i^N\right)^2 + \epsilon\right]^{-l}$$

Weights:

$$w^{NE} = \beta^{NE} / \left(\beta^{NE} + \beta^{NW} + \beta^{SE} + \beta^{SW}\right)$$

Linear combination:

$$q_i(\mathbf{x}) = \sum_{v=NE,NW,SE,SW} w^v q^v(\mathbf{x})$$

$$\sigma_i^E = \frac{q_{i+(1,0)} - q_i}{\Delta x}$$

$$\sigma_i^N = \frac{q_{i+(0,1)} - q_i}{\Delta y}$$

Solving the discretized systems

System on residual form: $\mathbf{F}(\mathbf{y}) = \mathbf{0}$ $\mathbf{y} = [\mathbf{p}, \mathbf{s}, \mathbf{c}]$

Use Newton-Raphson method: $\mathbf{y} = \mathbf{y_0} + \delta\mathbf{y}$

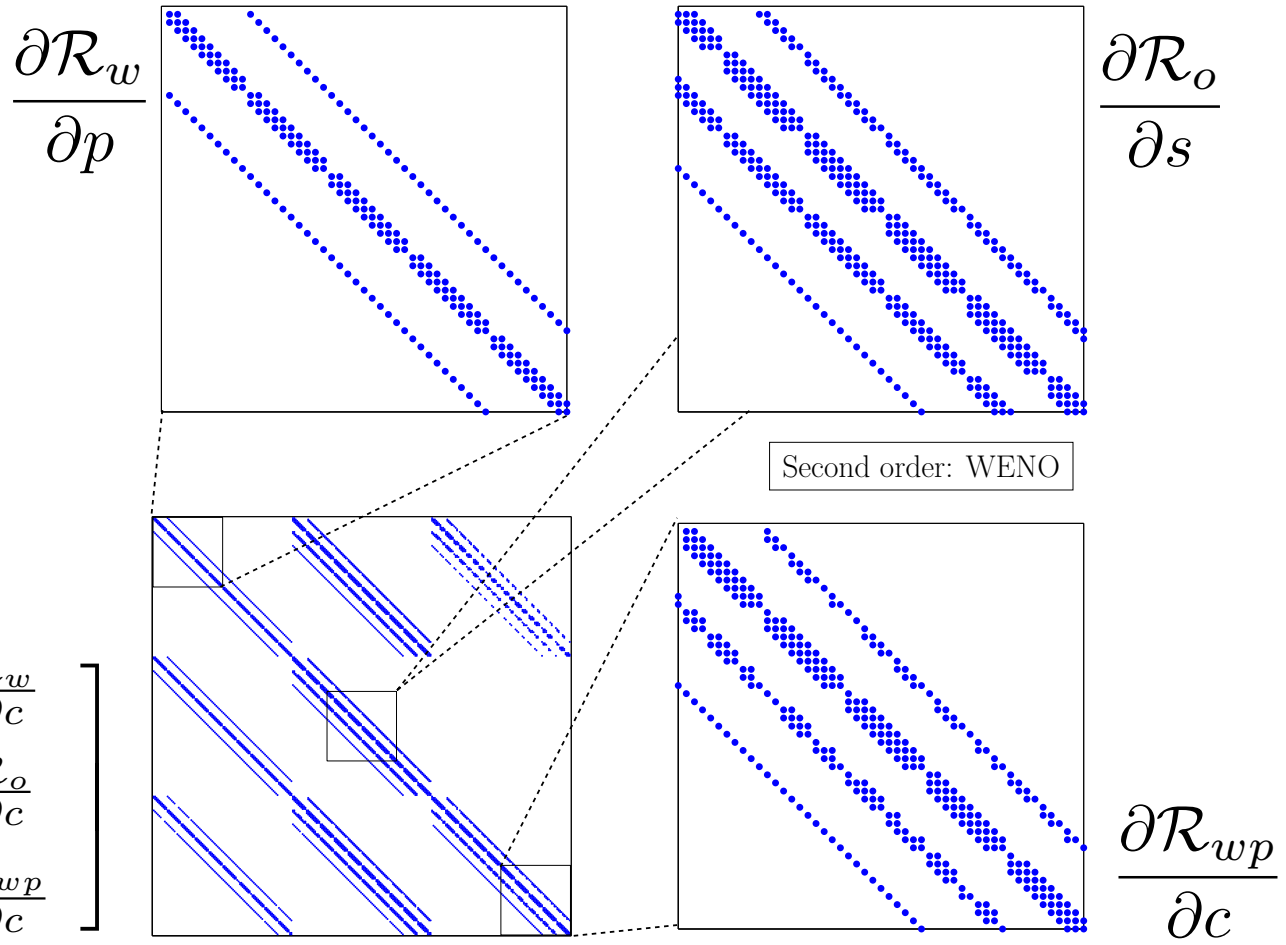
$$\mathbf{0} \approx \mathbf{F}(\mathbf{y_0}) + \mathbf{J}\delta\mathbf{y}, \qquad \mathbf{J} = \frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{y}}$$

Structure of the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\mathcal{R}_w}{\partial p} & \frac{\mathcal{R}_w}{\partial s} & \frac{\mathcal{R}_w}{\partial c} \\ \frac{\mathcal{R}_o}{\partial p} & \frac{\mathcal{R}_o}{\partial s} & \frac{\mathcal{R}_o}{\partial c} \\ \frac{\mathcal{R}_{wp}}{\partial p} & \frac{\mathcal{R}_{wp}}{\partial s} & \frac{\mathcal{R}_{wp}}{\partial c} \end{bmatrix}$$

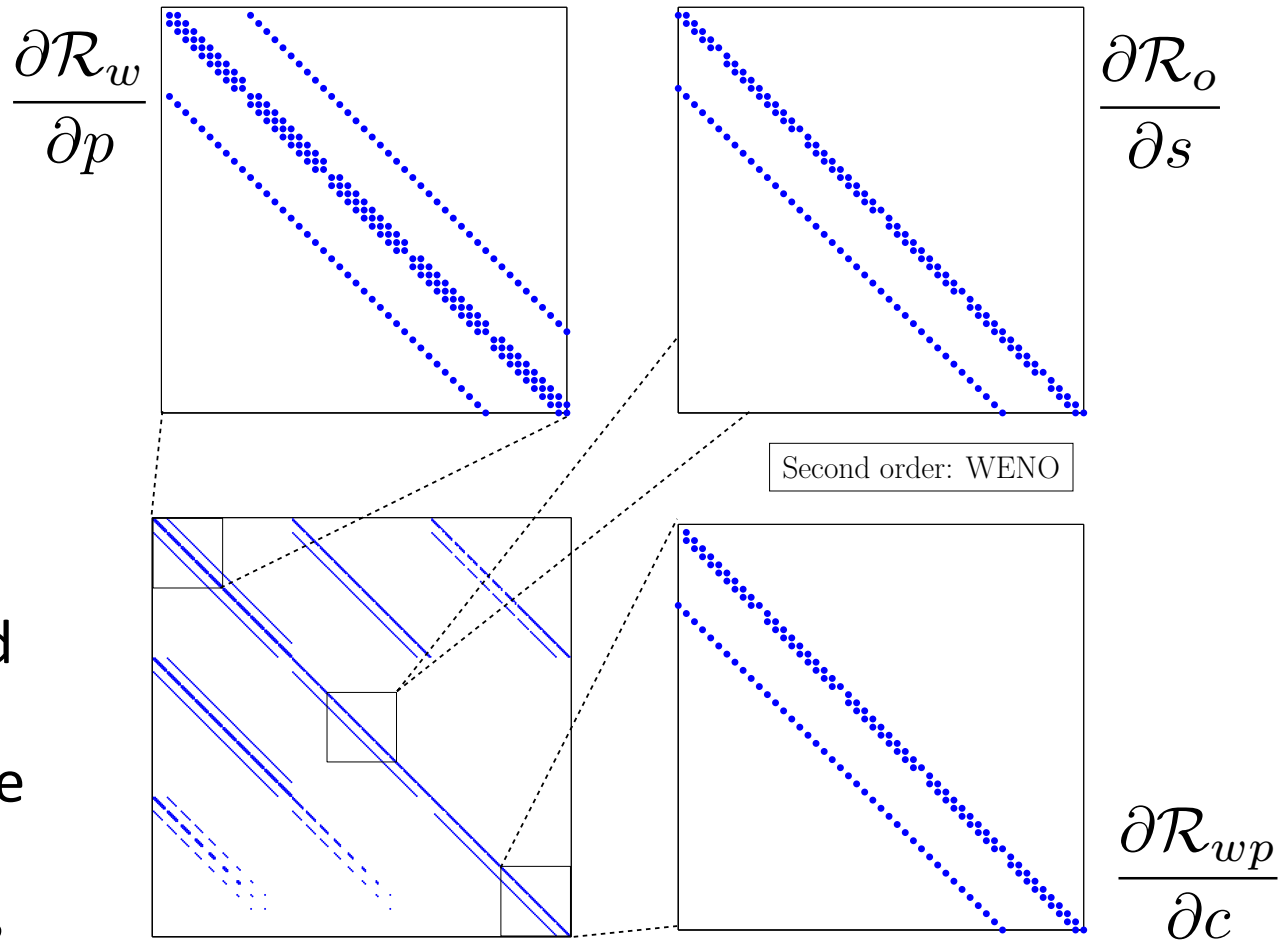
Structure of the Jacobian matrix for the implicit WENO scheme

$$\mathbf{J} = \begin{bmatrix} \frac{\mathcal{R}_w}{\partial p} & \frac{\mathcal{R}_w}{\partial s} & \frac{\mathcal{R}_w}{\partial c} \\ \frac{\mathcal{R}_o}{\partial p} & \frac{\mathcal{R}_o}{\partial s} & \frac{\mathcal{R}_o}{\partial c} \\ \frac{\mathcal{R}_{wp}}{\partial p} & \frac{\mathcal{R}_{wp}}{\partial s} & \frac{\mathcal{R}_{wp}}{\partial c} \end{bmatrix}$$



Structure of the Jacobian matrix for the implicit WENO scheme

When using lagged evaluation of the WENO weights, the structure of the Jacobian simplifies

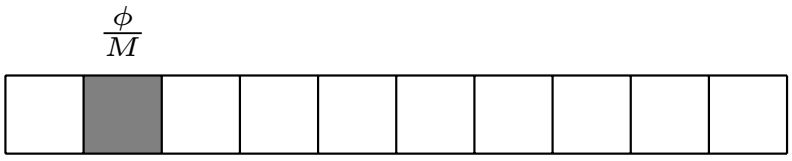


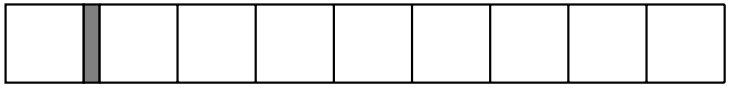
A motivating example

Consider $\phi q_t + q_x = 0$

A motivating example

Consider $\phi q_t + q_x = 0$ $\xrightarrow{\tau = x\phi}$ transformed $q_t + q_\tau = 0$





A motivating example

Consider $\phi q_t + q_x = 0$ $\xrightarrow{\tau = x\phi}$ transformed $q_t + q_\tau = 0$

Modified equation (implicit/explicit first order)

$$q_t + q_\tau = \tfrac{1}{2}(\Delta\tau \pm \Delta t) q_{\tau\tau}$$

Smearing of discontinuity $\mathcal{O}\big(\sqrt{t(\Delta\tau \pm \Delta t)}\big)$

A motivating example

Consider $\phi q_t + q_x = 0$ $\xrightarrow{\tau = x\phi}$ transformed $q_t + q_\tau = 0$

Modified equation (implicit/explicit first order)

$$q_t + q_\tau = \tfrac{1}{2}(\Delta\tau \pm \Delta t) q_{\tau\tau}$$

Smearing of discontinuity $\mathcal{O}\big(\sqrt{t(\Delta\tau \pm \Delta t)}\big)$

Hence:

$$\underbrace{\frac{9\phi}{10}\big(\Delta x\phi \pm \Delta t\big)}_{\text{high-porosity region}} + \underbrace{\frac{\phi}{10M}\Big(\frac{\Delta x\phi}{M} \pm \Delta t\Big)}_{\text{low-porosity region}}$$

A motivating example

Consider $\phi q_t + q_x = 0$ $\xrightarrow{\tau = x\phi}$ transformed $q_t + q_\tau = 0$

Modified equation (implicit/explicit first order)

$$q_t + q_\tau = \tfrac{1}{2}(\Delta\tau \pm \Delta t) q_{\tau\tau}$$

Smearing of discontinuity $\mathcal{O}\big(\sqrt{t(\Delta\tau \pm \Delta t)}\big)$

Hence:

$$\underbrace{\frac{9\phi}{10}\big(\Delta x\phi \pm \Delta t\big)}_{\text{high-porosity region}} + \underbrace{\frac{\phi}{10M}\Big(\frac{\Delta x\phi}{M} \pm \Delta t\Big)}_{\text{low-porosity region}} = \frac{9\Delta x\phi}{10}\big(1 \pm \frac{\nu}{M}\big) + \frac{\phi^2 \Delta x}{10M^2}\big(1 \pm \nu\big)$$

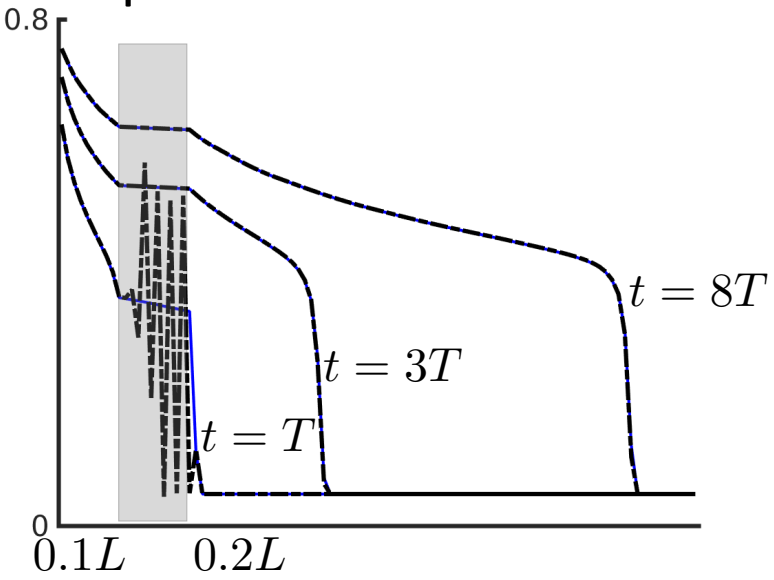
1D: pure waterflooding with regions of different porosity

$$\phi = \begin{cases} 0.01, & \text{if } x \in [0.1L, 0.2L], \\ 0.2, & \text{otherwise.} \end{cases}$$

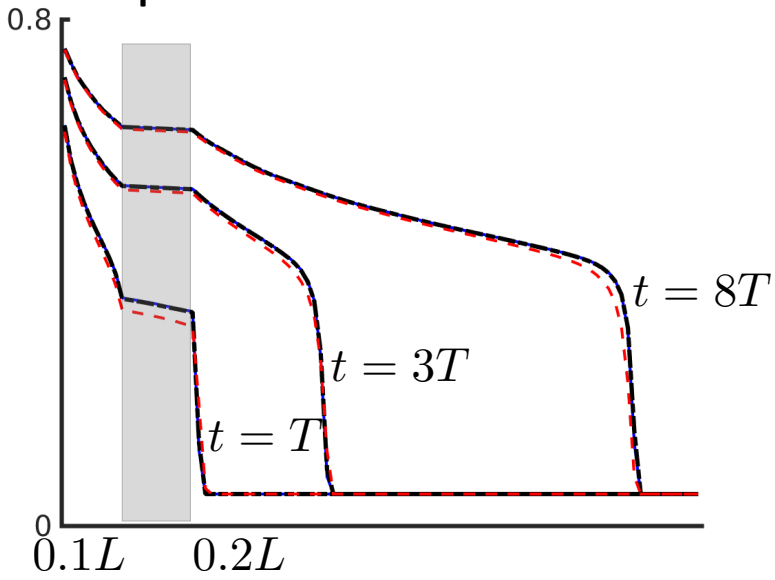
$$\Delta t = \begin{cases} 0.01, \\ 0.1, \\ 1, \end{cases}$$

(exceeds the CFL condition in the low-porosity region)

Explicit first-order scheme



Implicit first-order scheme



- Explicit: oscillations (disappear as time evolves)
- Implicit: stable solutions for both time steps

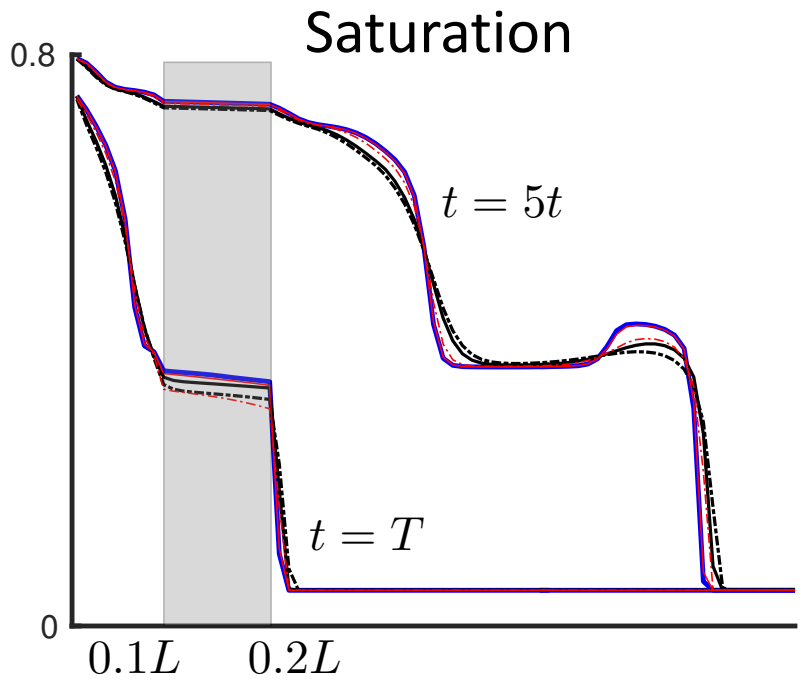
1D: polymer flooding with regions of different porosity

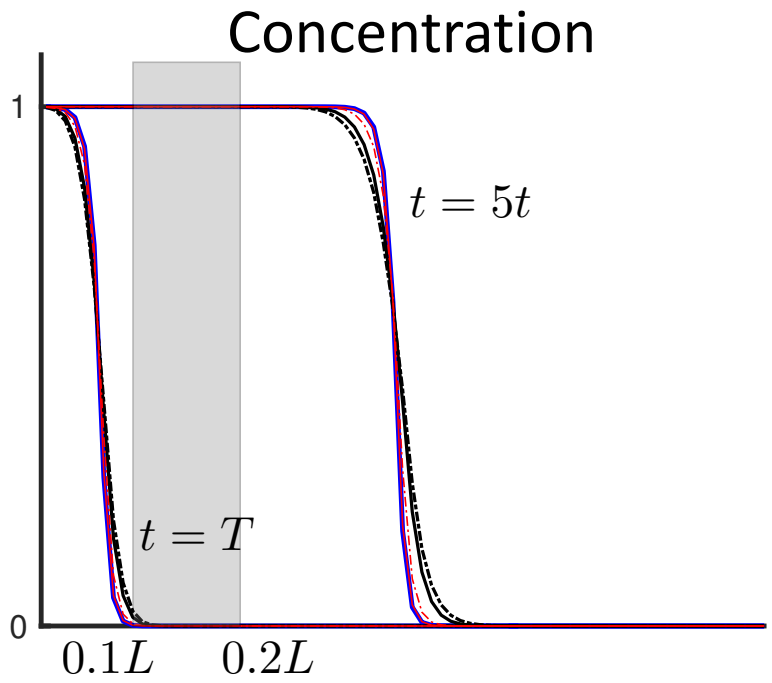
$$\phi = \begin{cases} 0.01, & \text{if } x \in [0.1L, 0.2L], \\ 0.2, & \text{otherwise.} \end{cases}$$

Explicit 1st - order — $\Delta t = 0.01$

Implicit 1st - order —/– –

Implicit 2nd - order —/– – $\Delta t = 0.1/\ 1$





- the high-order implicit scheme resolves the displacement fronts as good as the explicit scheme
- the low-order implicit scheme fails to sharply resolve the structure of the oil bank that arises

1D: polymer flooding with regions of different porosity

Comparison of the computational costs:

Simulation case			Linear	CPU
Δt (day)	method	order	solves	time
0.01	explicit	1st	15307	504.2 s
0.01		2nd	15330	541.0 s
0.03		1st	5297	175.5s
0.03		2nd	5328	184.6s
0.1	implicit	1st	2176	89.7 s
0.1		2nd	2960	223.6 s
1		1st	450	14.8 s
1		2nd	508	33.1 s
2		1st	267	8.5 s
2		2nd	330	20.5 s

1D: polymer flooding with regions of different porosity

Comparison of the computational costs:

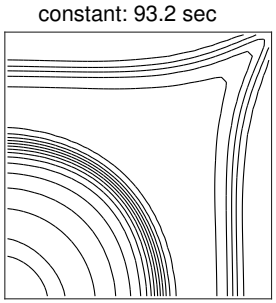
Simulation case			Linear	CPU
Δt (day)	method	order	solves	time
0.01	explicit	1st	15307	504.2 s
0.01		2nd	15330	541.0 s
0.03		1st	5297	175.5s
0.03		2nd	5328	184.6s
0.1	implicit	1st	2176	89.7 s
0.1		2nd	2960	223.6 s
1		1st	450	14.8 s
1		2nd	508	33.1 s
2		1st	267	8.5 s
2		2nd	330	20.5 s

oscillations in the low porosity region

- the second-order implicit scheme is at least as effective as the explicit schemes

Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

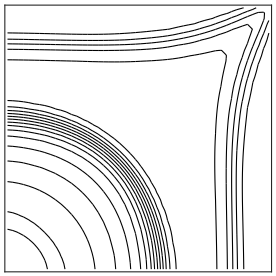
Explicit
(1600 control steps)



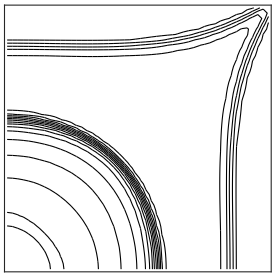
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

Explicit
(1600 control steps)

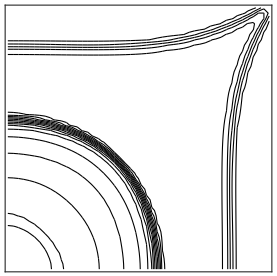
constant: 93.2 sec



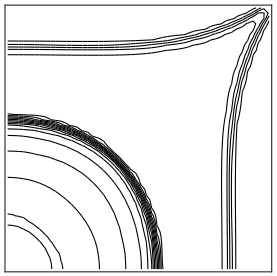
minmod: 98.1 sec



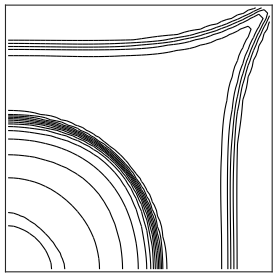
vanleer: 100.8 sec



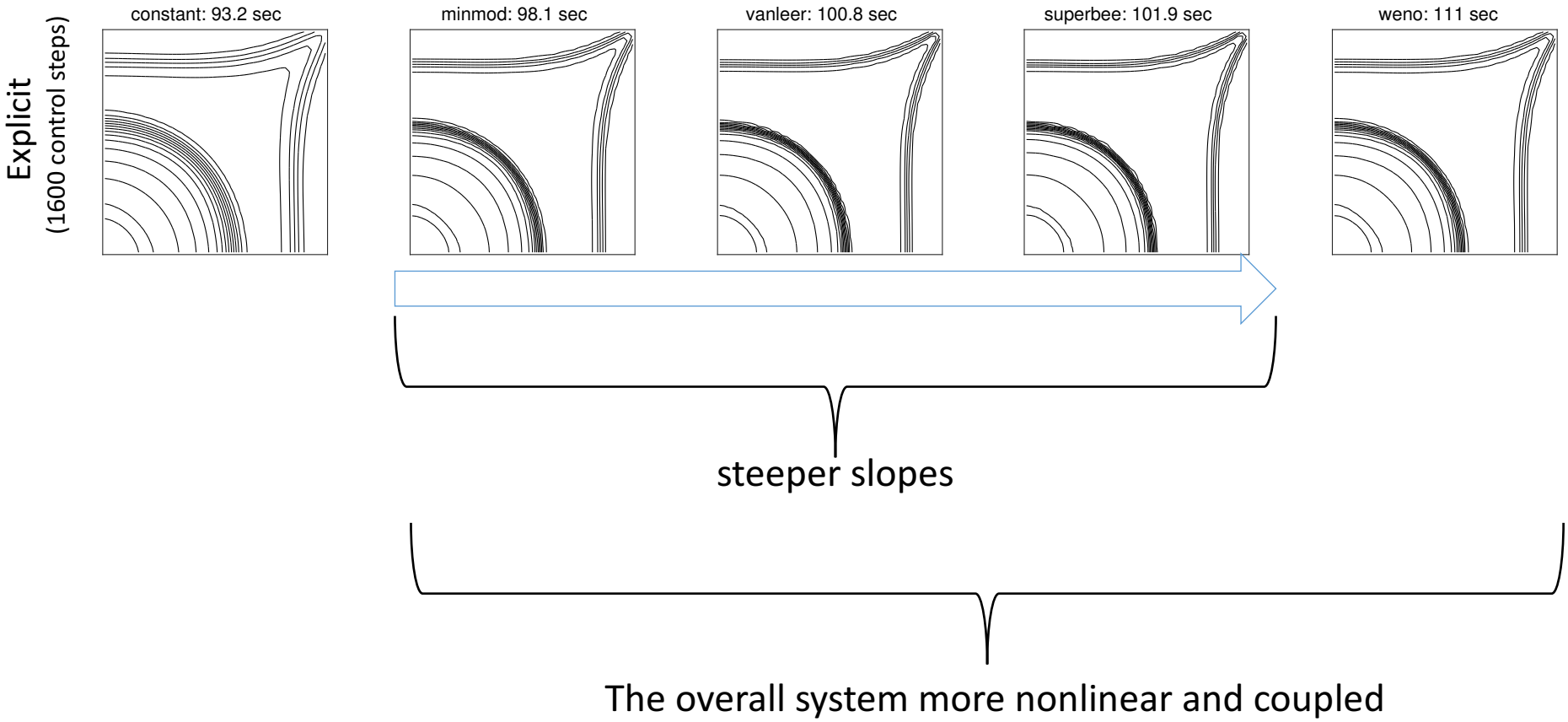
superbee: 101.9 sec



weno: 111 sec

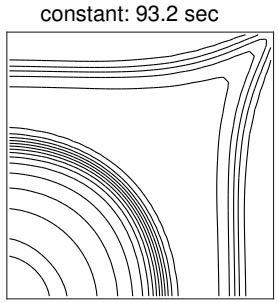


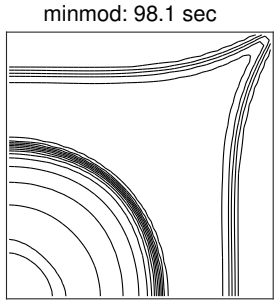
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

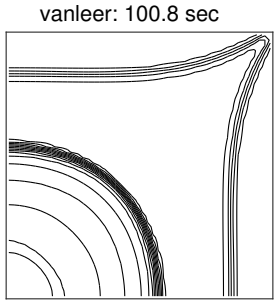


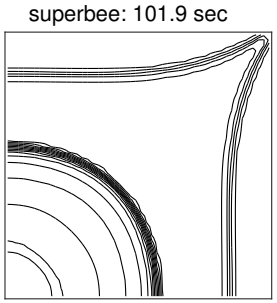
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

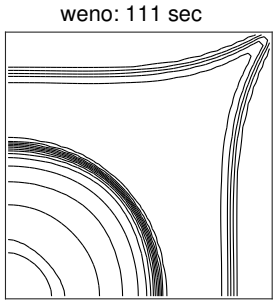
Explicit (1600 control steps)



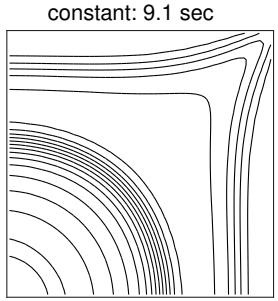


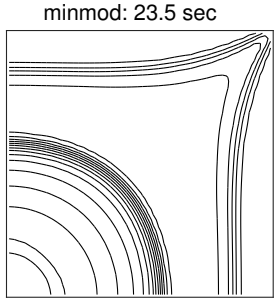


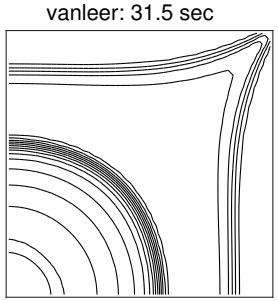


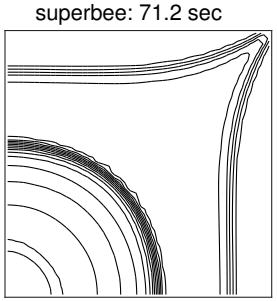


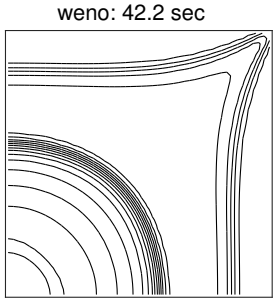
Implicit (50 control steps)



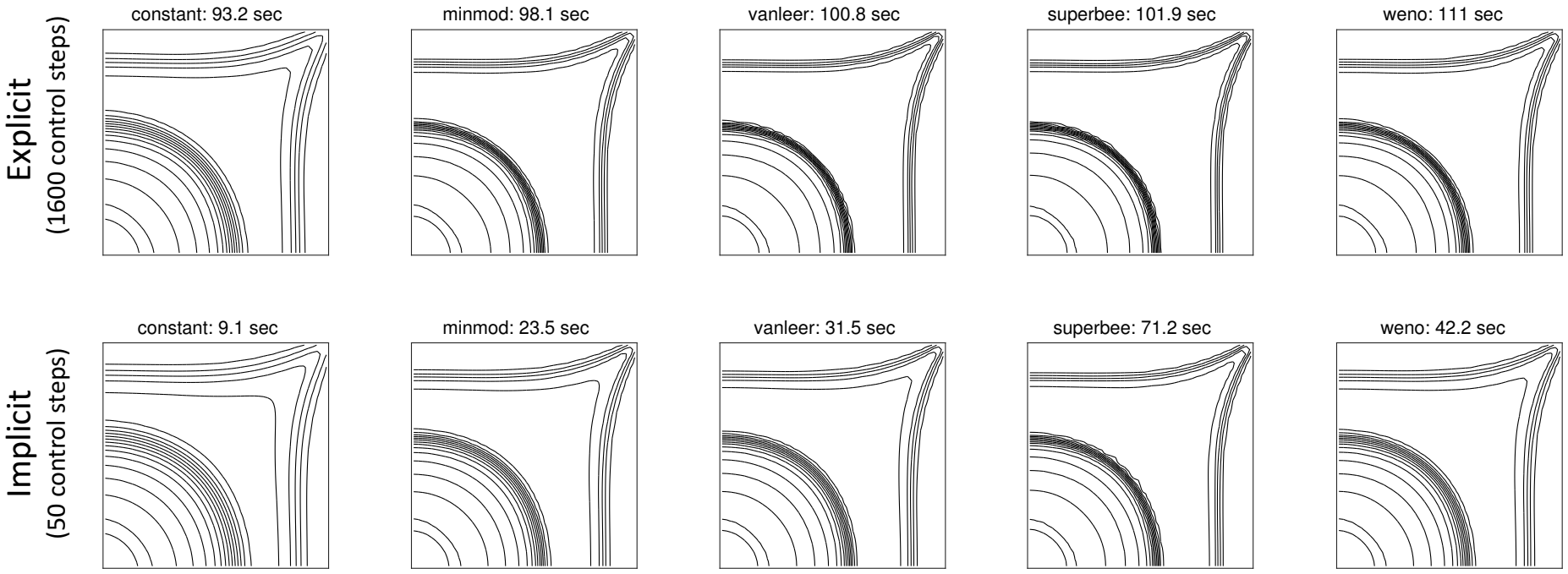








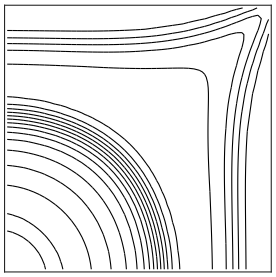
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

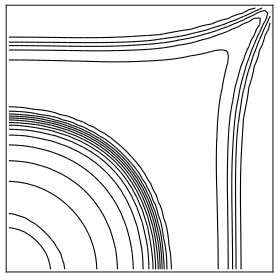


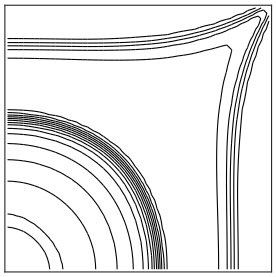
Using a second-order reconstruction and improved spatial quadrature gives more accurate solution profiles both for the explicit and the implicit schemes (as expected)

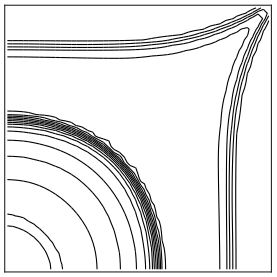
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells

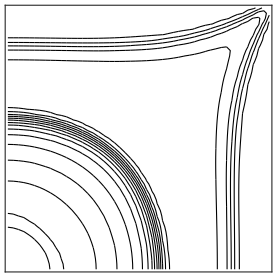
Implicit
(50 control steps)


constant: 9.1 sec

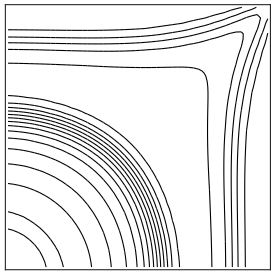

minmod: 23.5 sec

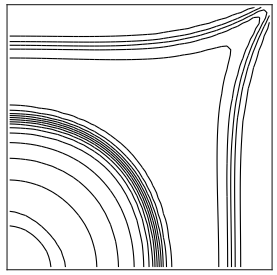

vanleer: 31.5 sec

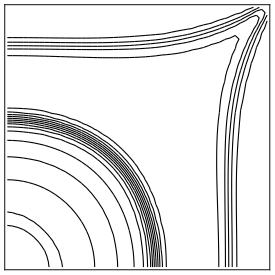

superbee: 71.2 sec

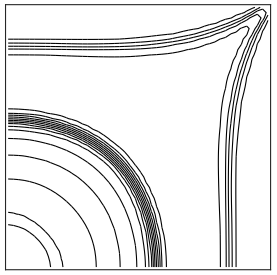

weno: 42.2 sec

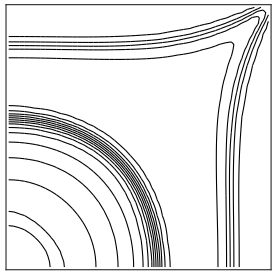
Implicit*
(50 control steps)


constant: 9.1 sec

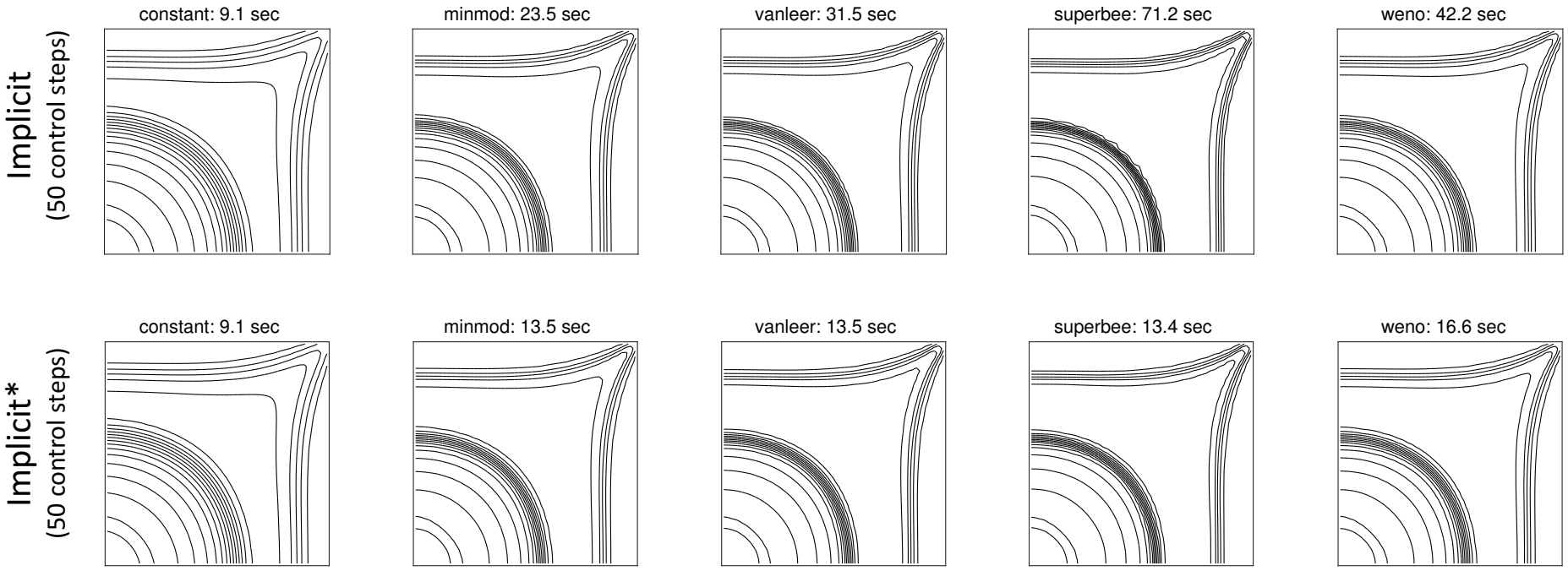

minmod: 13.5 sec


vanleer: 13.5 sec


superbee: 13.4 sec


weno: 16.6 sec

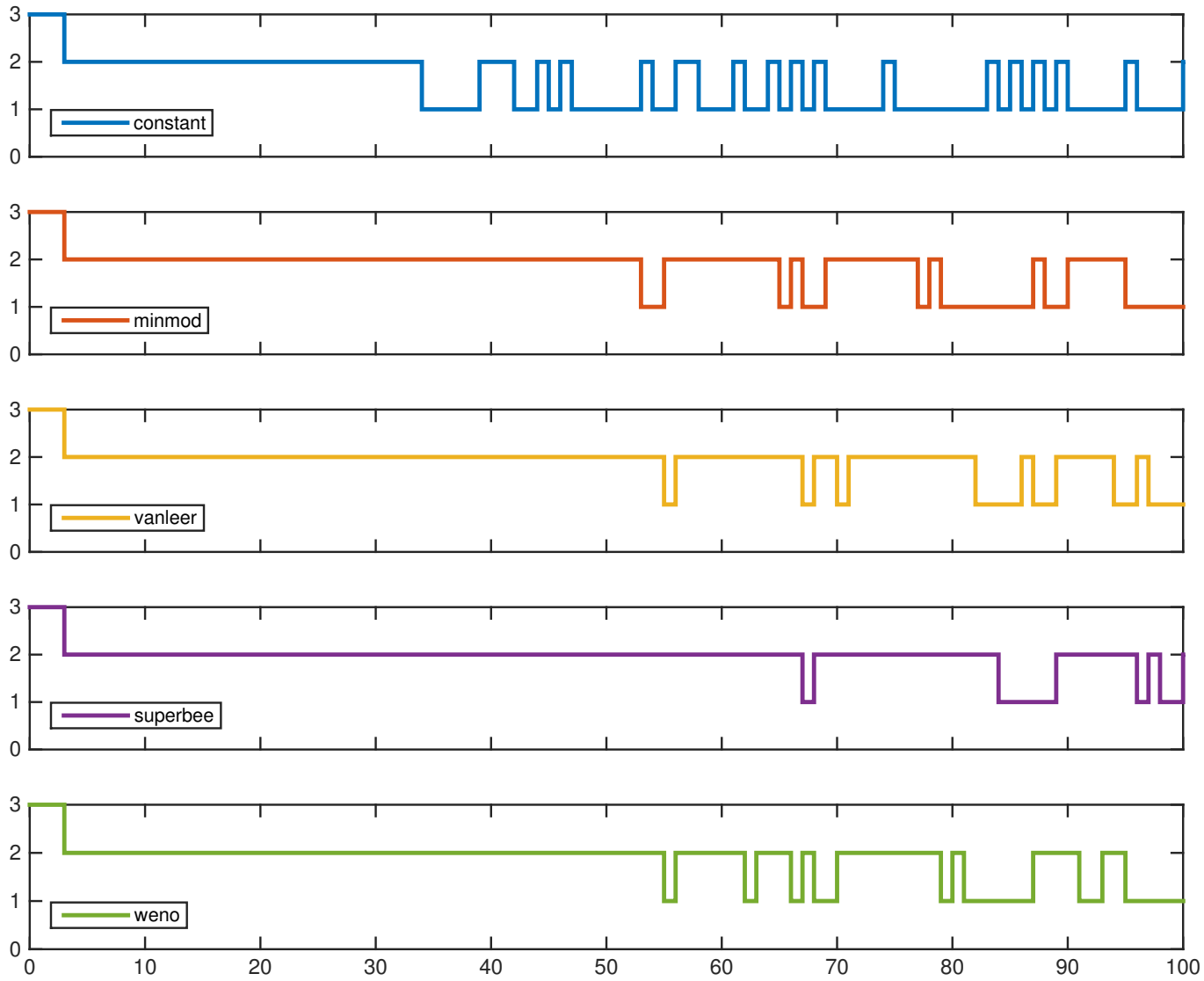
Approximated solution with different reconstructions, on a uniform Cartesian grid with 50x50 cells



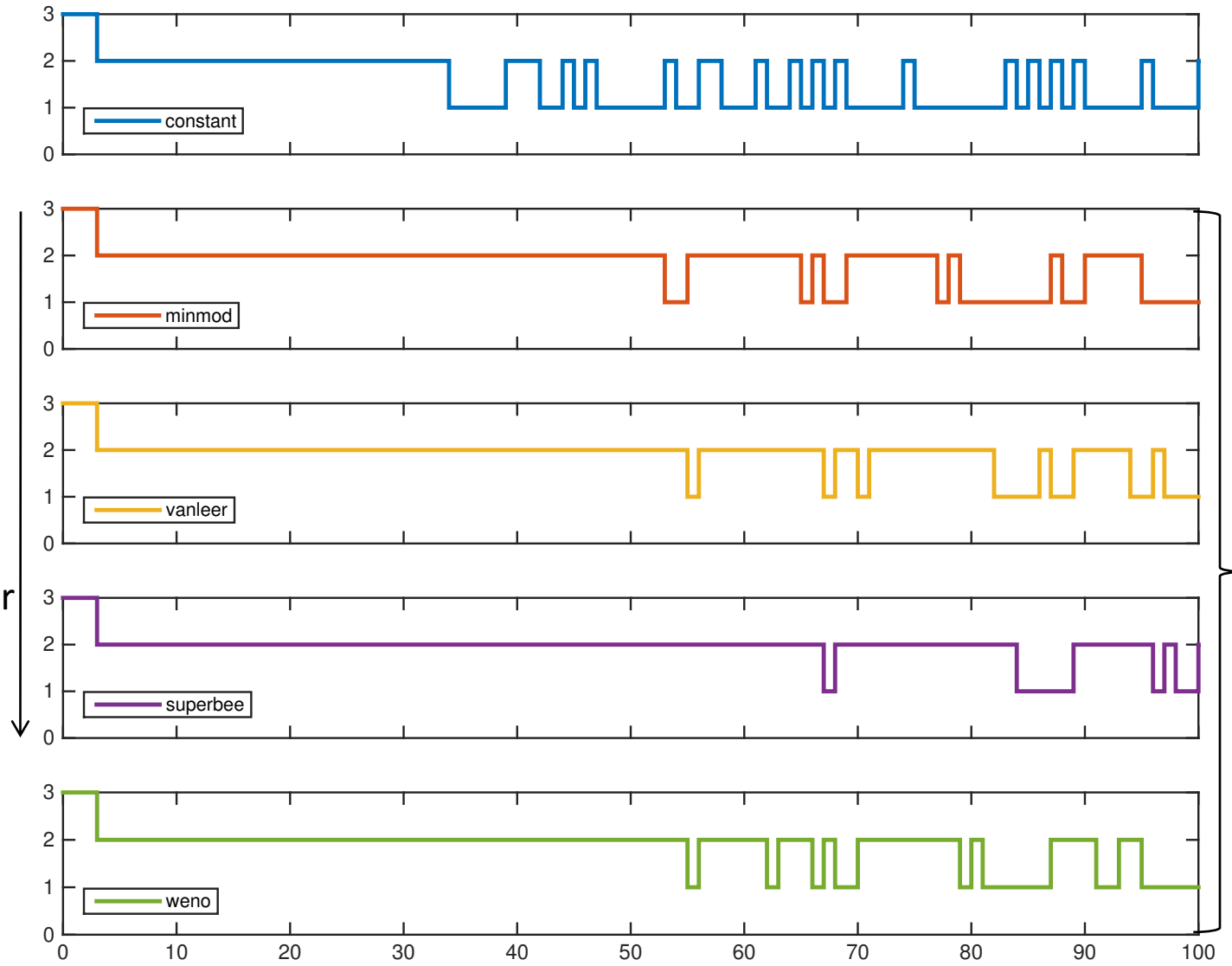
Implicit *: lagged evaluation of slope limiters and WENO weights

- reduces the number of nonlinear iterations
- no adverse effect on the stability and only reduces accuracy slightly

Number of iterations required by the explicit schemes



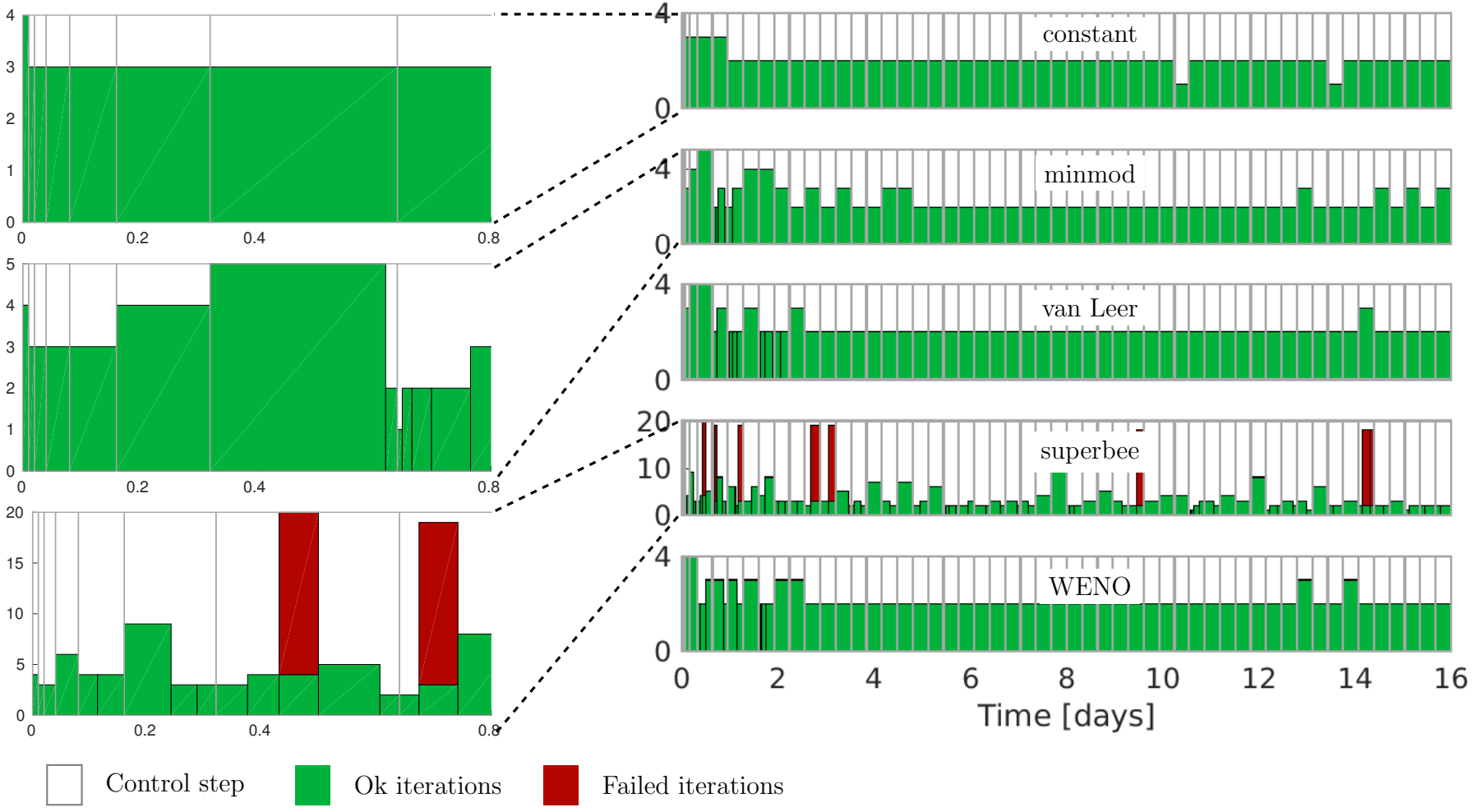
Number of iterations required by the explicit schemes



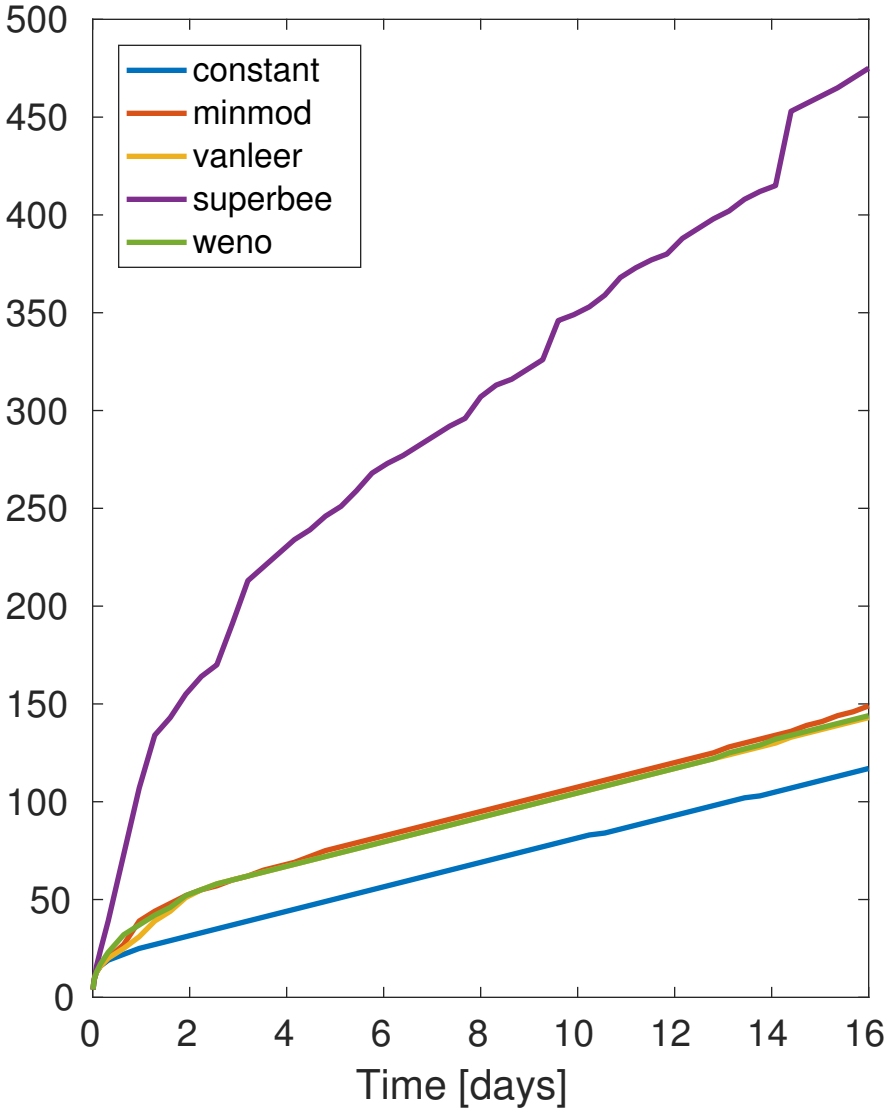
steeper slopes

The overall system more nonlinear and coupled

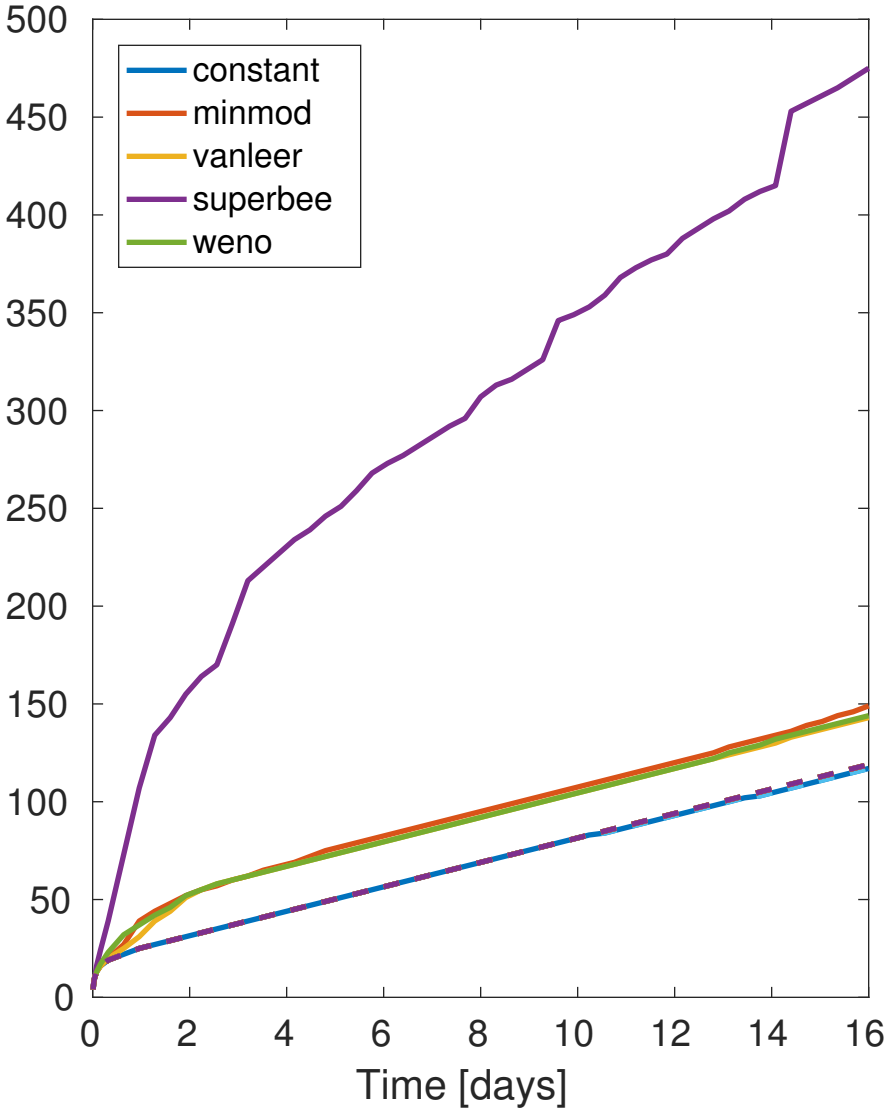
Number of iterations required by the implicit schemes



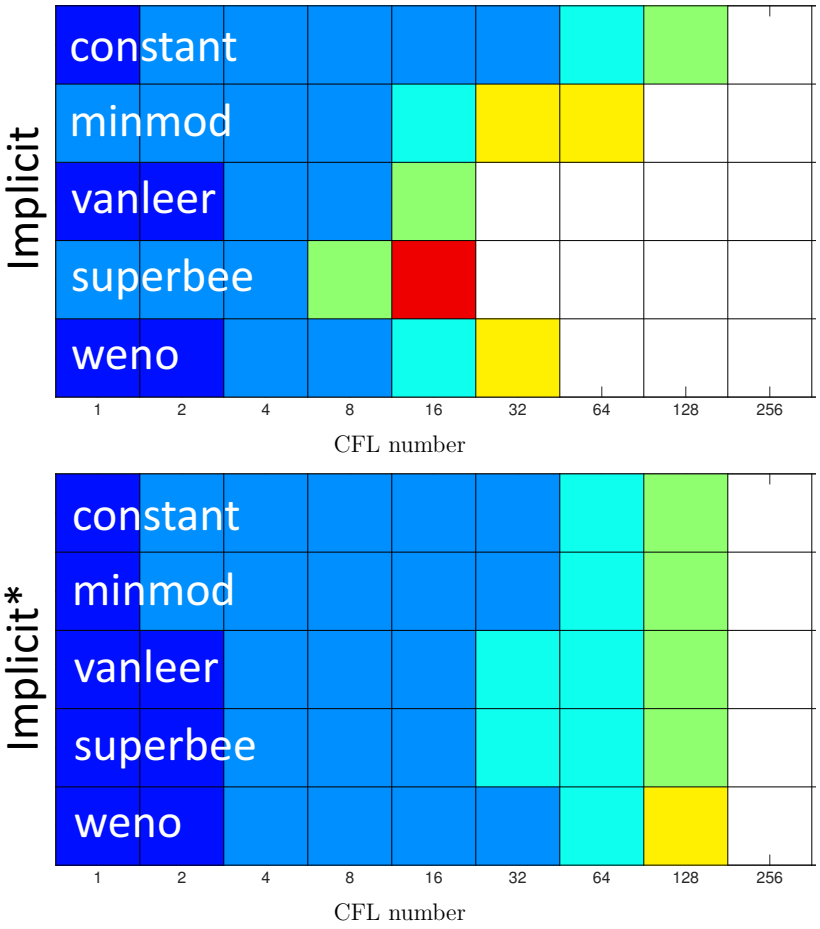
Number of iterations required by the implicit schemes



Number of iterations required by the implicit* schemes

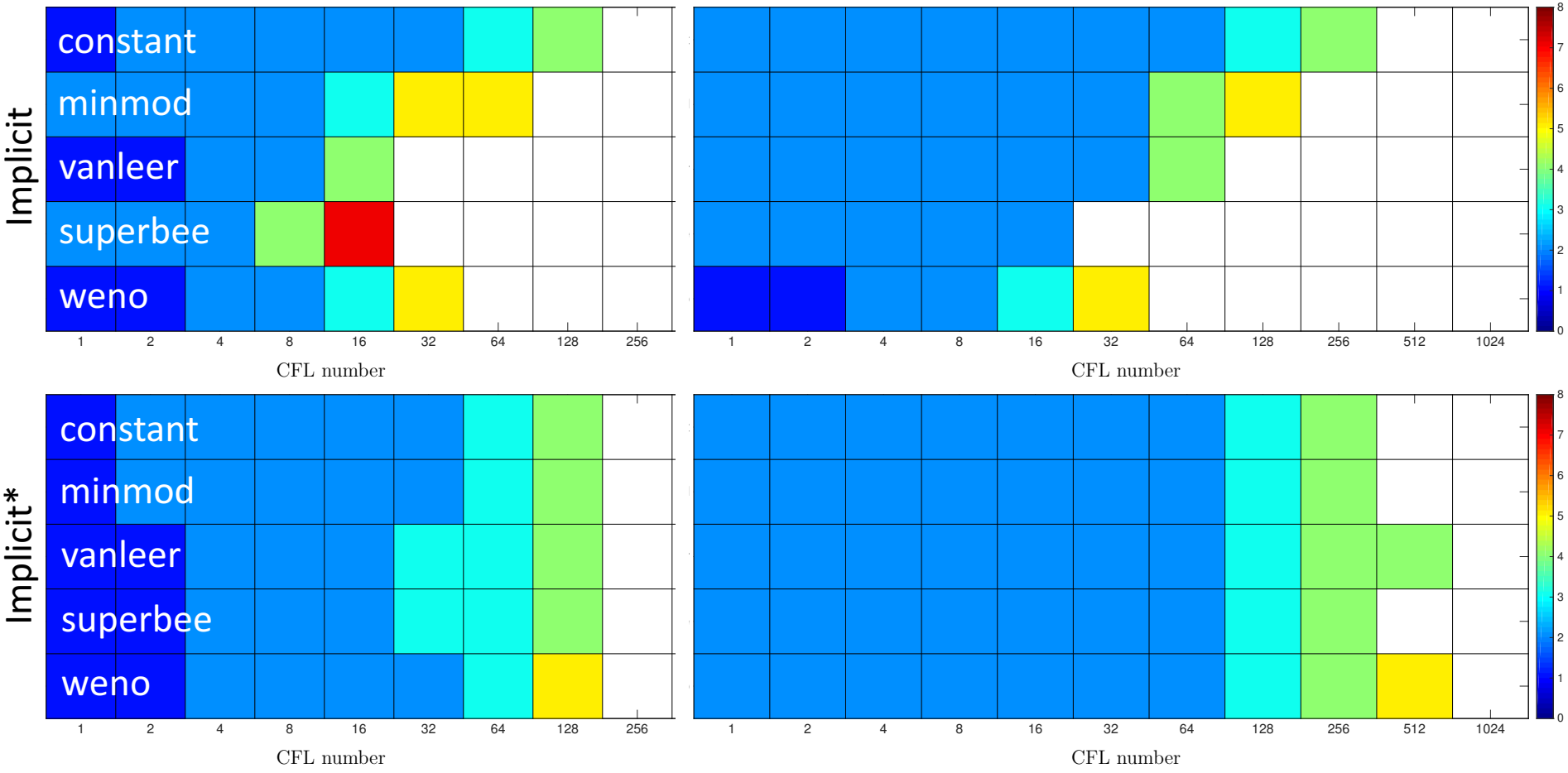


Test of large-time-step capability



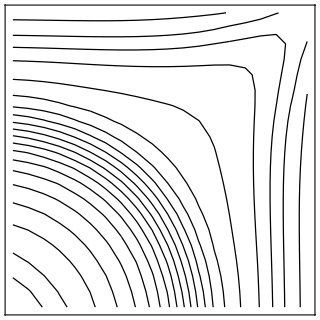
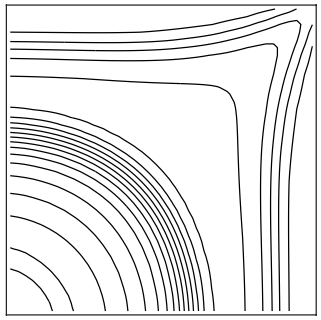
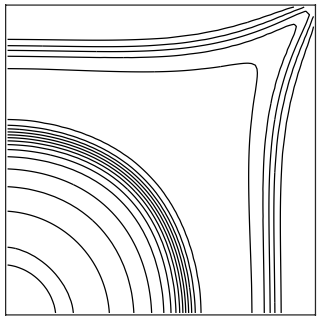
Simulating a single time-step, starting from a well-established displacement profile, 1 day

Test of large-time-step capability

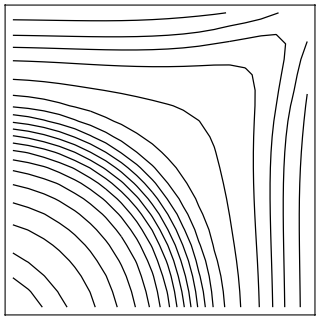
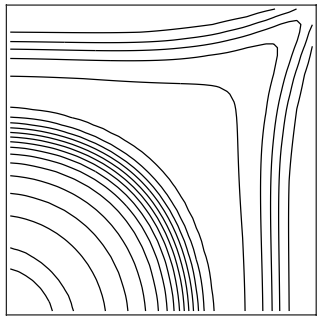
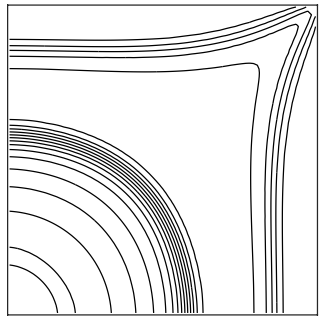
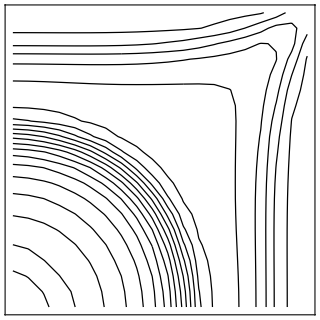
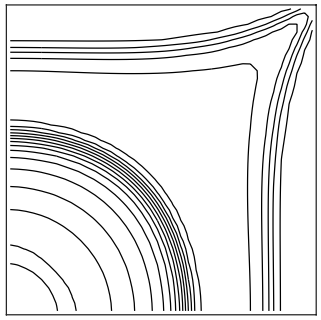
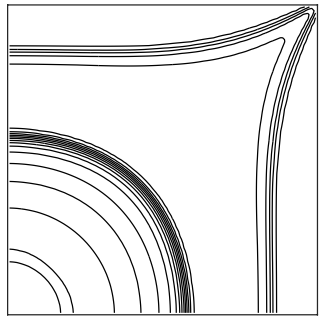


Simulating a single time-step, starting from a well-established displacement profile, 1 day (left) and 5 days (right)

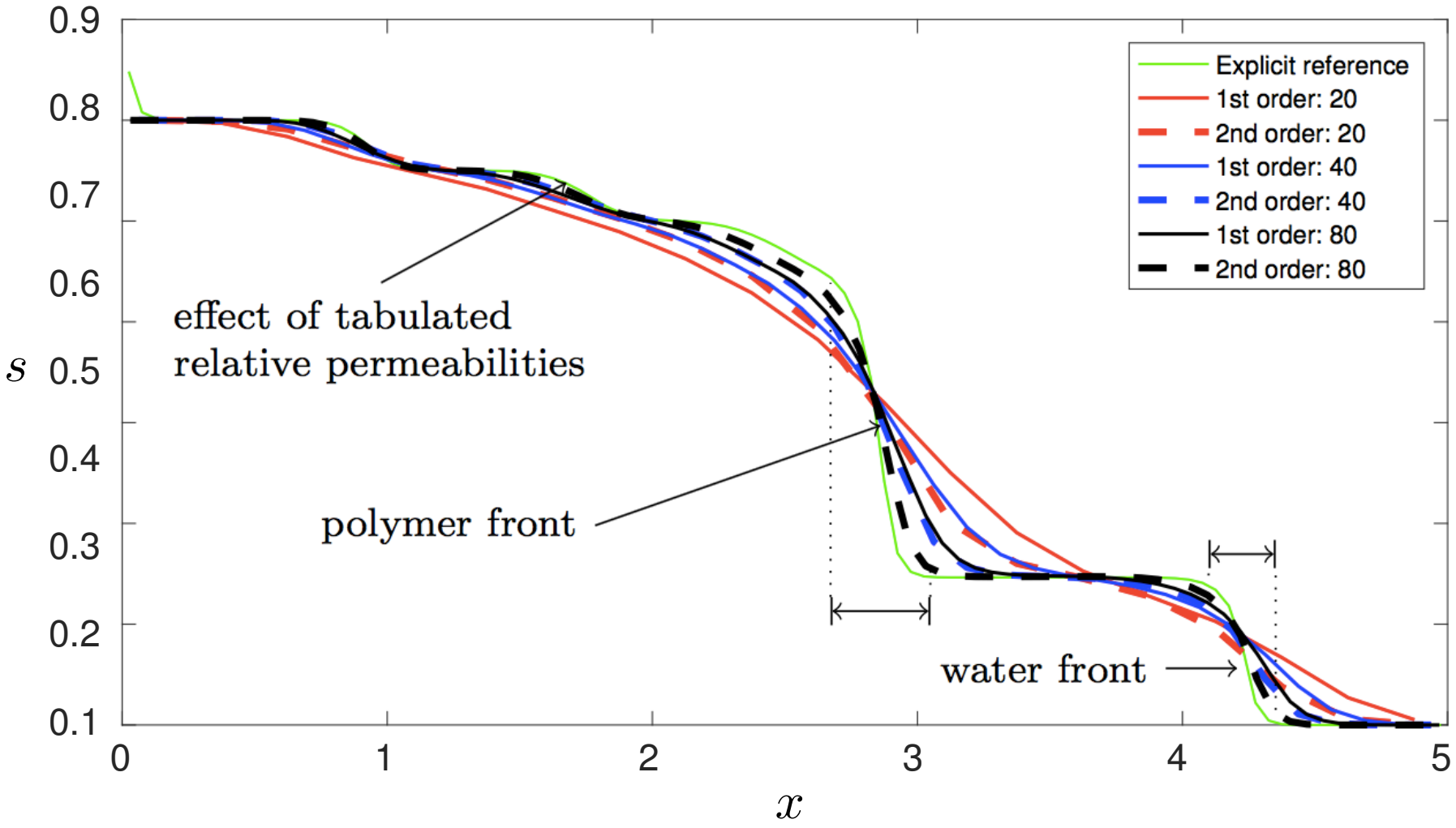
Spatial convergence

	20 x 20	40 x 40	80 x 80
Implicit first-order	constant: 1.6 sec 	constant: 5.9 sec 	constant: 26.7 sec 

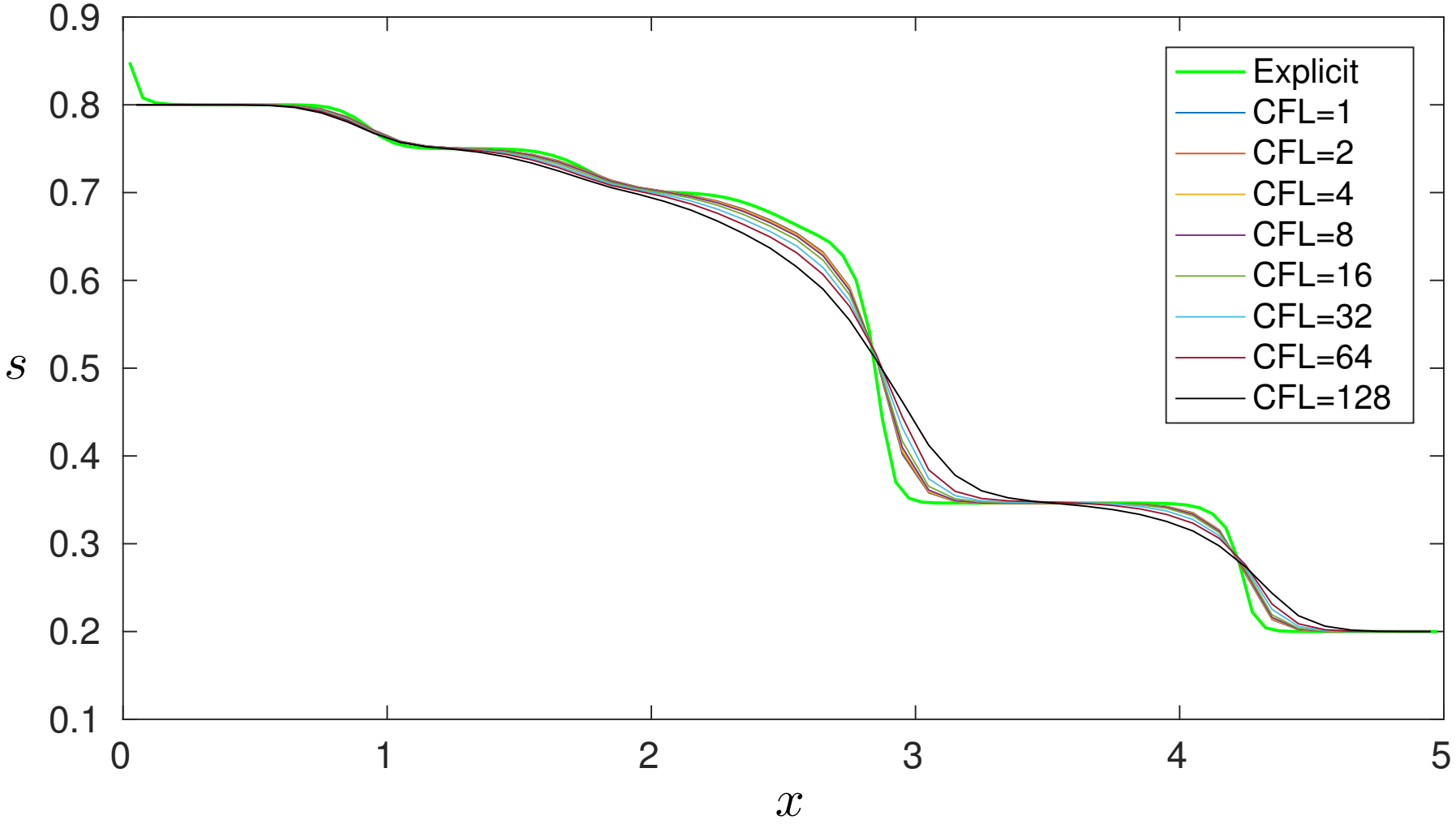
Spatial convergence

	20 x 20	40 x 40	80 x 80
Implicit first-order	constant: 1.6 sec 	constant: 5.9 sec 	constant: 26.7 sec 
Implicit second-order	minmod: 7.9 sec 	minmod: 15.8 sec 	minmod: 74.5 sec 

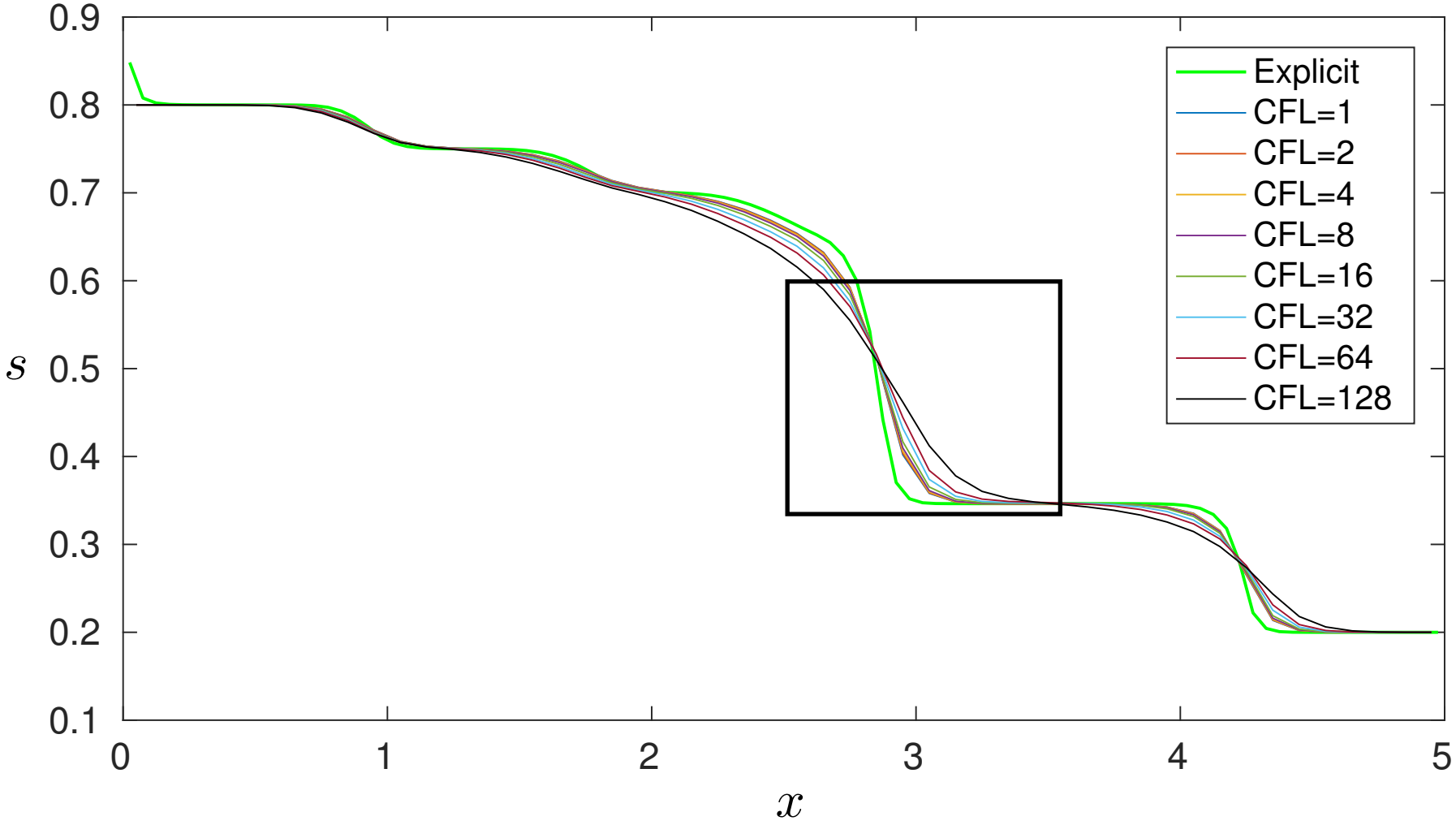
Spatial convergence



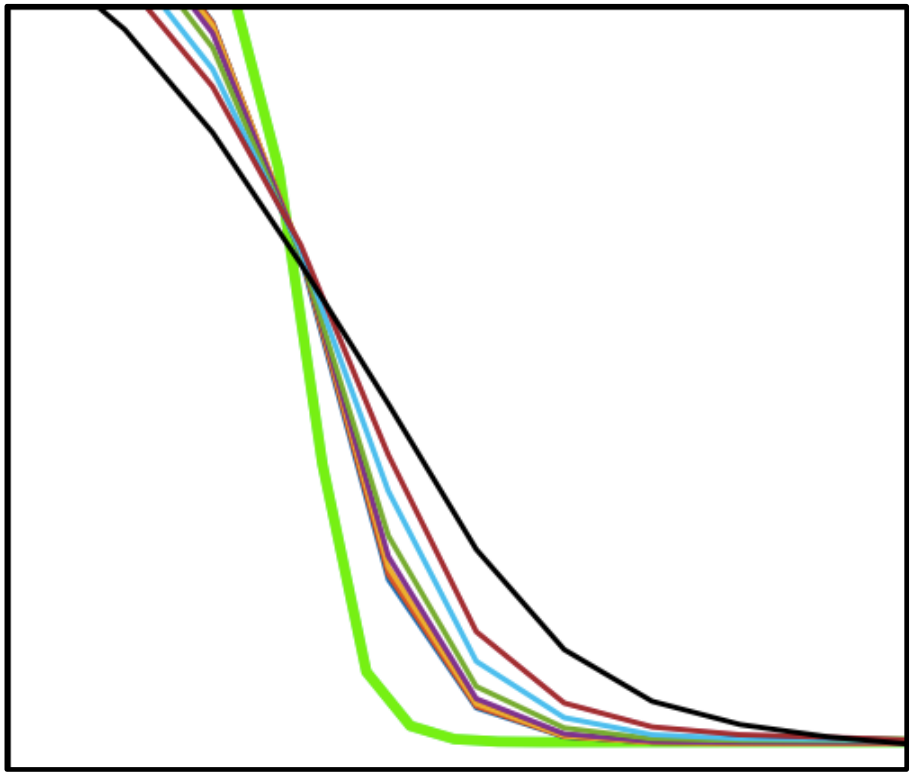
Temporal convergence

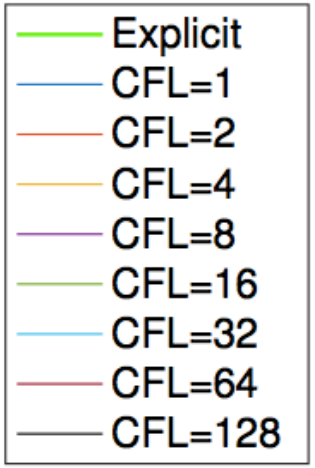


Temporal convergence

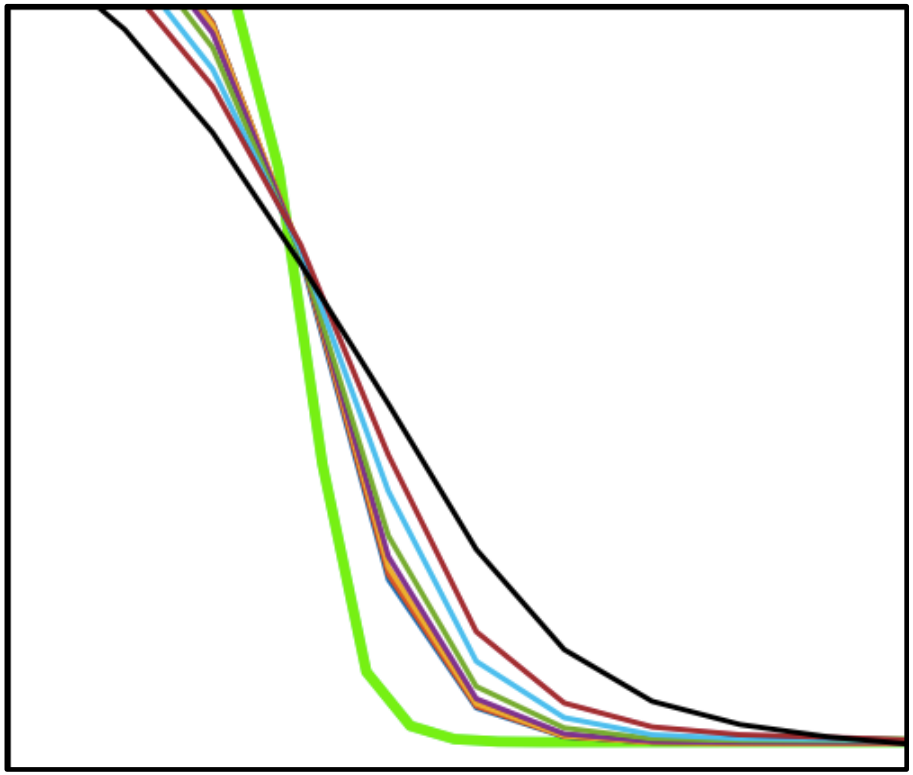


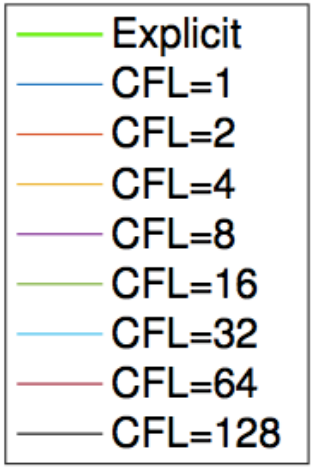
Temporal convergence





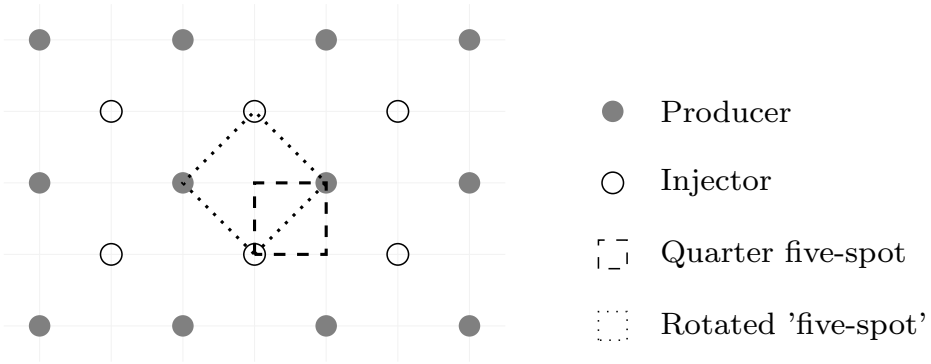
Temporal convergence



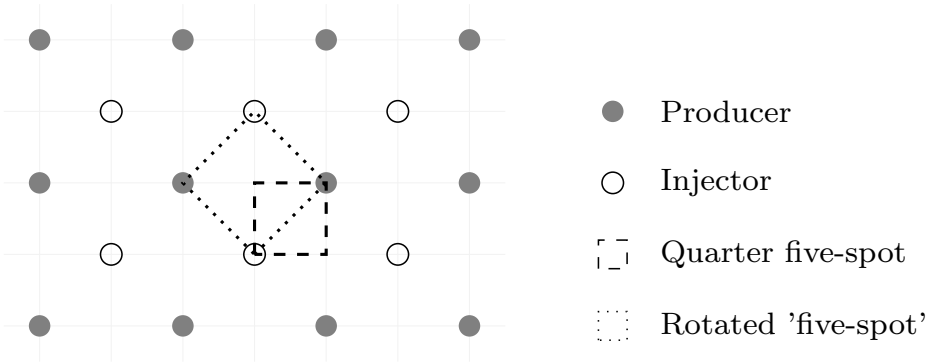


Observe:
can safely increase the CFL number for the implicit scheme to one order of magnitude beyond the stability limit before the numerical dissipation causes significant smearing

Test of grid-orientation errors



Test of grid-orientation errors

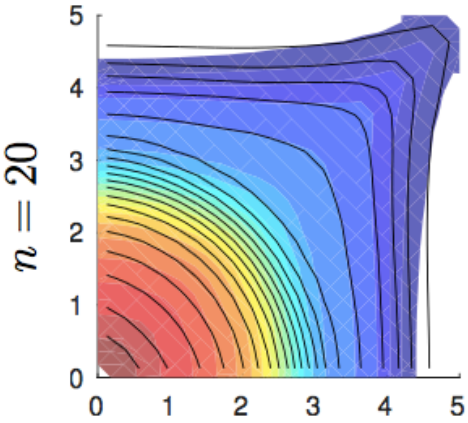


Original setup:
overestimates the frontal movement into stagnant regions, and *underestimates* the movement in the high-flow zones along the diagonal by smearing the tip of the finger

Rotated setup:
overestimates the movement of the front in the high-flow zone, and *underestimates* the movement towards stagnant zones

Test of grid-orientation errors

1st order constant



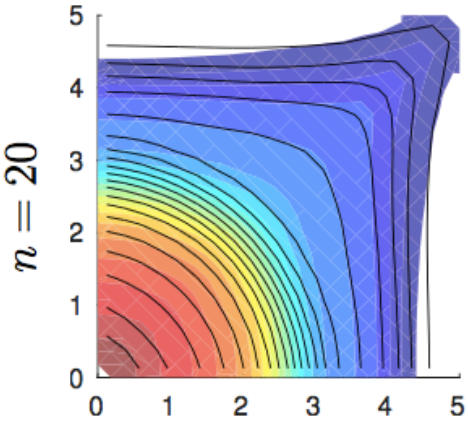
Colors: rotated setup

setup

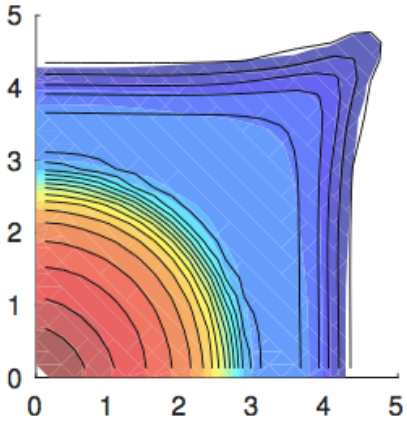
Solid lines: original

Test of grid-orientation errors

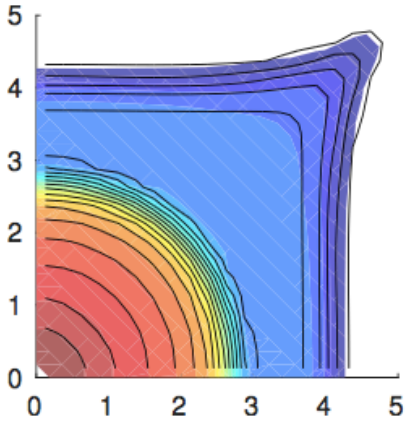
1st order constant



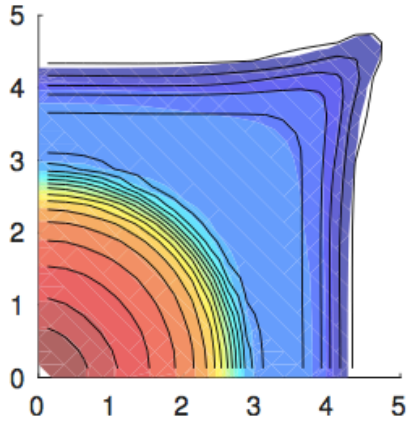
2nd order minmod



2nd order van Leer



2nd order WENO



Colors: rotated setup

Solid lines: original setup

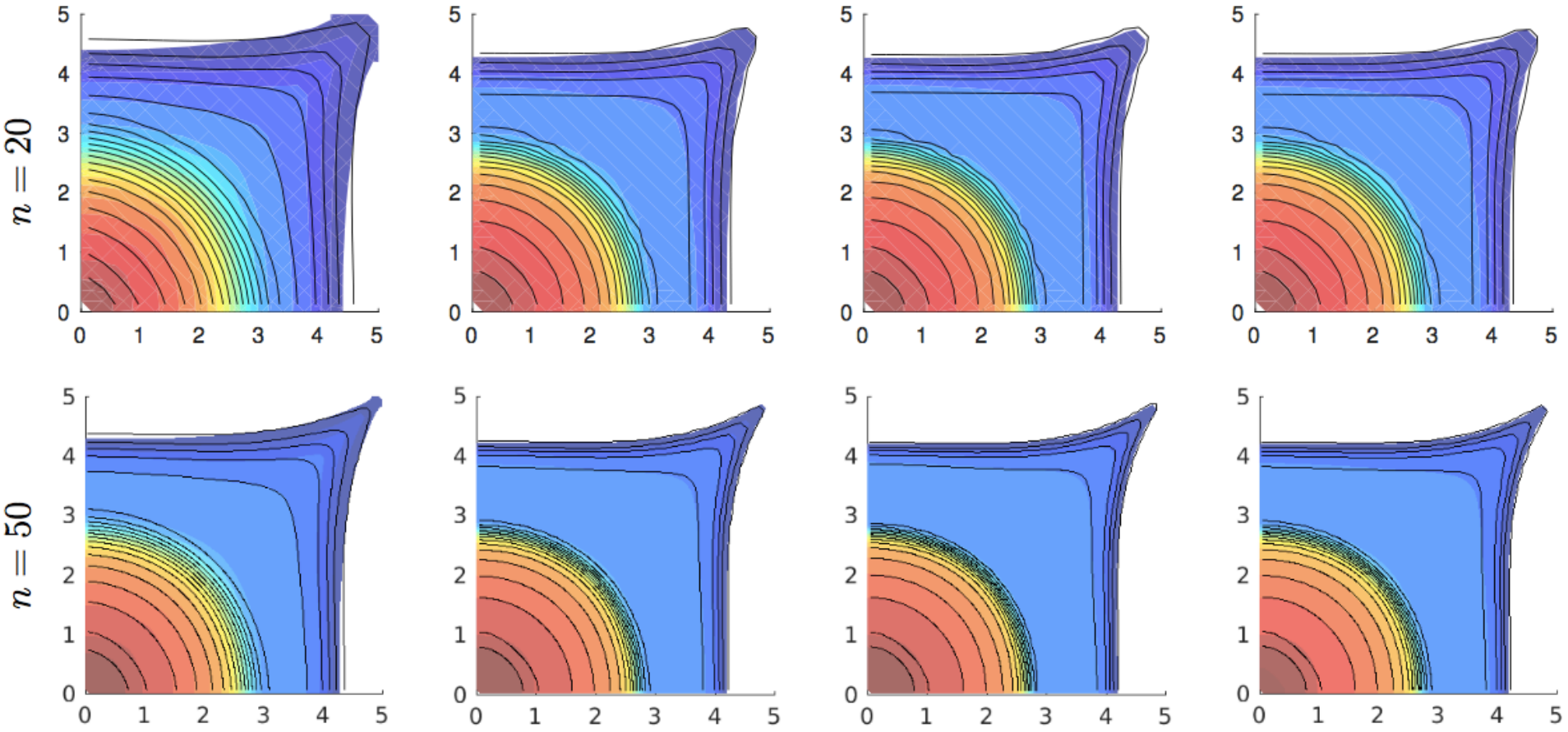
Test of grid-orientation errors

1st order constant

2nd order minmod

2nd order van Leer

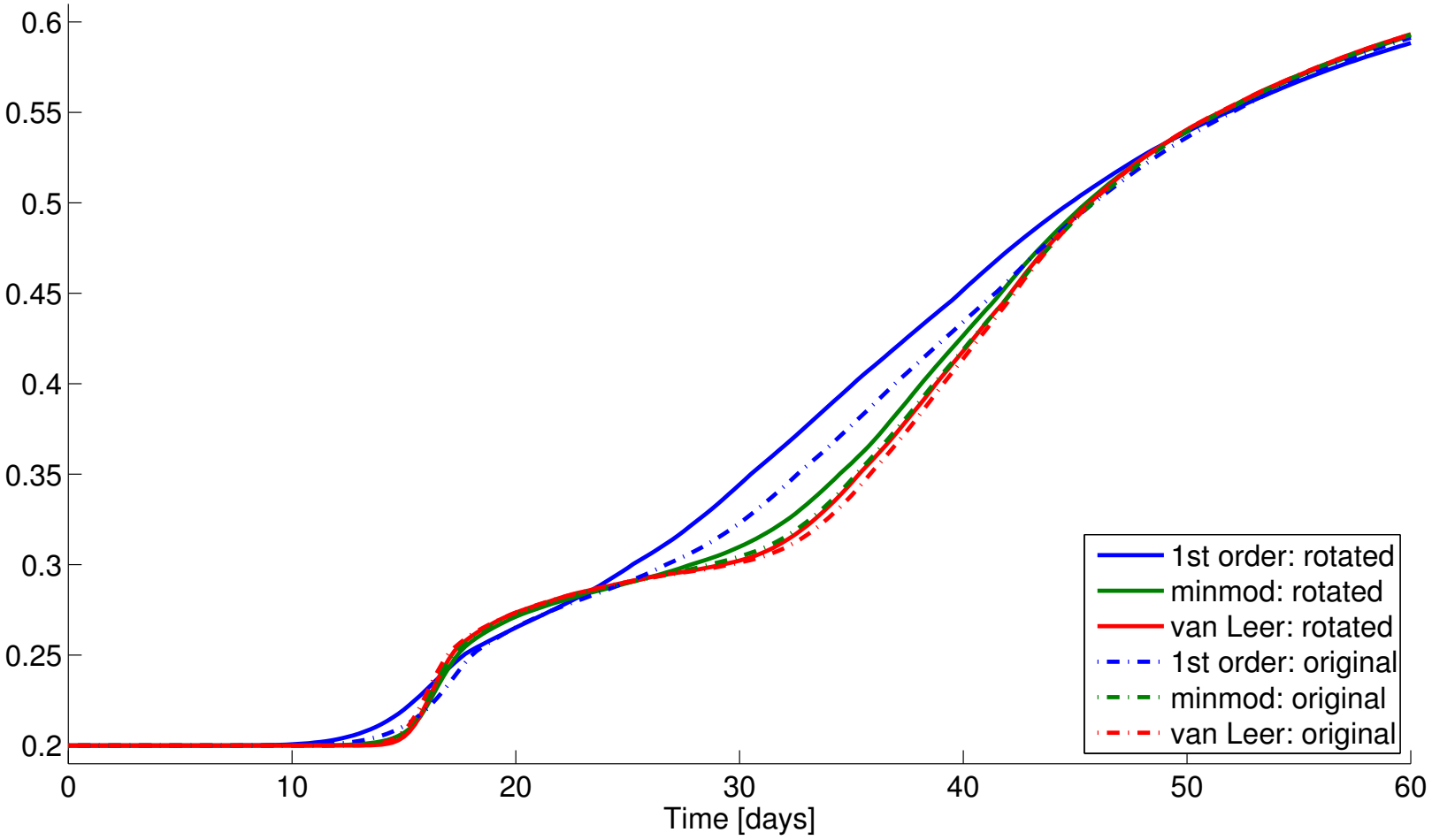
2nd order WENO



Colors: rotated setup

Solid lines: original setup

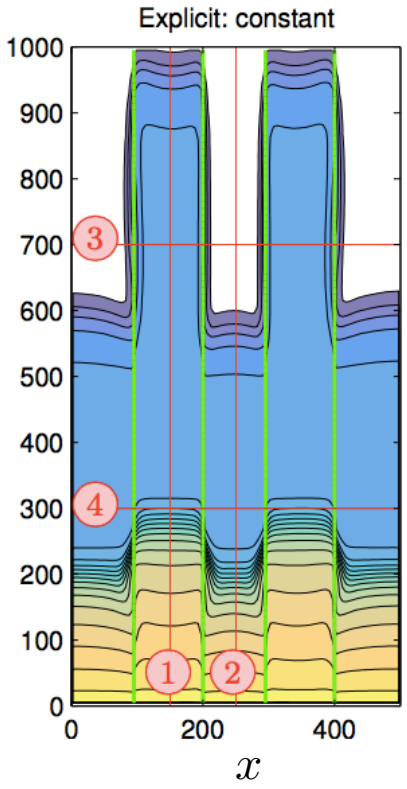
Test of grid-orientation errors



Water saturation in cell containing the producer

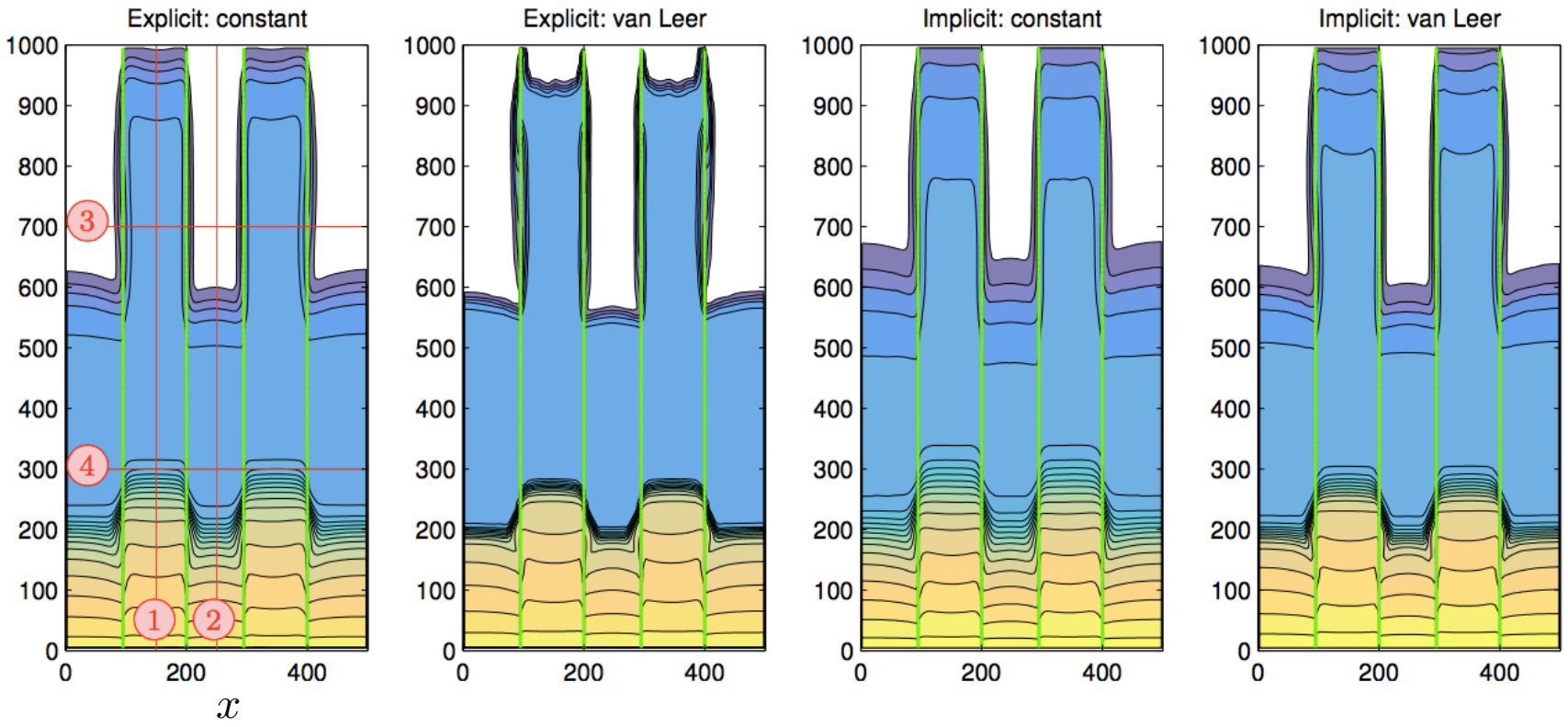
Channelized reservoir

$$(\mathbf{K}, \phi) = \begin{cases} (200 \text{ md}, 0.25), & \text{for } x \in [100, 200] \text{ and } x \in [300, 400], \\ (100 \text{ md}, 0.20), & \text{otherwise.} \end{cases}$$

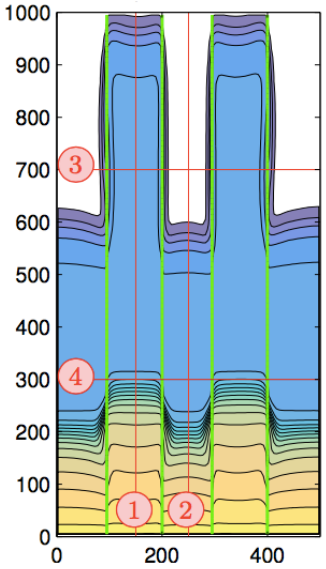


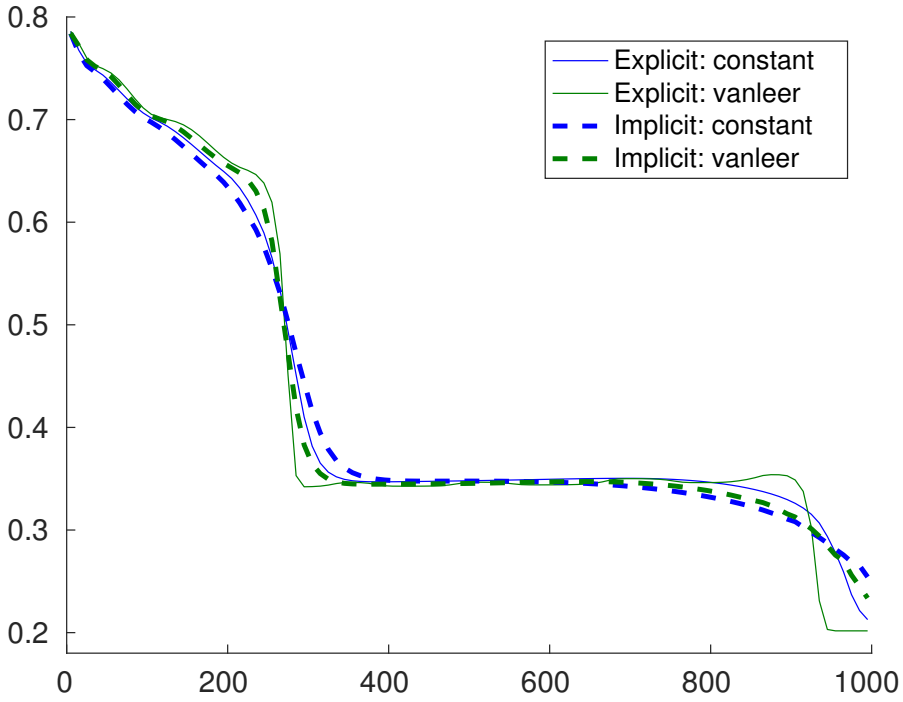
Channelized reservoir

$$(\mathbf{K}, \phi) = \begin{cases} (200 \text{ md}, 0.25), & \text{for } x \in [100, 200] \text{ and } x \in [300, 400], \\ (100 \text{ md}, 0.20), & \text{otherwise.} \end{cases}$$

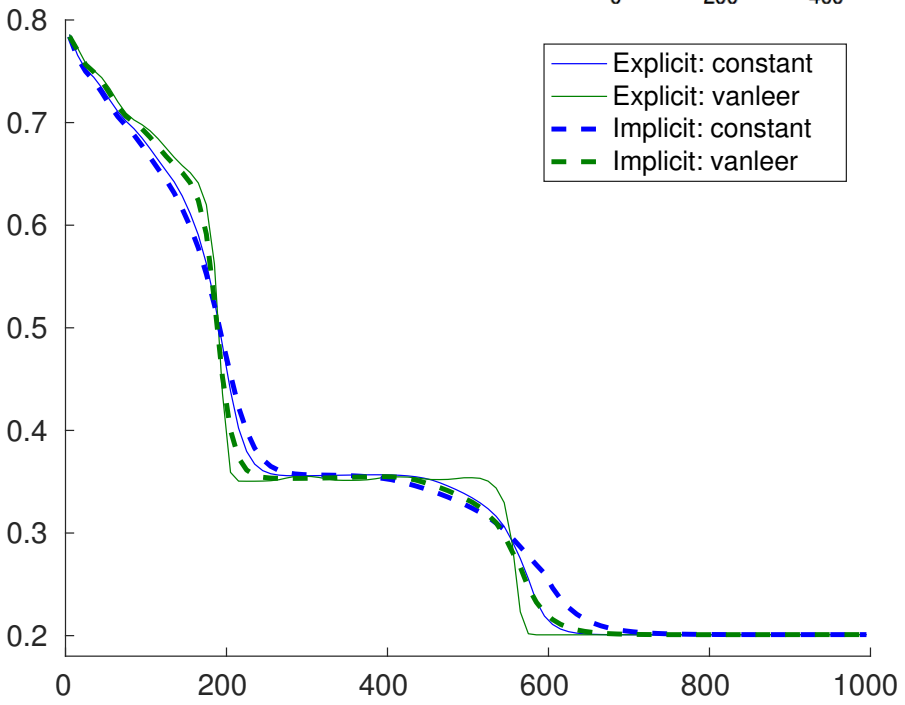


Channelized reservoir



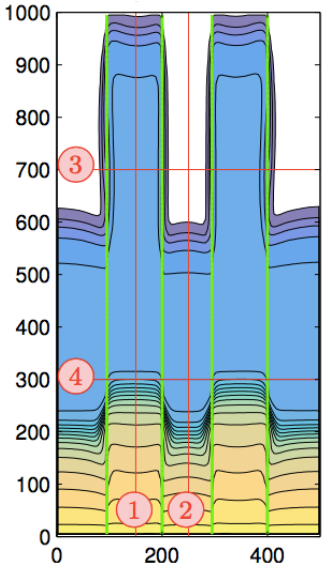


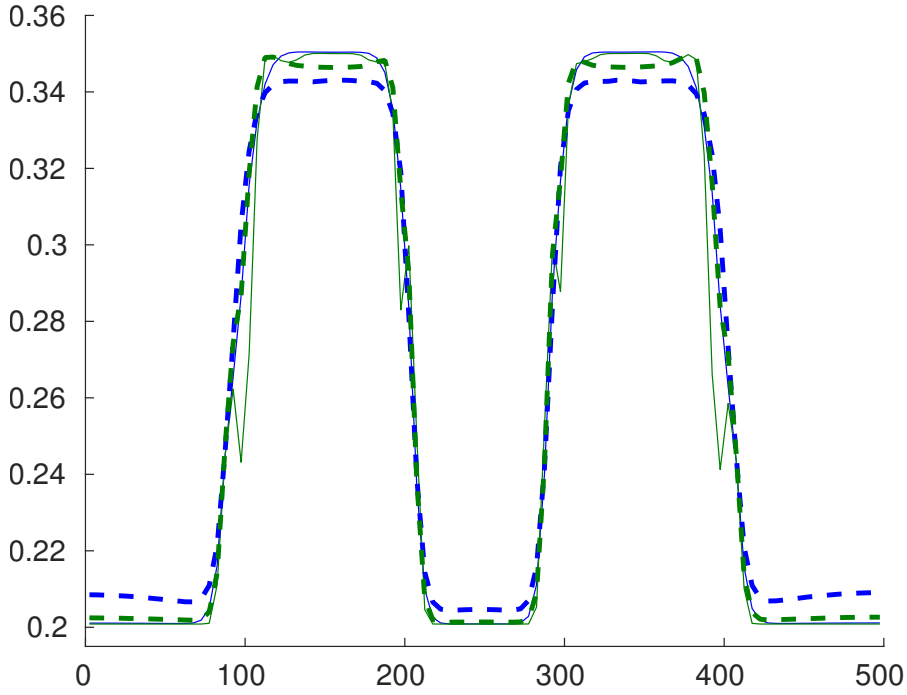
1) $x = 150$



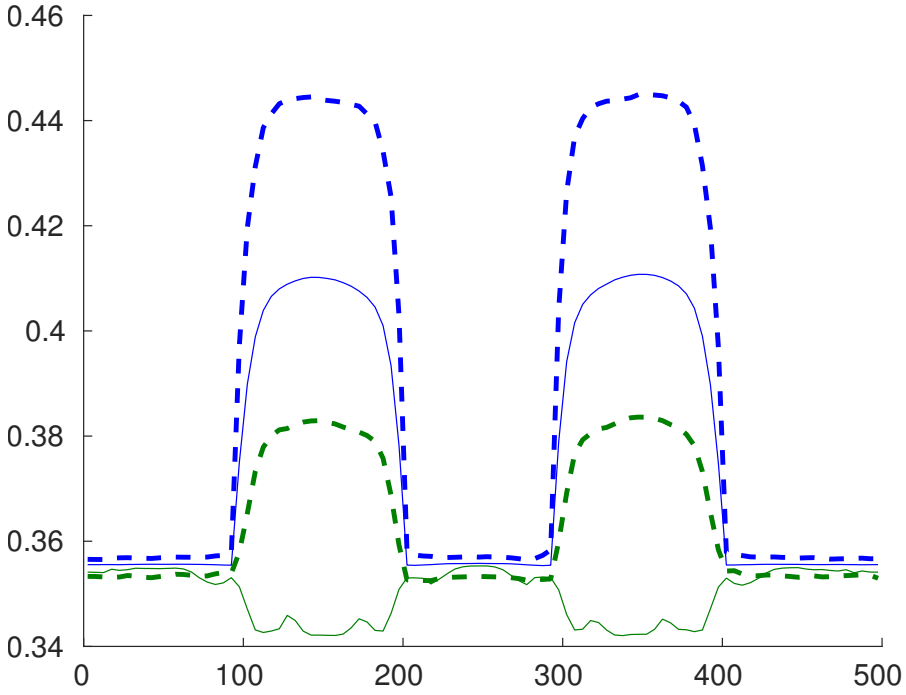
2) $x = 250$

Channelized reservoir





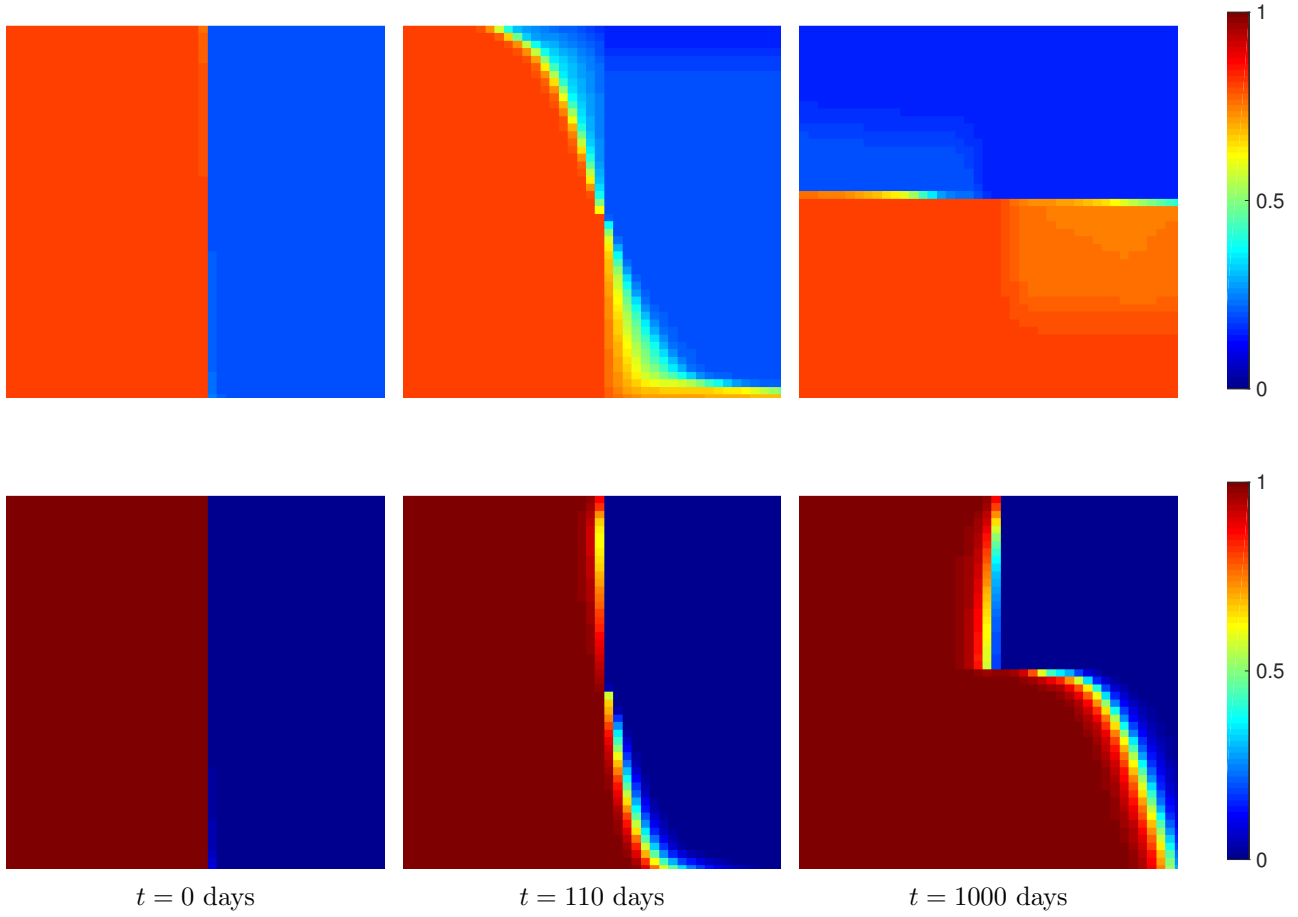
3) $y = 700$



4) $y = 300$

Including buoyancy

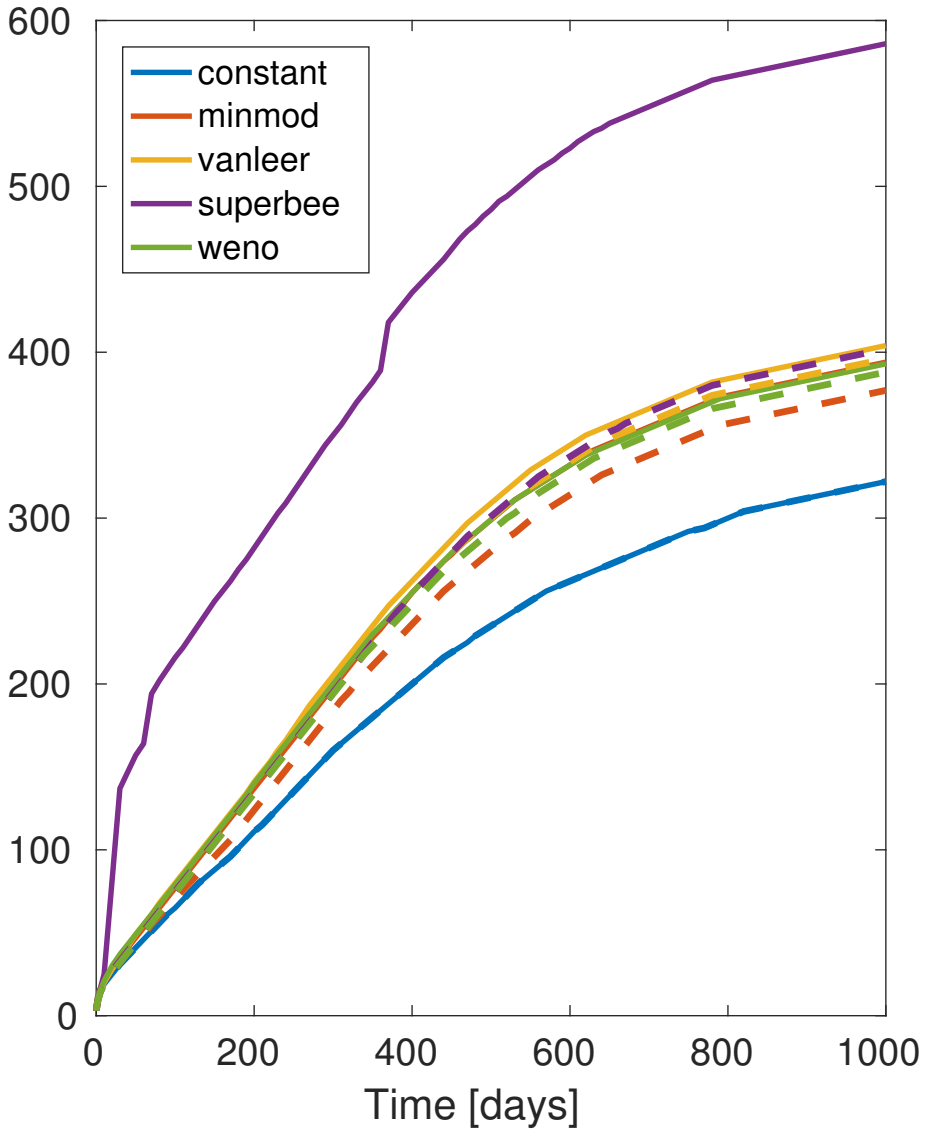
Does including buoyancy make the nonlinear convergence more challenging?



Including buoyancy

Cummulative number of iterations required by the nonlinear solver:

implicit (solid) and implicit* (dashed)



Summary

- Studied explicit and fully implicit schemes with second-order formal spatial accuracy applied to polymer flooding
- The use of high-resolution spatial stencils improves the accuracy and reduces grid-orientation effects
- Demonstrated that implicit time discretization is more suitable than explicit time integration through numerous numerical experiments
- To make high-resolution methods amenable for implicit discretizations, preference should be given to spatial stencils and nonlinear limiter functions that are as smooth as possible to avoid exacerbating the nonlinearity of the implicit flow equations