ARNE NÄGEL[a], PETER DEUFLHARD[b,c] AND
GABRIEL WITTUM[a,d]

[a]*Goethe-Center for Scientific Computing, Goethe University, Frankfurt a.M.*

[b]*Beijing University of Technology, Beijing Institute of Scientific and Engineering Computing*

[c]*Zuse Institute Berlin, Germany*

[d]*King Abdullah University of Science and Technology, Thuwal*

EFFICIENT STIFF INTEGRATION OF DENSITY DRIVEN FLOW PROBLEMS

ZIB Report 18-54 (November 2018)

Zuse Institute Berlin
Takustrasse 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: `bibliothek@zib.de`
URL: `http://www.zib.de`

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Efficient Stiff Integration of Density Driven Flow Problems

Arne Nägel[a], Peter Deuflhard[b,c], Gabriel Wittum[a,d]

[a] *Goethe-Center for Scientific Computing, Goethe University, Frankfurt a.M.*
[b] *Beijing University of Technology, Beijing Institute of Scientific and Engineering Computing*
[c] *Zuse Institute Berlin (ZIB)*
[d] *King Abdullah University of Science and Technology, Thuwal*

Abstract

The paper investigates the efficient use of a linearly implicit stiff integrator for the numerical solution of density driven flow problems. Upon choosing a one-step method of extrapolation type (code LIMEX), the use of full Jacobians and reduced approximations are discussed. Numerical experiments include nonlinear density flow problems such as diffusion from a salt dome (2D), a (modified) Elder problem (3D), the saltpool benchmark (3D) and a real life salt dome problem (2D). The arising linear equations are solved using either a multigrid preconditioner from the software package UG4 or the sparse matrix solver SuperLU.

Contents

1 Introduction **3**

2 The problem class **3**

3 Preliminary considerations **5**
3.1 Choice of Stiff Integrator 5
3.2 Computation of full and reduced Jacobian 7
3.3 Solvers and software infrastructure 8

4 Numerical experiments **9**
4.1 Diffusion from salt dome (2D) 9
4.2 Modified Elder (3D) 10
4.3 Saltpool Problem (3D) 18
4.4 Gorleben problem (2D) 21

5 Conclusion **27**

1. Introduction

Density-driven flow problems appear widely in the mathematical simulation of real life processes. In [1], their numerical solution has been compared in terms of several linear and nonlinear algorithms. The present paper introduces a further suggestion, the application of stiff integrators, here predominantly the quite popular stiff integrator based on the linearly implicit Euler discretization combined with extrapolation (code LIMEX).

The paper is organized as follows. In Section 2, we present the class of nonlinear density driven flow PDEs of interest here, indicating their degenerate character. In Section 3, we give essential algorithmic details: In Section 3.1, we survey possible stiff integrators especially for large scale problems; we end up with suggesting W-methods, among them the linearly implicit extrapolation code LIMEX. Any of the stiff integrators is known to require a Jacobian of the right-hand side or its approximation. Therefore, in Section 3.2, we discuss two choices, a "full" Jacobian version, J_{full}, and a "reduced" Jacobian version, J_{reduced}. Finally, in Section 4, we present comparative numerical experiments with the three stiff integrators IMPEX, LIMEX$^{\text{full}}$, and LIMEX$^{\text{reduced}}$ for four problems, which are treated in the order of the increase in complexity: *Diffusion from a saltdome* (2D) in Section 4.1, a *modified Elder* problem (3D) in Section 4.2, and the *saltpool problem* (3D) in Section 4.3 serve as benchmark problems. Finally, Section 4.4 is devoted to the so-called *Gorleben* problem, a real-world problem in a heterogeneous domain that had not been solved efficiently before.

2. The problem class

The governing equations for density driven flow are the conservation of salt mass and fluid, which read

$$\frac{\partial}{\partial t}(\phi\rho\omega) + \nabla\cdot[\omega\rho\vec{q} - \rho\mathbb{D}\nabla\omega] \;=\; \omega\rho Q\ , \tag{1a}$$

$$\frac{\partial}{\partial t}(\phi\rho) + \nabla\cdot[\rho\vec{q}] \;=\; \rho Q\ . \tag{1b}$$

where ω denotes the salt-mass-fraction, and ϕ and Q denote spatially dependent porosity and fluid sources and sinks respectively. The system is closed by constitutive laws for the Darcy flow

$$\vec{q} = -\frac{k}{\mu}(\nabla p - \rho\vec{g}) \tag{2}$$

which introduces the gradient of the pressure p as the second unknown into the system. Moreover, mechanical dispersion (Scheidegger-type) is given by

$$\mathbb{D} = \phi \mathbb{D}_{mol} + \alpha_T |\vec{q}| + (\alpha_L - \alpha_T) \frac{\vec{q} \cdot \vec{q}^T}{|\vec{q}|} \ . \tag{3}$$

Here, $\mathbb{D}_{mol}$ is the molecular diffusion coefficient, while α_T and α_L denote the dispersion lengths in transversal and lateral direction, respectively.

The density ρ and the viscosity μ depend on the salt mass fraction ω and are given by empirical relations. For the density this work considers the linear relationship

$$\rho(\omega) = \rho_0 + (\rho_1 - \rho_0) \frac{\omega}{\omega_{max}} \tag{4}$$

where ω_{max} denotes the salt mass fraction of a saturated haline solution. The viscosity is given by either of the following:

$$\mu_{\text{const}}(\omega) = \mu_0 \tag{5a}$$
$$\mu_{\text{real}}(\omega) = \mu_0 \left(1 + 1.85\omega - 4.1\omega^2 + 44.5\omega^3\right) \tag{5b}$$

Upon combining all these relations, the PDE system (1) is nonlinearly coupled.

Resulting system after spatial discretization. Due to the lack of time derivatives in the equation for the pressure p the PDE system appears to be *degenerate.* After discretization of this PDE system by the *method of lines* one arrives at a differential algebraic (DAE) system of the kind

$$M(u)\, u' = f(u), \quad u(0) = u_0 \ , \tag{6}$$

wherein $M(u)$ represents a solution dependent matrix, which is *singular* due to the fact that the PDE system is degenerate. In view of uniqueness of the solution the matrix pencil $\{M(u) - \tau J(u)\}$ must be regular, where J means

$$J(u) = J_f(\mathbf{u}) - \Gamma[\mathbf{u}, \mathbf{u}']$$

with

$$J_f(\mathbf{u}) := \left(\frac{\partial f(\mathbf{u})}{\partial u} \right) \quad \text{and} \quad \Gamma[\mathbf{u}, \mathbf{z}] := \left(\frac{\partial M(\mathbf{u})}{\partial u} z \right) \ . \tag{7}$$

3. Preliminary considerations

In this section, we want to display the most important algorithmic parts that are necessary to attack the challenging class of problems that we have in mind here.

3.1. Choice of Stiff Integrator

Suppose we are focusing on the regime where the PDE systems appear to be parabolic and large scale. As for the choice of a stiff integrator, we typically have the following options, see, e.g., the textbooks by Hairer/Wanner [2] or Deuflhard/Bornemann [3] .

- *Implicit one-step or multistep methods:*
  This class of algorithms covers the BDF method as a multistep method or the Radau method as a one-step method. In both cases, the realization of three nested loops is needed: the *outer* discretization loop (adaptive time step), the *medium-level* Newton-like iteration loop for the solution of the nonlinear system, and the *inner* loop for solving the linear equations to compute the Newton corrections.

- *Implicit Euler discretization:*
  In an earlier algorithmic approach [1], the problem class of interest here had been attacked by the simplest implicit one-step method, an implicit Euler discretization. That approach had been combined with extrapolation of order 2 and a heuristic time step control. In passing we note that with that version of stiff integrator the Gorleben problem addressed in Section 4.4 could not be solved.

  In contrast, the code IMPEX included in the numerical experiments in Section 4 below uses the same implicit discretization, but an adaptive order and time step control based on extrapolation, see, e.g., [3]. Convergence of the Newton-type iteration in each discretization step has been assumed, as soon as the outer residual has been reduced by 7 orders of magnitude.

- *Linearly implicit methods of Rosenbrock-Wanner type (ROW):*
  This class of algorithms only requires two loops, the *outer* loop for an (adaptive) time step discretization and the *inner* loop for the numerical solution of the arising linear systems. An *exact* Jacobian matrix is

required, an information that enters into the derivation of the algebraic equations for the coefficients.

- *Linearly implicit W-methods:*
  This class of algorithms also only requires two loops, the *outer* loop for an (adaptive) time step discretization and the *inner* loop for the numerical solution of the arising linear systems. In contrast to the ROW class above, only a reasonable approximation of the Jacobian is needed. Unfortunately, the construction of this class of methods leads to a blow-up in the number of the arising algebraic conditions for the coefficients. For this reason, only low orders have been developed by the scientific community.

- *Linearly implicit extrapolation methods:*
  These algorithms can be formally subsumed under W-methods, but do not require any algebraic conditions for coefficients. They, too, realize a certain freedom of choice of the Jacobian approximation.

Remark. In passing we note that, for incompressible Navier-Stokes problems, a long list of ROW methods has been collected in [4, Table 1, 4-8], which, however, excludes linearly implicit extrapolation methods.

Specification of the integrator LIMEX. In view of our challenging differential-algebraic problem class (6), we here selected the popular integrator LIMEX, based on a linearly implicit Euler method with adaptive time step and order control. The corresponding discretization has been worked out in detail in [5], see also the textbook [3]. In our numerical experiments below, we use the notation $\mathsf{LIMEX}_{q_{\max}}$ to indicate the maximum permitted number $q_{\max}$ of columns in the extrapolation table. Throughout the paper, the error tolerance for time integration has been set to $TOL = 0.001$. All of the LIMEX variants feature the usual sub-diagonal error criterion (see [3]). In order to estimate the error of the variable $u(t) = (\omega, p)^T(t)$, we use the scaled norm

$$|||(\omega, p)|||^2 := \|\frac{\kappa}{\mu_0}\nabla p(t)\|_\rho^2 + \|\vec{k}\omega(t)\|_\rho^2, \tag{8}$$

where $\|\vec{v}\|_\rho^2 := \int_\Omega \rho^* \vec{v}^2$ measures the kinetic energy of a velocity field $\vec{v}$ w.r.t a reference density ρ^*. The scaling in the second term is provided by the filter velocity

$$\vec{k}(x) \quad := \tfrac{\kappa(x)}{\mu_0}\alpha\vec{g}, \tag{9}$$

where $\alpha \approx \frac{\partial \rho}{\partial \omega}$. This choice can be motivated by the Darcy velocity (2): For the linear density (4) and constant viscosity (5a), one readily verifies

$$\|\vec{q}_1 - \vec{q}_2\|_\rho \leq \sqrt{2}\max(1, \frac{\rho_1 - \rho_0}{\alpha\omega_{\max}})|||(\omega_1 - \omega_2, p_1 - p_2)||| \ .$$

Thus convergence w.r.t $|||\cdot|||$ directly induces convergence of the Darcy velocity.

Order reduction. As theoretically analyzed by Lubich/Ostermann [6], LIMEX as a W-method will exhibit an order reduction to an "effective" order $p^* = 2$, as soon as the quasi-stationary solution has been reached, for more details see also Section 9.3.1 in the textbook [7]. It is not that easy to algorithmically identify the neighborhood of the quasi-stationary solution in a nonlinear PDE system. In view of this feature, the following heuristic strategy has been implemented: As soon as the suggested order q has been successfully reduced for consecutive two (or three) time steps to $q = 2$, the future order is fixed to $q = p^* = 2$. For step size selection, we use

$$\tau_{\text{new}} = \sqrt[q+1]{\rho\frac{\text{TOL}}{[[\epsilon]]}}\tau_{\text{old}}$$

where $\rho = 0.25$ is a safety factor, TOL is a user-provided tolerance, and $[[\epsilon]]$ is the estimate for the relative error w.r.t. (8). Note that this is a classic proportional controller. Experiments with controllers considering a longer time history (see, e.g., [8, 9]) turned out to be less successful in our tests.

3.2. Computation of full and reduced Jacobian

For the DAE system (6) Deuflhard/Nowak [5] suggest the following step:

$$(M_k - \tau J_0)(u_{k+1} - u_k) = \tau f(u_k)\,, \tag{10}$$

where $M_k = M(u_k)$ and $J_0 = J(u_0)$ according to (7).

In our PDE context, the Jacobian J_0 will be most easily computed via solving the linearized perturbation equations. Upon inserting

$$(\omega, p)^T := (\omega_0, p_0)^T + (\delta\omega, \delta p)^T \ ,$$

and considering the linearization of equation (1), we obtain a system for the increment $(\delta\omega, \delta p)^T$ that reads

$$\frac{\partial}{\partial t}((\rho_0 + \omega_0\rho_0')\delta\omega) + \nabla\cdot\left[(\rho_0\vec{q}_0)\delta\omega + \omega_0\left\{\frac{\partial}{\partial\omega}(\rho\vec{q})\delta\omega + \frac{\partial}{\partial p}(\rho\vec{q})\delta p\right\}\right.$$
$$\left. - (\rho_0\mathbb{D}_0)\nabla\delta\omega - \left\{\frac{\partial}{\partial\omega}(\rho\mathbb{D})\delta w + \frac{\partial}{\partial p}(\rho\mathbb{D})\delta p\right\}\nabla\omega_0\right] \tag{11a}$$
$$= (\rho_0 + \omega_0\rho_0')Q\delta\omega$$
$$\frac{\partial}{\partial t}(\rho_0'\delta\omega) + \nabla\cdot\left[\frac{\partial}{\partial\omega}(\rho\vec{q})\delta\omega + \frac{\partial}{\partial p}(\rho\vec{q})\delta p\right]$$
$$= (\rho' Q)\delta w \tag{11b}$$

The system decouples in the variables ω and p whenever

$$\omega_0\frac{\partial}{\partial p}(\rho\vec{q})\delta p = \omega_0\rho_0\frac{\partial\vec{q}}{\partial p}\delta p = 0 \ ,$$

and

$$\frac{\partial}{\partial p}(\rho\mathbb{D})\delta p\nabla\omega_0 = \rho_0\frac{\partial\mathbb{D}(\vec{q})}{\partial\vec{q}}\frac{\partial\vec{q}}{\partial p}\delta p\nabla\omega_0 = 0 \ .$$

Analogously, one can expect a block decoupling in the arising matrix after discretization, which will be beneficial for the linear solver. For a W-method like LIMEX, we can achieve this by using an inexact Jacobian J_{reduced}. J_{reduced} is constructed as follows: While assembling (11a), we use the assumption

$$\frac{\partial\vec{q}}{\partial p} = 0 \text{ and } \frac{\partial\vec{q}}{\partial\omega} = 0. \tag{12}$$

Note that, formally, the previous condition would be sufficient for the decoupling. However (12) proved to be more stable experimentally. In the remainder, a superscript indicates, how the Jacobian is approximated. LIMEX$^{\text{full}}$ refers to an exact Jacobian, whereas LIMEX$^{\text{reduced}}$ refers to an inexact Jacobian.

3.3. Solvers and software infrastructure

Multigrid framework UG4. All experiments were performed in the package d^3f++ of the UG4 software framework [10, 11]. At the core of UG4 are robust both parallel and adaptive multigrid methods as introduced in [12]. Note that multi-grid methods are optimal order solvers using only $\mathcal{O}(n)$ work amount to solve the linear systems of the type (10).

The underlying PDEs represent a degenerate parabolic system. Although the properties of linear discrete operators cannot be predicted exactly due to the nonlinearities, it is reasonable to assume that the basic properties carry over to the discrete case. Applicable smoothers used in the present paper are Jacobi, (symmetric) Gauss-Seidel, and ILU iterations. For the coupled systems, corresponding point-block versions have been devised by locally grouping all unknowns belonging to the same node. It should be emphasized that for $\mathsf{LIMEX}^{\text{reduced}}$, the arising linear system decouples and could, in principle, be treated by a sequential iteration solving first for ω and second for p. However, this property has not been explicitly exploited in the tests reported here.

SuperLU as alternative sparse matrix preconditioner. In some cases, we compare the performance of the multigrid method with a SuperLU solver [13].

Solver tolerances. Solver tolerances are a relative reduction of the residual by 10^{-8} (absolute: 10^{-9}) for the Newton iteration, and a relative reduction of the residual by 10^{-10} (absolute: 5×10^{-12}) for the linear solver. In general, linear systems were solved using preconditioned BiCGStab [14].

4. Numerical experiments

In this section, we will present comparative numerical results obtained with the algorithms IMPEX and LIMEX. We first consider three problems with homogeneous permeability coefficients, i.e. diffusion from a salt dome in 2D, as well as the Elder benchmark [15] in slightly modified form, and the saltpool benchmark [16, 17] in 3D. The underlying geometry is given by rectangular quadrilateral and hexahedral grids respectively. These basic tests, which are documented in Subsections 4.1 – 4.3, were performed on a single cluster node of Intel® Xeon® Prozessor E5-2660 v2 @ 2200 MHz. As a final step, we then consider a complicated field test with variable permeability on unstructured grids in 2D. The simulations were performed on the Hazel Hen supercomputer. For all problems, we explore reasonable combinations of $q \leq q_{max}$. Consistent velocity approximation has been used in all experiments.

4.1. Diffusion from salt dome (2D)

For this problem, the domain is given by $\Omega_2 := (0, 600) \times (0, 150) \subset \mathbb{R}^2$. We employ the Dirichlet condition $\omega(\vec{x}) = 1$ for $\vec{x} \in \Gamma_D^{BOT}$ on the part of the bottom boundary given by

$$\Gamma_{D,2}^{BOT} := \{(x_1, x_2) \in \partial\Omega : 150 \leq x_1 \leq 450, x_2 = 0\ .\}.$$

The initial condition is $\omega = 0$. For this setup, we compare time-stepping strategies and two different preconditioners. The geometry is given by $128 \times 512 = 65,536$ square elements, corresponding to $132,354$ degrees of freedom.

SuperLU preconditioner. The time stepping history is provided in Fig. 1: The methods IMPEX, $\mathsf{LIMEX}_2^{\text{full}}$, and $\mathsf{LIMEX}_2^{\text{reduced}}$ suggest similar time step sizes (Fig. 2a) and thus require almost identical number of time steps (Tab. 1a). As can be expected, the SuperLU preconditioner is always convergent. Since LIMEX avoids an additional linearization loop, its computational effort is reduced. The factor is slightly larger than the factor ≈ 0.5 that can be deduced from the number of iteration calls. This is due to additional assembly calls for the Jacobian. $\mathsf{LIMEX}_3^{\text{full}}$ and $\mathsf{LIMEX}_4^{\text{full}}$ are efficient over the whole time interval, which may be an indicator for the high regularity of the regularity of the problem. The algorithms with reduced Jacobian perform slightly worse: $\mathsf{LIMEX}_3^{\text{reduced}}$ takes slightly smaller time steps than $\mathsf{LIMEX}_3^{\text{full}}$. Similarly, $\mathsf{LIMEX}_4^{\text{reduced}}$ is almost on par with $\mathsf{LIMEX}_4^{\text{full}}$ initially. However, within a few time steps after $t = 10a$, the order first degradates to $q = 3$, and then to $q = 2$, even before the asymptotic order reduction phase (cf. Fig. 1c).

Multigrid preconditioner. Employing multigrid yields almost identical results (Fig. 2). Only in two instances for $\mathsf{LIMEX}_4^{\text{full}}$, the multigrid solver does not converge. We note that even for this small number of degrees of freedom, the multigrid solver is $3-4$ times faster than SuperLU w.r.t. wall clock timing in our experiments.

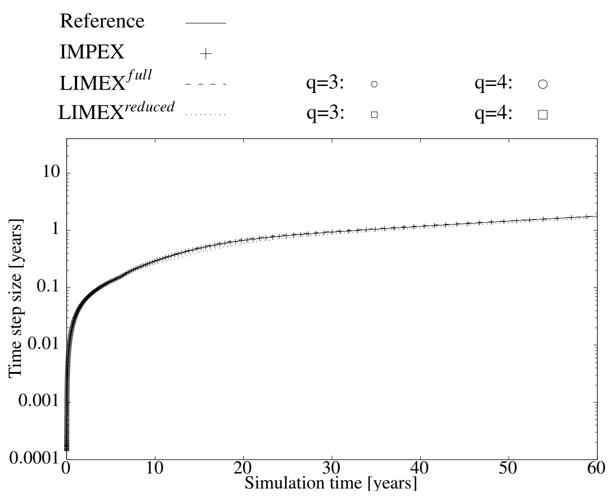
4.2. Modified Elder (3D)

The next test is a variation of the Elder problem [15] in the three-dimensional domain $\Omega_3 := (0, 600) \times (0, 600) \times (0, 150) \subset \mathbb{R}^3$. The transient solution is sensitive w.r.t. the grid and discretisation (see, e.g., [18]). Following [19] we use modified boundary and initial conditions: Dirichlet boundary conditions are provided on the upper part

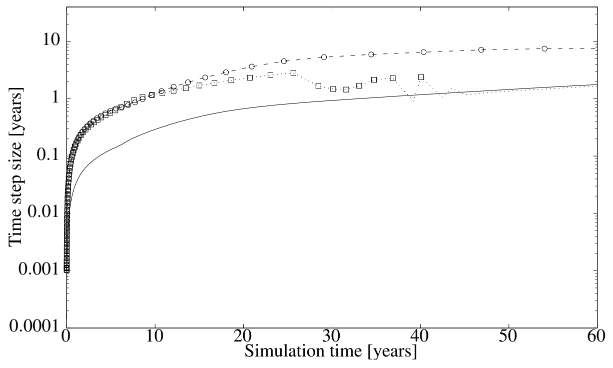
$$\Gamma_D^{TOP} = \{(x_1, x_2, x_3)^T \in \partial\Omega \mid 150 \leq x_1, x_2 \leq 450, x_3 = 0\}$$

with values given by $\omega(x_1, x_2, x_3) := \omega_0\phi(x_1)\phi(x_2)$ where

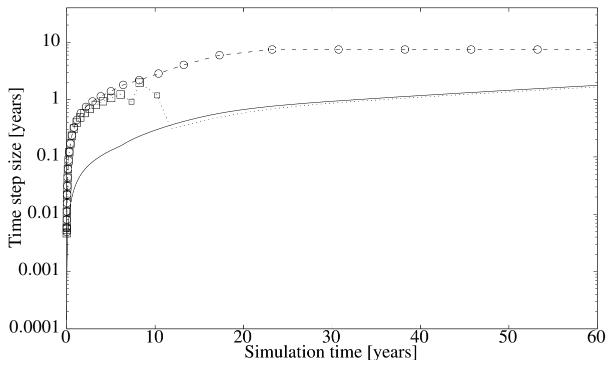
$$\phi(x) := \frac{9}{10} + \frac{1}{10}\cos\left(5\pi\frac{x - 300}{150}\right).$$



(a) $q_{max} = 2$

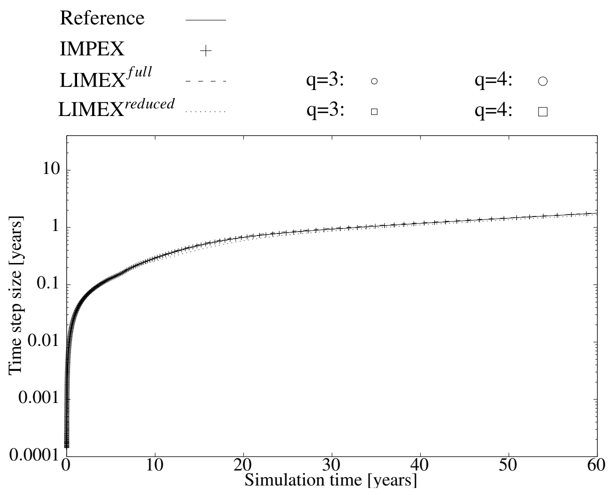


(b) $q_{max} = 3$

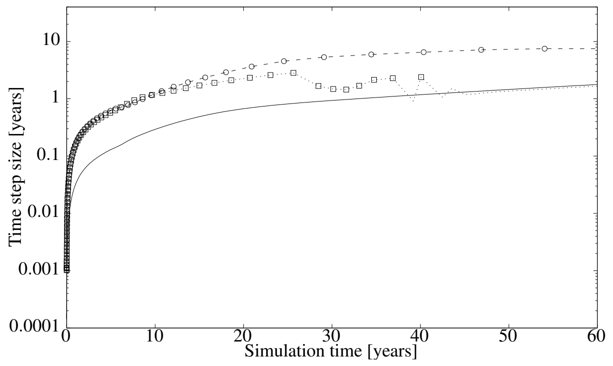


(c) $q_{max} = 4$

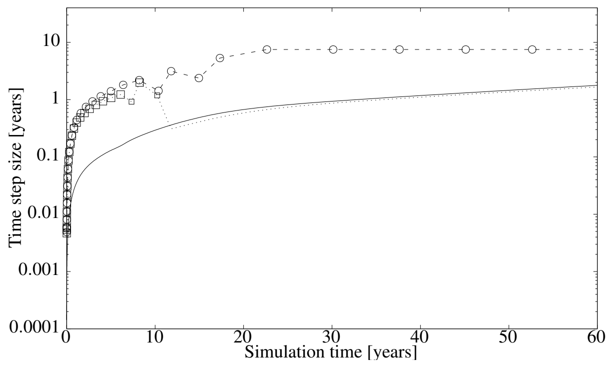
Figure 1: *Diffusion from salt dome (2D) with SuperLU preconditioner:* Time step size [yrs] vs. simulated time [yrs] for IMPEX, $\text{LIMEX}^{\text{full}}_{q_{max}}$, and $\text{LIMEX}^{\text{reduced}}_{q_{max}}$. The reference line is a continuous representation of IMPEX. It is identical in all figures to provide a guide to the eye. For LIMEX, the dashed lines represent steps with order $q = 2$, whereas symbols with different sizes indicate steps with order $2 < q \leq q_{max}$.



(a) $q_{max} = 2$



(b) $q_{max} = 3$



(c) $q_{max} = 4$

Figure 2: *Diffusion from salt dome (2D) with multigrid preconditioner*: Time step size [yrs] vs. simulated time [yrs] for IMPEX, $\mathsf{LIMEX}^{\text{full}}_{q_{max}}$, and $\mathsf{LIMEX}^{\text{reduced}}_{q_{max}}$. The reference line is a continuous representation of IMPEX. It is identical in all figures to provide a guide to the eye. For LIMEX, the dashed lines represent steps with order $q = 2$, whereas symbols with different sizes indicate steps with order $2 < q \leq q_{max}$.

Method	Time steps				Linear solver		Timing	
	q=2	q=3	q=4	reject	success	fail	CPU [s]	Effort
IMPEX	322	—	—	2	2094	0	18977.11	1.00
$\mathsf{LIMEX}_2^{\text{full}}$	311	—	—	2	941	0	8798.40	0.46
$\mathsf{LIMEX}_3^{\text{full}}$	0	68	—	4	434	0	4009.62	0.21
$\mathsf{LIMEX}_4^{\text{full}}$	0	0	33	5	382	0	3500.73	0.18
$\mathsf{LIMEX}_2^{\text{reduced}}$	320	—	—	2	968	0	9275.06	0.49
$\mathsf{LIMEX}_3^{\text{reduced}}$	22	70	—	10	548	0	5003.89	0.26
$\mathsf{LIMEX}_4^{\text{reduced}}$	68	2	24	7	528	0	4888.33	0.26

(a) SuperLU

Method	Time steps				Linear solver		Timing	
	q=2	q=3	q=4	reject	success	fail	CPU [s]	Effort
IMPEX	322	—	—	2	2094	0	4937.60	1
$\mathsf{LIMEX}_2^{\text{full}}$	311	—	—	2	941	0	2657.97	0.54
$\mathsf{LIMEX}_3^{\text{full}}$	0	68	—	4	434	0	1138.44	0.23
$\mathsf{LIMEX}_4^{\text{full}}$	0	0	34	7	392	2	1020.37	0.21
$\mathsf{LIMEX}_2^{\text{reduced}}$	320	—	—	2	968	0	2649.58	0.54
$\mathsf{LIMEX}_3^{\text{reduced}}$	5	80	—	8	545	0	1416.93	0.29
$\mathsf{LIMEX}_4^{\text{reduced}}$	55	3	27	5	505	0	1323.16	0.27

(b) Multigrid

Table 1: *Diffusion from salt dome (2D)*: Iteration counts (for time stepping and linear solver), and CPU wall clock time when using (a) multigrid or (b) SuperLU as a preconditioner.

For the initial value $\omega \equiv 0$, this provides an array of initially 25 fingers that collapses rapidly. Due to this convection induced fingering the problem is more challenging than the previous problem. This will be observed in the convergence of the linear solver. The geometry is given by $64 \times 64 \times 16 = 65,536$ cubic elements, corresponding to $143,650$ degrees of freedom.

SuperLU preconditioner. The time stepping history shown in Fig. 3 illustrates a very smooth behavior for $\mathsf{LIMEX}^{\text{full}}$. An order reduction is only observed in the very last steps of the highest order method $\mathsf{LIMEX}_4^{\text{full}}$. This is also reflected by Tab. 2a. As in the previous results, $\mathsf{LIMEX}^{\text{reduced}}$selects smaller time steps and the order reduction is activated earlier. In this sense, the results are similar to those from the previous section in Fig. 1. Note that the SuperLU preconditioner did not obtain a solution for IMPEX within 30 days. Hence, the computation was interrupted at this point. In all plots, we use $\mathsf{LIMEX}_2^{\text{full}}$ as a reference to provide a guide to the eye.

Multigrid preconditioner. When switching to the multigrid preconditioner, the methods using an exact Jacobian (IMPEX and $\mathsf{LIMEX}^{\text{reduced}}$) suffer from a severe drawback. As is illustrated in Figs. 4 and Tab.2b both methods are now limited by lack of convergence convergence of the iterative method, which leads to reduction time step. Although $\mathsf{LIMEX}^{\text{reduced}}$tends to select smaller time steps, it essentially revocers the results obtained with the SuperLU solver. As a consequence, the method provides greater robustness in particular towards the end of the simulation.

With respect to the total timings in Tab. 2b, the LIMEX-variants outperform IMPEX and are essentially on par with each other. Considering the asymptotic behavior, however, $\mathsf{LIMEX}^{\text{reduced}}$should be preferred.

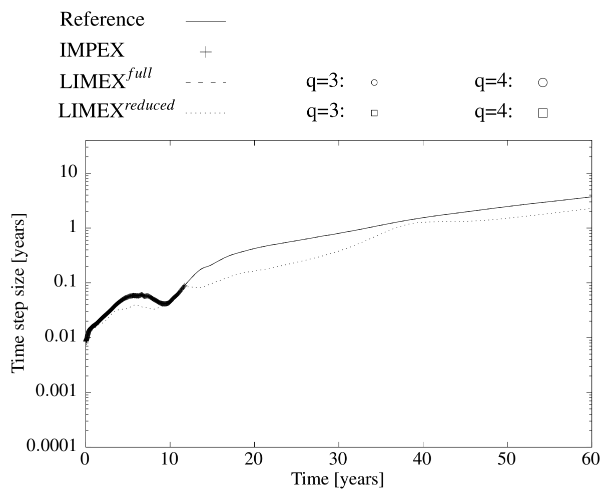
Method	Time steps				Linear solver		Timing
	q=2	q=3	$q = 4$	reject	success	fail	CPU [s]
IMPEX	(Aborted after 1 month.)						
$\mathsf{LIMEX}_2^{\text{full}}$	408	—	—	1	1230	0	1508625.38
$\mathsf{LIMEX}_3^{\text{full}}$	0	98	0	0	591	0	732320.94
$\mathsf{LIMEX}_4^{\text{full}}$	0	3	50	3	551	0	680860.07
$\mathsf{LIMEX}_2^{\text{reduced}}$	578	0	0	1	1742	0	2164447.66
$\mathsf{LIMEX}_3^{\text{reduced}}$	3	159	0	0	968	0	1212680.52
$\mathsf{LIMEX}_4^{\text{reduced}}$	35	2	63	4	785	0	981973.82

(a) SuperLU

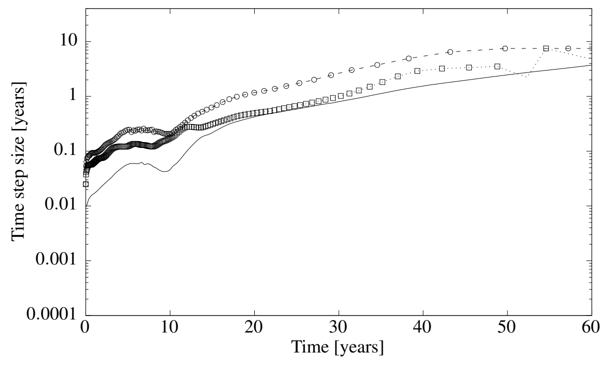
Method	Time steps				Linear solver		Timing	
	q=2	q=3	q=4	reject	success	fail	CPU [s]	Effort
IMPEX	461	—	—	37	3075	36	21228.19	1
$\mathsf{LIMEX}_2^{\text{full}}$	455	—	—	130	1371	129	13564.55	0.64
$\mathsf{LIMEX}_3^{\text{full}}$	75	90	—	131	768	131	7984.37	0.38
$\mathsf{LIMEX}_4^{\text{full}}$	81	7	40	136	708	134	7510.61	0.35
$\mathsf{LIMEX}_2^{\text{reduced}}$	578	—	—	1	1742	0	14375.54	0.68
$\mathsf{LIMEX}_3^{\text{reduced}}$	3	159	—	0	968	0	7464.05	0.35
$\mathsf{LIMEX}_4^{\text{reduced}}$	35	2	63	4	785	0	6115.75	0.29

(b) Multigrid

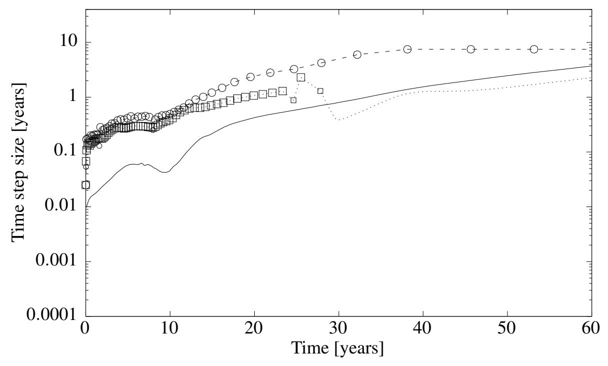
Table 2: *Modified Elder (3D)*: Iteration counts (for time stepping and linear solver), and CPU time when using (a) multigrid, or (b) SuperLU as preconditioner for the linear problems.



(a) $q_{max} = 2$

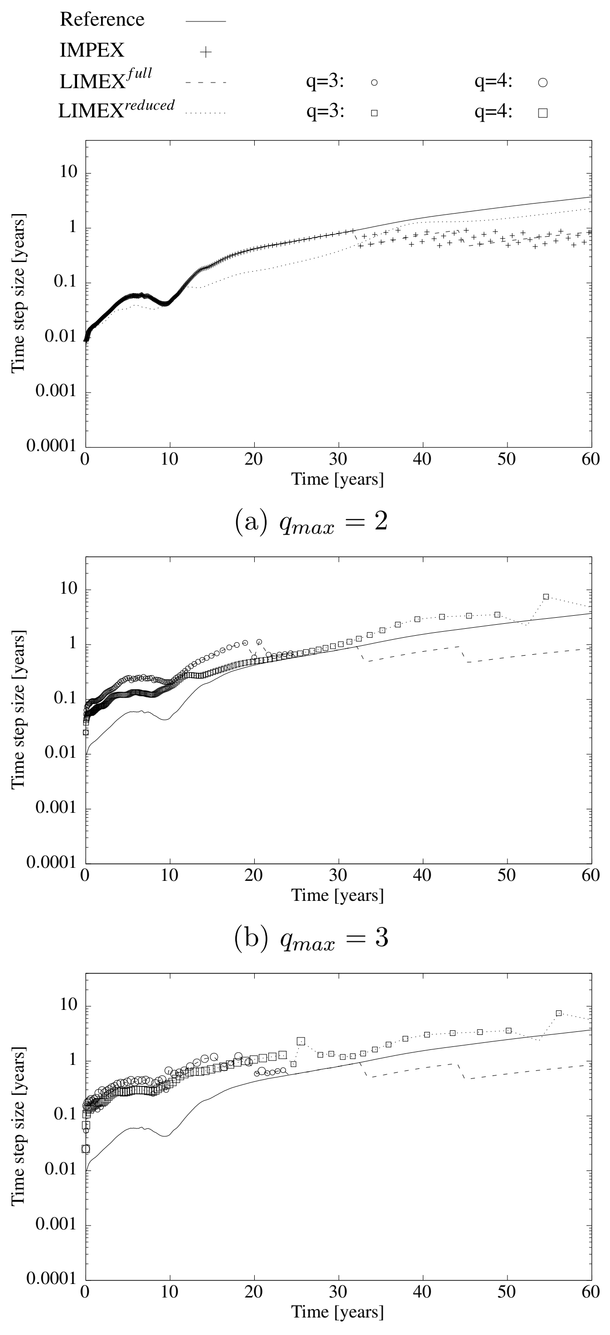


(b) $q_{max} = 3$



(c) $q_{max} = 4$

Figure 3: *Modified Elder (3D) with SuperLU*: Time step size [yrs] vs. simulated time [yrs] for IMPEX, $\mathsf{LIMEX}^{\text{full}}_{q_{max}}$, and $\mathsf{LIMEX}^{\text{reduced}}_{q_{max}}$. The reference line is a continuous representation of $\mathsf{LIMEX}^{\text{full}}_{2}$. It is identical in all figures to provide a guide to the eye. For LIMEX, the dashed lines represent steps with order $q = 2$, whereas symbols with different sizes indicate steps with order $2 < q \leq q_{max}$.



(a) $q_{max} = 2$

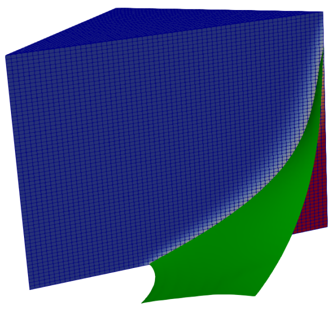
(b) $q_{max} = 3$

(c) $q_{max} = 4$

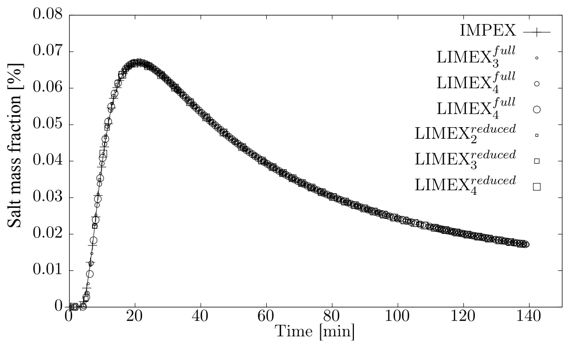
Figure 4: *Modified Elder (3D) with multi-grid*: Time step size [yrs] vs. simulated time [yrs] for IMPEX, $\mathsf{LIMEX}^{\text{full}}_{q_{max}}$, and $\mathsf{LIMEX}^{\text{reduced}}_{q_{max}}$. The reference line is a continuous representation of $\mathsf{LIMEX}^{\text{full}}_{2}$. It is identical in all figures to provide a guide to the eye. For LIMEX, the dashed lines represent steps with order $q = 2$, whereas symbols with different sizes indicate steps with order $2 < q \leq q_{max}$.

4.3. Saltpool Problem (3D)

The saltpool problem, suggested by Oswald and Kinzelbach [16], is an experimental benchmark for an upconing process. In this laboratory scale experiment, a cubic box conaining homogeneously shaped and distributed glass beads is filled with water. A stable brine layer with a given salt mass fraction is placed at the bottom. By injecting fresh water in one corner and simultanaeously extracting fluid in the opposite corner, an upconing of the brine is induced. Injection and extraction occur with a controlled rate and the breakthrough curves, i.e., the salinity of the extracted brine at the outlet, were monitored.



(a)



(b)

Figure 5: *Saltpool problem (3D):* Simulated results for (a) the solution at the end of experiment; (b) breakthrough curve at outlet (for all simulations).

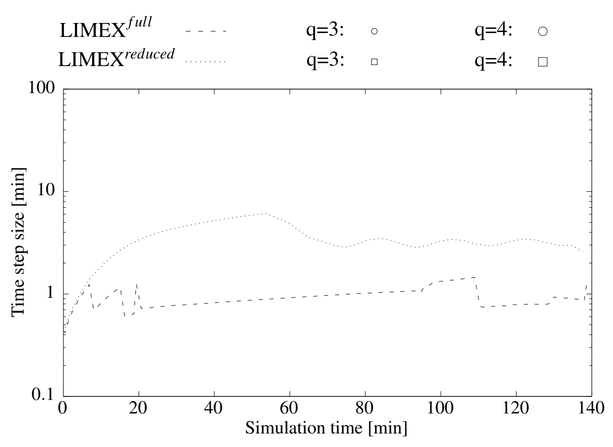
This problem imposes a challenge to time integration, as identified by Johannsen and coworkers [17]. In this work, we focus in particular on the low concentration case with only 1% salt mass fraction in the brine. In this case, the salinity at the outlet first rises rapidly to a peak $20min$, before the system relaxes towards the end of the experiment. Of course, these dynamics must be captured correctly. Figures 5a and 5b shows the salt distribution at the end of the experiments as well as the breakthrough curves computed by all methods.

Treatment of the dispersive terms. Unfortunately, the dispersion (3) leads to undesired properties of the discrete linear systems. As a consequence $\mathsf{LIMEX}^{\text{full}}$ is inefficient, as the multi-grid method is not converging properly with the consequence that time steps must be reduced. The same holds for $\mathsf{LIMEX}^{\text{reduced}}$, when only the derivatives w.r.t. convection are neglected. However, disabling derivatives w.r.t. dispersion as well mitigates this problem.

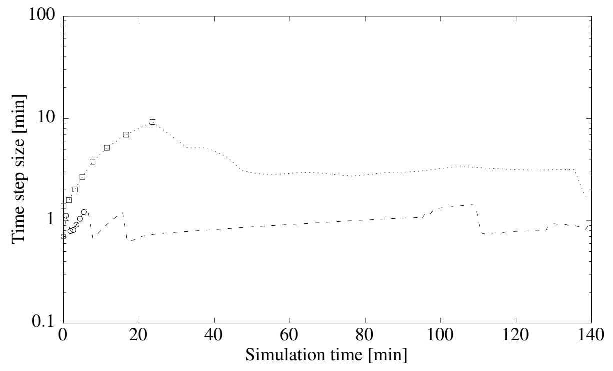
Iteration statistics and time-stepping history for a uniform refinement with 2,097,152 hexahedral fine grid elements (4,293,378 degrees of freedom) are shown in Tab. 3 and Fig. 6 respectively. Timings are for a parallel multigrid solver executed on 20 cores. Due to the problem size, the problem could not be solved by SuperLU. The IMPEX scheme failed after 38 steps (corresponding to 30 min simulated), with a time step contracting to zero. For $\mathsf{LIMEX}^{\text{full}}$, the time step is restricted by the linear solver. $\mathsf{LIMEX}^{\text{reduced}}$permits larger time steps. In all cases the nuber of stages is rapidly reduced to $q = 2$, however.

Method	Time steps				Linear solver		Timing
	q=2	q=3	$q = 4$	reject	success	fail	CPU [s]
IMPEX	(Failed after 38 steps)						—
$\mathsf{LIMEX}_2^{\text{full}}$	157	0	0	318	478	316	39135.5
$\mathsf{LIMEX}_3^{\text{full}}$	148	7	0	323	487	323	39924.5
$\mathsf{LIMEX}_4^{\text{full}}$	146	3	4	320	497	320	39877.0
$\mathsf{LIMEX}_2^{\text{reduced}}$	49	0	0	5	163	0	3674.7
$\mathsf{LIMEX}_3^{\text{reduced}}$	34	8	0	7	181	0	4028.6
$\mathsf{LIMEX}_4^{\text{reduced}}$	35	2	2	7	169	0	3757.5

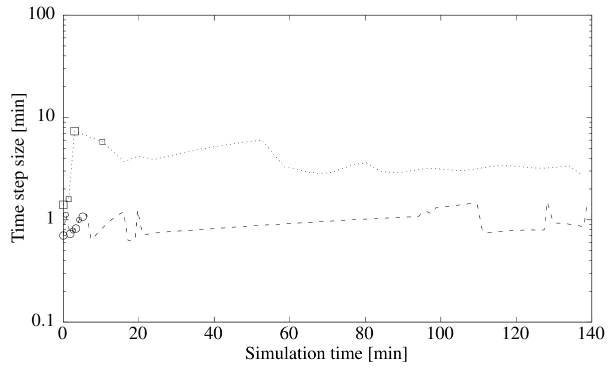
Table 3: *Saltpool problem (3D) with multigrid*: Iteration counts (for time stepping and linear solver), and CPU time.



(a) $q_{max} = 2$



(b) $q_{max} = 3$



(c) $q_{max} = 4$

Figure 6: *Saltpool problem (3D) with multi-grid*: Time step size [yrs] vs. simulated time [yrs] for $\mathsf{LIMEX}^{\text{full}}_{q_{max}}$ and $\mathsf{LIMEX}^{\text{reduced}}_{q_{max}}$. IMPEX failed for non-obvious reasons. The dashed lines represent steps with order $q = 2$, whereas symbols with different sizes indicate steps with order $2 < q \leq q_{max}$.

4.4. Gorleben problem (2D)

The last problem is taken from a field test first suggested in [20]. The task is to compute the ground water flow over a saltdome in the North German Plain, close to the village of Gorleben. The computational domain is a 2D cross section from geological data taken in north-south direction. It is given by an elongated strip with physical dimensions 16, 370 m in horizontal direction and 395 m in vertical direction. The domain including the coarse grid triangulation is shown in Fig. 7. Mesh statistics are provided in Tab. 4.

Refinement level	# Elements	# Degrees of freedom
Base	4,924	8,506
3	315,136	498,954
4	1, 260,544	1,982,866
5	5,042,176	7,905,570

Table 4: *Gorleben problem (2D):* Mesh statistics after uniform refinement.

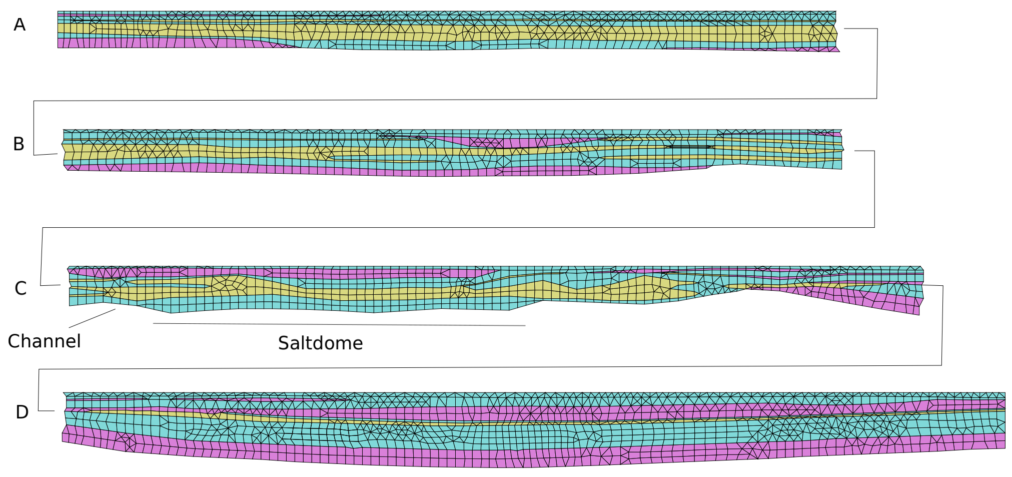


Figure 7: *Gorleben problem (2D):* Permeability distribution in the computational domain indicated by colors (cyan: high permeability; magenta: medium permeability; yellow: low permeability). For the ease of presentation, the domain is presented broken into sections A (horizontal extension: 0km – 4.4km) B (4.4km – 9.3km), C (9.3km – 14.3km), and D (14.3km – 19.8km). The saltdome is located under Section C as indicated.

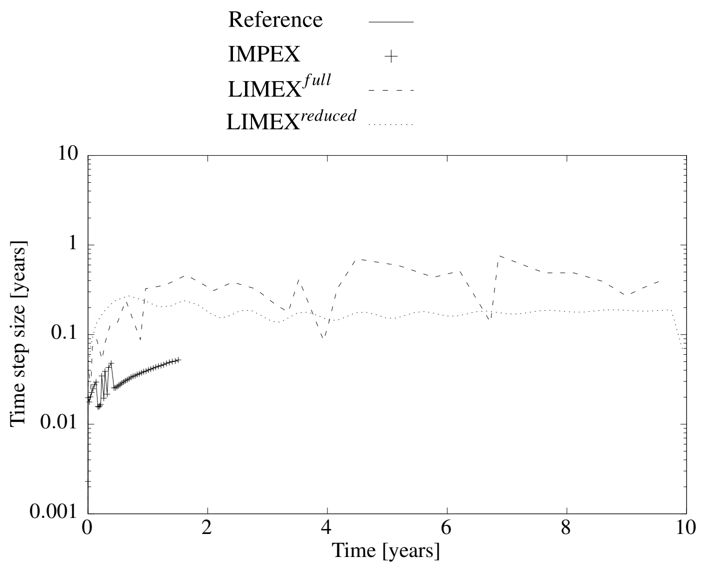
For the sake of simplification, the illustration is broken into four sections shown from top to bottom, that can be aligned from left to right. The shading of the elements indicates zones with three different permeabilities (cyan: high permeability, $\kappa = 10^{-12}\,\mathrm{m}^2$; magenta: medium permeability, $\kappa = 10^{-14}\,\mathrm{m}^2$; yellow: low permeability, $\kappa = 10^{-16}\,\mathrm{m}^2$). The saltdome is in contact with the domain in the middle of section C. In this region, a constant mass fraction $\omega = \omega_{max} = 0.26$ is assumed as a boundary condition. The boundary conditions on the upper part of the domain are given by the fresh water condition $\omega = 0$. Moreover, the pressure is given as a piecewise linear function derived from experiments. It starts from 173 kPa on the southernmost tip (left) and decreases to 100 kPa in the northernmost tip (right). We use the *linear density function* (4) (with $\rho_0 = 998.2\,\mathrm{kg/m}^3$, $\rho_1 = 1197.2\,\mathrm{kg/m}^3$ and $\omega_{max} = 0.26$), and the *real viscosity model* (5b) (with $\mu_0 = \times 10^{-3}\mathrm{kg\,m^{-1}s^{-1}}$). Diffusion and dispersion are characterized by porosity $\phi = 0.2$, molecular diffusion $D_m = 10^{-9}\,\mathrm{m}^2/\mathrm{s}$, dispersion lengths $\alpha_L = 10\mathrm{m}$, and $\alpha_T = 1\mathrm{m}$. Initally, no salt is present in the domain ($\omega = 0$).

The scientific interest for the problem is in particular the long term evolution. Based on the experience from the previous tests, in particular from the saltpool benchmark, we limit the number of stages to $q_{max} = 2$. This is consistent with the low regularity, e.g., since $\mathbb{D}(\vec{q})$ is not differentiable at $\vec{q} = 0$, and jumping permeability coefficients.

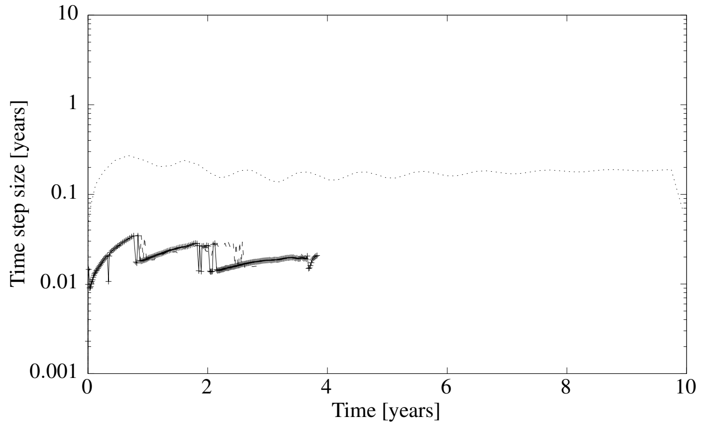
Performance of the different solvers. As a preliminary step, solver performance is investigated in the small time interval of the first 10 years only on the rather coarse mesh at refinement level 3. The linear system for the corresponding $498,954$ degrees of freedom is still small enough for the (serial) SuperLU solver. Results are provided in Fig. 8 and Tab. 5. After 12h of wall clock time unfinished jobs were terminated by the scheduler.

For this problem, IMPEX showed unexpected behavior, since the the method did not suceed to compute the solution at $t = 10\mathrm{h}$ within the wall-clock time limit of twelve hours. For large time steps, only linear convergence of the Newton method was observed, and the desired tolerance was not achieved within the prescribed maximum number of $n = 10$ iterations. As a result, the time stap had to be recomputed with bisected step size. At the end, the computational cost for this strategy turned out to be too high.

LIMEX$^{\mathrm{full}}$succesfully computed a solution using the SuperLU preconditioner. However, when switching to multigrid, linear solver turned out to be the limiting factor. Again, this could be avoided using LIMEX$^{\mathrm{reduced}}$,



(a) SuperLU



(b) Multi-grid

Figure 8: *Gorleben problem (2D) – initial phase:* Time stepping history for the first 10 years with both preconditioners. Due to the interest in long term behavior, results are provided for $q_{max} = 2$ stages only. Wall clock time was restricted to 12h. IMPEX (both preconditioners) and LIMEX$^{\text{full}}$ (with multi-grid) did not finish within this limit. For these instances, data is provided until termination.

Method	Time steps				Linear solver		Timing
	q=2	q=3	$q=4$	reject	success	fail	CPU [s]
IMPEX	Did not terminate within 12h						
$\mathsf{LIMEX}_2^{\text{full}}$	36	0	0	3	122	0	2447.12
$\mathsf{LIMEX}_2^{\text{reduced}}$	60	0	0	0	185	0	3766.70

(a) SuperLU

Method	Time steps				Linear solver		Timing
	q=2	q=3	$q=4$	reject	success	fail	CPU [s]
IMPEX	Did not terminate within 12h						
$\mathsf{LIMEX}_2^{\text{full}}$	Did not terminate within 12h						
$\mathsf{LIMEX}_2^{\text{reduced}}$	60	0	0	0	185	0	2250.10

(b) Multigrid

Table 5: *Gorleben problem (2D) – initial phase*: Iteration counts (for time stepping and linear solver), and CPU time when using (a) multigrid, or (b) SuperLU as preconditioner for the linear problems.

which generated identical time stepping histories indepdendent of the preconditioner. In passing we note that even for this small problem size, using multigrid resulted in smaller wall clock times.

Computational results. Results computed with $\mathsf{LIMEX}_2^{\text{reduced}}$ over 10.000 years are shown in Fig. 9. The illustration is centered around region 3 above the salt dome. The grid has been obtained by 4 uniform refined steps, which is accurate enough for our purposes. The brine distribution far away from the salt dome is almost independent of the grid size h (data not shown). The pressure distribution at the top surface induces a flow from left to right, which is observed in the highly conductive layers close to the surface. The solution was obtained within $\approx$ 12h wall clock time on HazelHen using 192 cores in 9885 time steps (and corresponding 29656 calls of a parallel multgrid solver).

Mesh size dependent time stepping. In the deeper strata in proximity of the saltdome a bidirectional flow pattern is observed: In the upper region of these strata, fresh water flows towards the salt dome, while brine is transported away from the saltdome at the bottom of the strata. Both regions are well

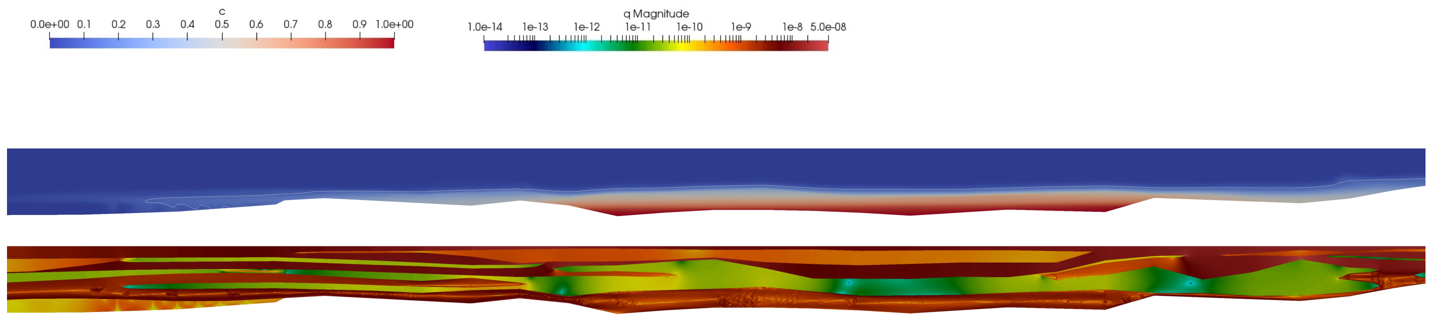
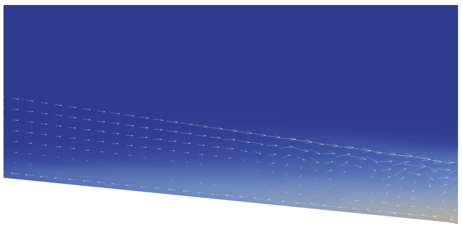
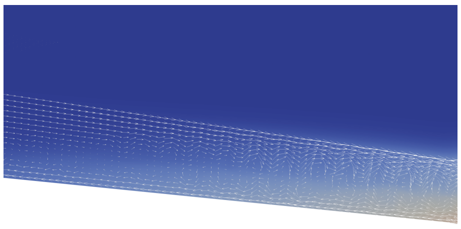
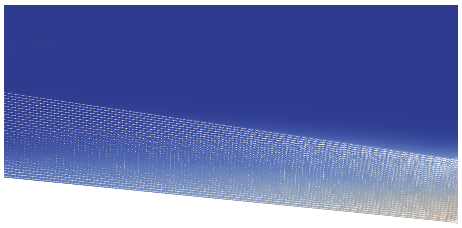


Figure 9: *Gorleben problem (2D):* Results at $t = 10,000$ years. *Top:* Salt mass distribution. The white line indicates the isoline for $\omega = 0.1$. *Bottom:* Magnitude of $\vec{q}$. All results are at refinement level 4.


(a) Refinement level 3


(b) Refinement level 4


(c) Refinement level 5

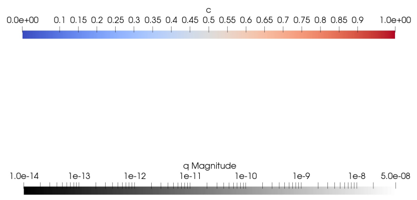


Figure 10: *Gorleben problem*: Magnification of the bidirectional flow pattern in the channel region in section 3 left of the salt dome ($t \approx 1,000$ years; refinement levels 3-5): Brine with high density is removed from the saltdome on the bottom, whereas fluid with lower density is transported towards the salt dome in the upper region of the channel. This gives rise to convective cells. Their size depends on the mesh size.

separated by a region of small velocity, giving rise to transient convective cells. This depends on the mesh size h is visualized in Fig. 10 for $t = 1,000$ years in a magnified region at the southern (left) end of the aquifer above the salt dome. The mesh size dependence is also reflected in the corresponding LIMEX time stepping history in Fig 11.

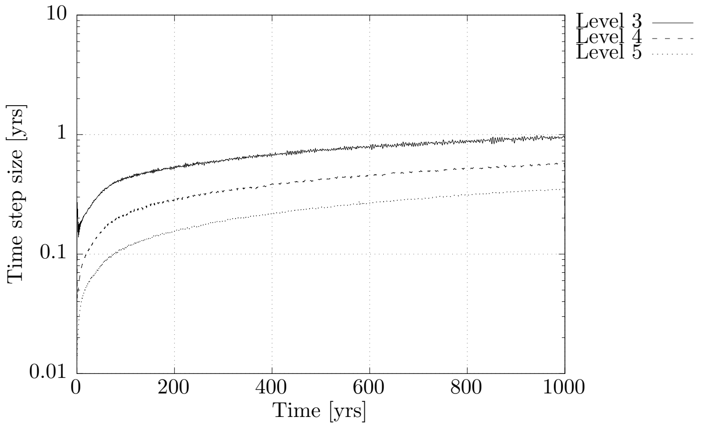


Figure 11: *Gorleben problem (2D)*: Time step size [yrs] vs. time [yrs] for $\mathsf{LIMEX}_2^{\text{reduced}}$ (omitting derivatives w.r.t. convection and diffusion).

5. Conclusion

For density-driven flow problems LIMEX-methods have been demonstrated as an attractive and efficient alternative to standard time-stepping strategy. To that end, we devised two versions, differing in the approximation of the Jacobian:

In principle, the basic version $\mathsf{LIMEX}^{\text{full}}$ with an exact Jacobian is the method of choice. It is the most efficient method w.r.t. time stepping. In some cases, however, convergence of the linear multi-grid solver may be poor, and thus total efficiency is deterioriating.

When this happens, the version $\mathsf{LIMEX}^{\text{reduced}}$ may be an attractive alternative. This is constructed based on an inexact Jacobian. As demonstrated theoretically, neglecting certain terms of the Jacobian, leads to a linear system, where the variables ω and p decouple. This can be achieved by treating the partial derivatives of Darcy velocity $\vec{q}$ in the transport equation explicitly. Even though the time steps turn out to be smaller than with $\mathsf{LIMEX}^{\text{full}}$, it was demonstrated that convergence problems of the linear solver can be avoided. With $\mathsf{LIMEX}^{\text{reduced}}$, the field test for the *Gorleben* problem could be solved robustly.

Acknowledgements

We would like to thank Anne Gehrke (GRS Braunschweig) for providing the specification of the Gorleben problem in Section 4.4 and its initial triangulation. The work of the first and third author was supported by the German *Bundesministerium für Wirtschaft und Energie* in the grant *Grundwasserströmung und Stofftransport in komplexen realen Systemen (FKZ 02E11476)*. The second author is grateful to Jens Lang for an intensive discussion on order reduction in nonlinear parabolic equations. All authors want to thank Beijing Institute of Scientific and Engineering Computing (BISEC), Beijing University of Technology (BJUT), for its kind hospitality while working on this paper. We acknowledge HLRS (Stuttgart, Germany) for providing access and computational resources at the Hazel Hen supercomputer.

References

[1] A. Nägel, A. Vogel, G. Wittum, Evaluating linear and nonlinear solvers for density driven flow, Computer Methods in Applied Mechanics and Engineering 292 (2015) 3–15.

[2] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Springer Heidelberg Dordrecht London New York, 2002.

[3] P. Deuflhard, F. Bornemann, Scientific Computing with Ordinary Differential Equations, Springer, 2002.

[4] V. John, J. Rang, Adaptive time step control for the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 199 (2010) 514–524.

[5] P. Deuflhard, U. Nowak, Extrapolation integrators for quasilinear implicit ODEs, in: P. Deuflhard, B. Engquist (Eds.), Large scale scientific computing, Birkhaeuser, 1987, pp. 37–50.

[6] C. Lubich, A. Ostermann, Linearly implicit time discretization of nonlinear parabolic equations, IMA J. Numer. Anal. 15 (1995) 555–583.

[7] P. Deuflhard, M. Weiser, Adaptive Numerical Solution of PDEs, de Gruyter, 2012.

[8] K. Gustafsson, M. Lundh, G. Söderlind, Api stepsize control for the numerical solution of ordinary differential equations, BIT Numerical Mathematics 28 (2) (1988) 270–287.

[9] J. Lang, Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems - Theory, Algorithm, and Applications, Springer, 2001.

[10] A. Vogel, S. Reiter, M. Rupp, A. Nägel, G. Wittum, UG 4: A novel flexible software system for simulating PDE based models on high performance computers, Computat. and Visualiz. in Science 16 (4) (2013) 165–179.

[11] S. Reiter, A. Vogel, I. Heppner, M. Rupp, G. Wittum, A massively parallel geometric multigrid solver on hierarchically distributed grids, Computat. and Visualiz. in Science 16 (4) (2013) 151–164.

[12] P. Bastian, G. Wittum, On robust and adaptive multi-grid methods, in: Hemker, P. W.; Wesseling, P. (Ed.), Multigrid methods IV: Proceedings of the Fourth European Multigrid, Vol. 116 of International series of numerical mathematics, Birkhäuser, Basel, 1994, pp. 1–17.

[13] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, J. W. H. Liu, A supernodal approach to sparse partial pivoting, SIAM J. Matrix Analysis and Applications 20 (3) (1999) 720–755.

[14] H. van der Vorst, Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (2) (1992) 631–644. `doi:10.1137/0913035`.

[15] C. I. Voss, W. R. Souza, Variable Density Flow and Solute Transport Simulation of Regional Aquifers Containing a Narrow Freshwater-Saltwater Transition Zone, Water Resources Research 23 (10) (1987) 1851–1866.

[16] S. E. Oswald, W. Kinzelbach, Three-dimensional physical benchmark experiments to test variable-density flow models, Journal of Hydrology 290 (2004) 290 (2004) 22–42.

[17] K. Johannsen, W. Kinzelbach, S. Oswald, G. Wittum, The saltpool benchmark problem – numerical simulation of saltwater upconing in a porous medium, Advances in Water Resources 25 (2002) 335–348.

[18] K. Johannsen, On the validity of the Boussinesq approximation for the Elder problem, Computational Geosciences 7 (2003) 169–182.

[19] C. Schwarz, Dichteabhängige Strömungen in homogenen und heterogenen porösen Medien, Ph.D. thesis, Eidgenössische Technische Hochschule Zürich (1999).

[20] S. Keesmann, U. Noseck, D. Buhmann, E. Fein, A. Schneider, Modellrechnungen zur Langzeitsicherheit von Endlagern in Salz- und Granitformationen, Tech. rep., Gesellschaft für Anlagen- und Reaktorsicherheit (2005).