Computational Geosciences (2021) 25:1667–1679
https://doi.org/10.1007/s10596-021-10067-w

ORIGINAL PAPER

Exponential time differencing for problems without natural stiffness separation

Nutchapol Dendumrongsup[1] · Daniel M. Tartakovsky[1]

Received: 26 April 2020 / Accepted: 3 June 2021 / Published online: 26 June 2021

Abstract

Explicit numerical schemes are popular in multiphysics and multiscale simulations, yet their use in stiff problems often requires time steps to be so small as to render simulations over large time horizons infeasible. Exponential time differencing (ETD) has proved to be an efficient scheme for tackling differential operators with linear stiff and nonlinear non-stiff parts. Such natural separation, however, is absent in many important applications, including multiphase flow and transport in porous media. We introduce a strategy for using ETD in such problems and demonstrate its efficiency in numerical experiments. We also compare the ETD performance to that of an explicit scheme. We conclude that the best outcome is achieved by combining ETD with a fourth-order Runge-Kutta method. Although our methodology is demonstrated on two-dimensional multiphase flow in porous media, it is equally applicable to other applications described by parabolic differential equations of this kind.

Keywords Stiff · ETD · Explicit · Multiphase · Nonlinear

1 Introduction

While implicit numerical methods have a number of theoretical advantages, many multiphysics and multiscale simulations find it necessary to resort to explicit schemes [1]. The latter are straightforward to implement, but are only conditionally stable, placing a constraint on the time-step size. For stiff problems, such as multiphase flow and solute transport in porous media, the required step size can become so small as to render numerical simulations over a time horizon of interest prohibitively expensive.

Several numerical schemes for stiff problems have been proposed [2], some of which found their way into computational geosciences (see, e.g., [3–5] for some of the more recent examples). Among these, exponential time differencing (ETD) [6, 7] has attracted a lot of attention as an efficient approach to dealing with stiffness [8–11].

In its standard implementation, ETD leverages an empirical observation that the stiffness of an equation arrises from its linear elliptic part (e.g., the Laplace operator representing diffusion, dispersion, viscous forces). That precludes the direct application of ETD to multiphase flow problems, including Richards' equation for flow in partially saturated porous media [12], in which the elliptic term is highly nonlinear due to dependence of its coefficient on the state variable (e.g., of relative permeability/conductivity on fluid saturation).

A major goal of this study is to explore numerical strategies for adopting ETD to problems, whose differential operators do not possess linear stiff components. Our strategy relies on defining the stiff linear operator via linearization of its nonlinear stiff counterpart around a reference value of the state variable. Conceptually similar ideas of stiffness separation have been explored in combination with other time-integration schemes [13, 14]. We are unaware of the use of this strategy in the context of ETD.

In lieu of theoretical analysis, we conduct a series of numerical experiments to ascertain the performance of our ETD strategy for problems without natural stiffness separation, in terms of both accuracy and computational

This research was supported in part by Air Force Office of Scientific Research under award number FA9550-18-1-0474; by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001202; and by a gift from Total.

✉ Daniel M. Tartakovsky
tartakovsky@stanford.edu

[1] Department of Energy Resources Engineering, Stanford University, 367 Panama St., Stanford, CA 94305, USA

efficiency. Multiphase flow equations, a coupled system of highly nonlinear partial differential equations of parabolic type, serve as our computational testbed. Following the standard practice in the ETD literature (e.g., [7]), we compare the performance of ETD with that of the standard explicit (Runge–Kutta) and implicit (Newton–Raphson) schemes. A detailed comparison with more advanced techniques such as implicit/explicit (IMEX) methods [15], which are tailor-made for multiphase flow equations, is left for further studies. We do compare our method with the implicit pressure, explicit saturation (IMPES) finite volume approach [16].

The standard ETD method and its modification for problems without stiffness separation are presented in Section 2. This modified ETD is used to solve two-phase flow equations, which are formulated in Section 3. Alternative numerical implementations of our ETD approach are presented in Section 4. Simulation results comparing the performance of ETD with that of implicit and explicit solvers are presented in Section 5. Main conclusions drawn from this analysis are summarized in Section 6.

2 Exponential time differencing

Consider a d-dimensional simulation domain $\Omega \subset \mathbb{R}^d$ and a simulation time horizon $[0, T] \subset \mathbb{R}^+$. Dynamics of a state variable $u(\mathbf{x}, t) : \Omega \times [0, T] \to \mathbb{R}^+$ defined on this domain is described by a nonlinear partial-differential equation (PDE)

$$\frac{\partial u}{\partial t} = \mathcal{M}(u, \mathbf{x}, t), \qquad (\mathbf{x}, t) \in \Omega \times (0, T], \tag{1}$$

subject to appropriate initial and boundary conditions. Suppose, first, that the nonlinear differential operator

$$\mathcal{M}(u, \cdot) = \mathcal{L}u + \mathcal{N}(u, \cdot) \tag{2}$$

can be decomposed into linear and nonlinear parts, $\mathcal{L}$ and $\mathcal{N}$, respectively. If the simulation domain Ω is discretized into N_{dis} elements or nodes, then a suitable discretization of the differential operators $\mathcal{L}$ and $\mathcal{N}$ yields a system of ordinary differential equations (ODEs)

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{L}\mathbf{u} + \mathbf{N}(\mathbf{u}, \mathbf{x}, t), \qquad \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0 \tag{3}$$

where $\mathbf{u}$ is the vector of unknowns of length N_{dis}, $\mathbf{L}$ is an $N_{\text{dis}} \times N_{\text{dis}}$ matrix, and $\mathbf{N}$ is a vector function with N_{dis} components. Consider a set of N_{eig} eigenvalues λ_i of the matrix $\mathbf{L}$, with $i = 1, \ldots, N_{\text{eig}}$. The system of ODEs (3) is said to be stiff if the ratio

$$s = \frac{\max_i\{|\Re(\lambda_i)|\}}{\min_i\{|\Re(\lambda_i)|\}},$$

which is referred to as stiffness, is large.

Consider the diagonalization of $\mathbf{L}$ such that $\mathbf{L} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, where $\mathbf{P}$ is an invertible matrix and

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_{N_{\text{dis}}} \end{bmatrix}. \tag{4}$$

Exponential of the matrix $\mathbf{L}$ is defined by

$$\begin{aligned} \mathrm{e}^{\mathbf{L}} &= \mathrm{e}^{\mathbf{P}\mathbf{D}\mathbf{P}^{-1}} = \sum_{k=0}^{\infty} \frac{(\mathbf{P}\mathbf{D}\mathbf{P}^{-1})^k}{k!} = \sum_{k=0}^{\infty} \frac{\mathbf{P}\mathbf{D}^k\mathbf{P}^{-1}}{k!} \\ &= \mathbf{P}\left(\sum_{k=0}^{\infty} \frac{\mathbf{D}^k}{k!}\right)\mathbf{P}^{-1} = \mathbf{P}\mathrm{e}^{\mathbf{D}}\mathbf{P}^{-1}, \end{aligned} \tag{5}$$

where $\exp(\mathbf{D})$ is the diagonal matrix whose non-zero components are $\exp(D_{ii}) = \exp(\lambda_i)$ with $i = 1, \ldots, N_{\text{dis}}$. Multiplying (3) by the integrating factor $\exp(-\mathbf{L}t)$ and integrating the resulting equation over a single time step τ, from t_n to $t_{n+1} = t_n + \tau$, one obtains

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathrm{e}^{\mathbf{L}\tau}\mathbf{u}_n \\ &\quad + \mathrm{e}^{\mathbf{L}\tau} \int_0^{\tau} \mathrm{e}^{-\mathbf{L}s}\mathbf{N}(\mathbf{u}(t_n + s), \mathbf{x}, t_n + s)\mathrm{d}s, \end{aligned} \tag{6}$$

with $\mathbf{u}_n \equiv \mathbf{u}(t_n)$. Formula (6) is exact; different approximations of the integral term give rise to alternative ETD schemes [7].

This study deals with problems in which the differential operator $\mathcal{M}$ in Eq. 1 lacks a *linear* stiff component $\mathcal{L}$, i.e., cannot be decomposed according to Eq. 2. A ubiquitous example of this kind is a nonlinear diffusion equation, $u_t = \mathcal{M}(u)$, whose elliptic operator $\mathcal{M}(u) = \nabla \cdot [D(u)\nabla u]$ involves the diffusion coefficient $D(u)$ that is an invertible function of the concentration $u(\mathbf{x}, t)$. Our approach is to i) replace this nonlinear stiff term with its linear counterpart $\mathcal{L}u$, e.g., replace $\mathcal{M}(u)$ with its linearized counterpart $\mathcal{L}u = \nabla \cdot [D(\tilde{u})\nabla u]$ where $\tilde{u}$ can be the initial state u_{in}, the maximal or average value of u, etc.; ii) define the nonlinear term $\mathcal{N}(u) = \mathcal{M}(u) - \mathcal{L}u$; and iii) use ETD in Eq. 6 to solve the ODEs resulting from the spatial discretization of these differential operators.

Selection of the linear stiff terms $\mathcal{L}$ or, equivalently, the corresponding matrix $\mathbf{L}$ requires careful consideration. If too much stiffness is delegated to the nonlinear term $\mathcal{N}$ (or $\mathbf{N}$), then the benefit of using ETD, i.e., the ability to solve the large linear stiff part analytically, is quickly lost, as shown in the numerical experiments reported below.

3 Two-phase flow in porous media

Our computational testbed deals with two-dimensional ($d = 2$) flow of two immiscible compressible fluids (e.g., oil and gas) of viscosity μ_l, density ρ_l, and compressibility B_l (l = oil, gas) in a homogeneous anisotropic porous medium of porosity ϕ and intrinsic permeability $\boldsymbol{\kappa}$. Under isothermal conditions, the flow is described by equations of conservation of mass and momentum for each phase l,

$$\phi \frac{\partial}{\partial t}\left(\frac{S_l}{B_l}\right) + \nabla \cdot \left(\frac{\mathbf{U}_l}{B_l}\right) = q_l, \quad l = \text{gas, oil} \tag{7a}$$

$$\mathbf{U}_l = -\boldsymbol{\kappa}\frac{\kappa_{\text{rel},l}}{\mu_l \rho_l}(\nabla p_l - g), \quad l = \text{gas, oil.} \tag{7b}$$

Here, S_l is the medium's saturation with phase l, such that $S_{\text{gas}} + S_{\text{oil}} = 1$; $\mathbf{U}_l$ and p_l are Darcy velocity of and pressure in the lth phase, respectively; and q_l denotes the source/sink with the convention that $q_l > 0$ for injection, and $q_l < 0$ for production; and g is the gravitational acceleration constant. The saturation-dependent relative permeabilities of the two phases, $\kappa_{\text{rel},l}$, are defined by, e.g., [17]

$$\kappa_{\text{rel, gas}} = S_{\text{gas}}^2, \qquad \kappa_{\text{rel, oil}} = (1 - S_{\text{gas}})^{1.5}. \tag{7c}$$

We neglect gravity and capillary forces, i.e., assume $p_{\text{oil}} = p_{\text{gas}} \equiv p$. The compressibility of oil and water satisfies constitutive laws

$$B_{\text{gas}} = \mathrm{e}^{c_{\text{gas}} \Delta_p}, \tag{7d}$$

$$B_{\text{oil}} = \begin{cases} \mathrm{e}^{-\alpha_{\text{oil}} \Delta_p} & \text{if } p < p_{\text{bub}} \\ \mathrm{e}^{-\alpha_{\text{oil}} \Delta_p - c_{\text{oil}}(p - p_{\text{bub}})} & \text{otherwise} \end{cases} \tag{7e}$$

where $c_{\text{gas}} = 1.7 \cdot 10^{-3}$ psi^{-1}; $\alpha_{\text{oil}} = 8.0 \cdot 10^{-5}$ psi^{-1}; $c_{\text{oil}} = 8.0 \cdot 10^{-6}$ psi^{-1}; p_{atm} and p_{bub} are the atmospheric and bubbling pressure (in psi), respectively; and

$$\Delta_p = \begin{cases} p_{\text{atm}} - p & \text{if } p < p_{\text{bub}} \\ p_{\text{atm}} - p_{\text{bub}} & \text{otherwise.} \end{cases}$$

Among the plethora of methods for solving (7), we took an inspiration from an implicit pressure, explicit saturation (IMPES) finite volume approach [16]. To simplify the presentation, we use a square simulation domain that is discretized with an $N \times N$ grid, whose elements are indexed alternatively either by (i, j) with $1 \leq i, j \leq N$ or by $1 \leq k \leq N^2$. Pressure in the oil phase and gas-phase saturation are arranged into vectors of length N^2,

$$\mathbf{p} = \begin{bmatrix} \vdots \\ p_{\text{oil}}^k \\ \vdots \end{bmatrix}, \qquad \mathbf{S} = \begin{bmatrix} \vdots \\ S_{\text{gas}}^k \\ \vdots \end{bmatrix}, k = 1, \ldots, N^2. \tag{8}$$

In the traditional IMPES method, Eq. 7 is formulated as (see the Appendix for details)

$$\check{\mathbf{D}}(\mathbf{p}^{n+1}, \mathbf{S}^n)(\mathbf{p}^{n+1} - \mathbf{p}^n) = \check{\mathbf{T}}(\mathbf{p}^{n+1}, \mathbf{S}^n)\mathbf{p}^{n+1} - \check{\mathbf{Q}}(\mathbf{p}^{n+1}, \mathbf{S}^n). \tag{9}$$

Its significant computational cost comes from the implicit solver.

Instead, we modify this method in a way that enables us to take advantage of the explicit ETD scheme:

$$\mathbf{D}(\mathbf{u})\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{T}(\mathbf{u})\mathbf{u} - \mathbf{Q}(\mathbf{u}), \tag{10}$$

where $\mathbf{u} \equiv \mathbf{p}$; the dependence on $\mathbf{S}^n$, which is known from the previous time step, is suppressed to simplify the notation; $\mathbf{T}$ is the $N^2 \times N^2$ penta-diagonal transmissibility matrix; $\mathbf{Q}$ denotes a flow-rate vector of length N^2; and $\mathbf{D}$ is the product of the $N^2 \times N^2$ diagonal compressibility matrix and the time step size. These matrices are defined, respectively, by Eqs. 48, 49, and 50 in the Appendix.

This system of ODEs does not lend itself to a direct ETD treatment because it lacks a linear stiff part. Instead, we rewrite it first as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{D}^{-1}\mathbf{T}\mathbf{u} - \mathbf{D}^{-1}\mathbf{Q}, \tag{11}$$

and then as Eq. 3 with

$$\mathbf{L} = \mathbf{D}^{-1}(\tilde{\mathbf{u}})\mathbf{T}(\tilde{\mathbf{u}}) \tag{12a}$$

and

$$\mathbf{N} = [\mathbf{D}^{-1}\mathbf{T} - \mathbf{L}]\mathbf{u} - \mathbf{D}^{-1}\mathbf{Q}. \tag{12b}$$

Alternative strategies for choosing the state $\tilde{\mathbf{u}}(t)$, which is used to construct the linear operator $\mathbf{L}$, are discussed in Section 5. The nonlinear operator $\mathbf{N}$ retains some of the stiffness of the original equation; specifically, the matrix $\mathbf{D}^{-1}\mathbf{T}$ is not invertible. That is because, by construction, the summation of any column of the transmissibility matrix $\mathbf{T}$ is a zero vector. Thus, the transmissibility matrix $\mathbf{T}$ is singular. Once the pressure $\mathbf{u}^{n+1}$ is computed, the saturation is updated explicitly, in accordance with Eq. 46, to obtain $\mathbf{S}^{n+1}$.

The formulation (11) and (12) is beneficial if one can identify $\mathbf{L}$ that captures rapid changes in the solution, as illustrated in Section 5.1.

Remark 1 In $d = 3$ spatial dimensions, the matrices in Eqs. 8–12 are modified as follows. If a three-dimensional simulation domain were discretized with a $N \times N \times N$ grid, then the vectors $\mathbf{p}$ and $\mathbf{S}$ would have length N^3; $\mathbf{T}$ would be the $N^3 \times N^3$ hepta-diagonal transmissibility matrix; the flow-rate vector $\mathbf{Q}$ would have length N^3; and $\mathbf{D}$ would be the product of the $N^3 \times N^3$ diagonal compressibility matrix and the time step size.

Remark 2 Construction of the matrices in Eqs. 8–12 can be readily generalized to account for different numbers of elements in each spatial dimension. If the number of elements in the ith direction is N_i $(i = 1, \ldots, d)$, then the length of the vectors $\mathbf{p}$ and $\mathbf{S}$ is $\prod_{i=1}^{d} N_i$, etc.

Remark 3 The matrices $\mathbf{p}$, $\mathbf{S}$, and $\mathbf{Q}$ in Eqs. 8–12 satisfy the physics-imposed constraints, e.g., that saturation is a non-negative number between 0 and 1. Additionally, the transition from Eqs. 10 to 11 requires the matrix $\mathbf{D}$ to be invertible. The latter requirement is not satisfied when one of the phase saturation approaches 0, the derivatives of the saturation-pressure curve that form the matrix $\mathbf{D}$ become 0, yielding a singular $\mathbf{D}$. This phase-vanishing scenario can be treated by introducing either negative saturation [18], or a new set of primary variables [19] or a regularization term [20]. The use of ETD in such a scenario requires further exploration.

4 Numerical implementation

Explicit ETD schemes of arbitrary order have been derived in [7]. We explore several alternative implementations of the modified ETD approach, including its combination with Runge-Kutta (RK) methods. This is done for second- and fourth-order schemes, but higher-order approximations can be derived in a similar manner. However, these numerical schemes are not directly applicable to the multiphase flow equations of Section 3, for which the matrix $\mathbf{L}$ in Eqs. 3 and 12 is not invertible. The possibility of combining ETD with RK methods in this setting remains unexplored.

4.1 Multi-Step Method: High-Order ETD Schemes

Alternative ways to approximate the integral term in Eq. 6 give rise to different ETD schemes. For example, one can treat the operator $\mathbf{N}(\cdot, t)$ as constant during each time step, so that Eq. 6 reduces to

$$\mathbf{u}_{n+1} = \mathrm{e}^{\mathbf{L}\tau}\mathbf{u}_n + \boldsymbol{\Psi}_1\mathbf{N}_n, \tag{13a}$$

where $\mathbf{N}_m = \mathbf{N}(\mathbf{u}(t_m), \mathbf{x}, t_m)$ for any $m \in \mathbb{N}$, and

$$\boldsymbol{\Psi}_1 = \mathbf{L}^\dagger(\mathrm{e}^{\mathbf{L}\tau} - \mathbf{I}) + \tau\mathrm{e}^{\mathbf{L}\tau}(\mathbf{I} - \mathbf{L}^\dagger\mathbf{L}), \tag{13b}$$

where $\mathbf{I}$ is the $(N^2) \times (N^2)$ identity matrix, and $\mathbf{L}^\dagger$ is the pseudo-inverse of $\mathbf{L}$. We use a singular-value decomposition to represent $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{V}\mathbf{D}\mathbf{U}^\top, \tag{13c}$$

where $\mathbf{V}$ and $\mathbf{U}$ are $(N^2) \times (N^2)$ orthogonal matrices, and $\mathbf{D}$ is a $(N^2) \times (N^2)$ diagonal matrix. Then,

$$\mathbf{L}^\dagger = \mathbf{U}\mathbf{D}^\dagger\mathbf{V}^\top, \tag{13d}$$

where $\mathbf{D}^\dagger$ denotes the inverse $\mathbf{D}$ except for its 0 elements. For example, if $\mathbf{D} = \mathrm{diag}(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)$, then $\mathbf{D}^\dagger = \mathrm{diag}(1/\lambda_1, \ldots, 1/\lambda_r, 0, \ldots, 0)$ [21, 22]. We refer to this first-order ETD as ETD1.

4.1.1 Second-Order ETD

A second-order ETD scheme, ETD2, is constructed by approximating $\mathbf{N}(\mathbf{u}(t_n + s), \mathbf{x}, t_n + s)$ on the interval $0 \leq s \leq \tau$ with a linear function,

$$\mathbf{N} \approx \mathbf{N}_n + \frac{\mathbf{N}_n - \mathbf{N}_{n-1}}{\tau} s, \tag{14}$$

(recall that $t_{n+1} = t_n + \tau$). Substituting Eq. 14 into Eq. 6 yields

$$\mathbf{u}_{n+1} = (\mathrm{e}^{\mathbf{L}\tau} + \boldsymbol{\Psi}_1)\mathbf{u}_n + \boldsymbol{\Psi}_2\frac{\mathbf{N}_n - \mathbf{N}_{n-1}}{\tau}, \tag{15a}$$

where

$$\boldsymbol{\Psi}_2 = (\mathbf{L}^\dagger)^2(\mathrm{e}^{\mathbf{L}\tau} - \mathbf{I} - \tau\mathbf{L}) + \frac{\tau^2}{2}\mathrm{e}^{\mathbf{L}\tau}(\mathbf{I} - \mathbf{L}^\dagger\mathbf{L}). \tag{15b}$$

4.1.2 Fourth-Order ETD

A fourth-order ETD scheme, ETD4, is a special case of higher-order ETD schemes. A scheme of order τ^{P+1} is constructed by approximating $\mathbf{N}(\mathbf{u}(t_n + s), \mathbf{x}, t_n + s)$ on the interval $0 \leq s \leq \tau$ with a Pth-degree polynomial in s. Substituting this approximation into Eq. 6 leads to [7]

$$\mathbf{u}_{n+1} = \mathrm{e}^{\mathbf{L}\tau}\mathbf{u}_n + \tau\sum_{m=0}^{P-1} g_m \sum_{k=0}^{P} (-1)^k \binom{m}{k}\mathbf{N}_{n-k}, \tag{16a}$$

where $\mathbf{L}hg_0 = \mathrm{e}^{\mathbf{L}h} - \mathbf{I}$ and, for $m \geq 0$,

$$\mathbf{L}hg_{m+1} + \mathbf{I} = \sum_{k=0}^{m} \frac{g_k}{m+1-k}. \tag{16b}$$

ETD4 is derived by setting $P = 3$ in Eq. 16. The equation above is valid for invertible $\mathbf{L}$ matrices. Its generalization to the noninvertible case remains to be explored. Consequently, we do not rely on ETD4 in the numerical experiments reported below.

4.2 One-Step Method: Fusion of ETD and RK

The ETD schemes in Section 4.1 fall under the category of multistep methods, since they require prior evaluations of the nonlinear term $\mathbf{N}$. Initiation of such a scheme is problematic because the previous state of $\mathbf{N}$ is not available for the first time step. RK methods not only solve this problem but also typically have smaller error constants and larger stability regions than multistep methods [7]. An RK method of order M yields a solution of Eqs. 3 and 11 in the form

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau\sum_{i=1}^{M} c_i\mathbf{f}(\mathbf{u}_{n,i}), \quad \mathbf{u}_{n,i} = \mathbf{u}(t_{n,i}). \tag{17}$$

Here, c_i are the weights, $t_{n,i} \in [t_n, t_{n+1}]$ for all $i \leq m$, and the vector function $\mathbf{f}$ is defined below.

A second-order ($M = 2$) RK scheme, RK2, is

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau \frac{\mathbf{k}_1 + \mathbf{k}_2}{2}, \tag{18a}$$

where

$$\mathbf{k}_1 = \mathbf{L}\mathbf{u}_n + \mathbf{N}(\mathbf{u}_n), \tag{18b}$$

$$\mathbf{k}_2 = \mathbf{L}\mathbf{u}_n + \mathbf{L}\mathbf{k}_1 + \mathbf{N}(\mathbf{u}_n + \mathbf{k}_1). \tag{18c}$$

A fourth-order ($M = 4$) RK scheme, RK4, is

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \tau \frac{\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4}{6}, \tag{19a}$$

where

$$\mathbf{k}_1 = \mathbf{L}\mathbf{u}_n + \mathbf{N}(\mathbf{u}_n, \cdot), \tag{19b}$$

$$\mathbf{k}_2 = \mathbf{L}\mathbf{u}_n + \mathbf{N}(\mathbf{u}_n + \mathbf{k}_1/2, \cdot), \tag{19c}$$

$$\mathbf{k}_3 = \mathbf{L}\mathbf{u}_n + \mathbf{N}(\mathbf{u}_n + \mathbf{k}_2/2, \cdot), \tag{19d}$$

$$\mathbf{k}_4 = \mathbf{L}\mathbf{u}_n + \mathbf{N}(\mathbf{u}_n + \mathbf{k}_3, \cdot). \tag{19e}$$

4.2.1 ETD-RK2

This scheme is constructed by augmenting RK2 in Eq. 18 with ETD1 in Eq. 13. First, we compute

$$\mathbf{a}_n = \mathrm{e}^{\mathbf{L}\tau}\mathbf{u}_n + \boldsymbol{\Psi}_1 \mathbf{N}_n. \tag{20a}$$

Then, a solution is advanced to the next time step,

$$\mathbf{u}_{n+1} = \mathbf{a}_n + \boldsymbol{\Psi}_2 \frac{\mathbf{N}(\mathbf{a}_n) - \mathbf{N}_n}{\tau}. \tag{20b}$$

4.2.2 ETD-RK4 with Contour Integration

ETD-RK4 is constructed by augmenting RK4 in Eq. 19 with the ETD approximation of $\mathbf{u}$ in between the current and future time steps [7]. The resulting expressions are derived with a symbolic manipulation system.

$$\begin{aligned} \mathbf{u}_{n+1} = \mathrm{e}^{\mathbf{L}\tau}\mathbf{a}_n + \tau^{-2}\mathbf{L}^{-3}[&\boldsymbol{\Omega}_1 \mathbf{N}_{n+1/2} \\ &+ \boldsymbol{\Omega}_2 \mathbf{N}_{ab,n+1/2} + \boldsymbol{\Omega}_3 \mathbf{N}_{c,n+1}]. \end{aligned} \tag{21a}$$

Here,

$$\begin{aligned} \mathbf{N}_{n+1/2} &= \mathbf{N}(u_n), \\ \mathbf{N}_{ab,n+1/2} &= \mathbf{N}_{a,n+1/2} + \mathbf{N}_{b,n+1/2}, \\ \mathbf{N}_{a,n+1/2} &= \mathbf{N}(a_n), \\ \mathbf{N}_{b,n+1/2} &= \mathbf{N}(b_n), \\ \mathbf{N}_{c,n+1} &= \mathbf{N}(c_n); \end{aligned} \tag{21b}$$

$$\begin{aligned} \boldsymbol{\Omega}_1 &= -4\mathbf{I} - \tau\mathbf{L} + \mathrm{e}^{\mathbf{L}\tau}(4\mathbf{I} - 3\tau\mathbf{L} + \tau^2\mathbf{L}^2), \\ \boldsymbol{\Omega}_2 &= 4\mathbf{I} + 2\tau\mathbf{L} - 2\mathrm{e}^{\mathbf{L}\tau}(2\mathbf{I} - \tau\mathbf{L}), \\ \boldsymbol{\Omega}_3 &= -4\mathbf{I} - 3\tau\mathbf{L} - \tau^2\mathbf{L}^2 + \mathrm{e}^{\mathbf{L}\tau}(4\mathbf{I} - \tau\mathbf{L}); \end{aligned} \tag{21c}$$

and

$$\begin{aligned} \mathbf{a}_n &= \mathbf{e}_n + \mathbf{E}\mathbf{N}_n, \\ \mathbf{b}_n &= \mathbf{e}_n + \mathbf{E}\mathbf{N}_{a,n+1/2}, \\ \mathbf{c}_n &= \mathrm{e}^{\mathbf{L}\tau/2}\mathbf{a}_n + \mathbf{E}(2\mathbf{N}_{b,n+1/2} - \mathbf{N}_{n+1/2}), \\ \mathbf{e}_n &= \exp(\mathbf{L}\tau/2)\mathbf{u}_n \end{aligned} \tag{21d}$$

with $\mathbf{E} = \mathbf{L}^{-1}[\exp(\mathbf{L}\tau/2) - \mathbf{I}]$.

This form of ETD-RK4, and of any other ETD scheme of order higher than two, is numerically unstable [23]. To understand why, consider the expression

$$g(z) = \frac{\mathrm{e}^z - 1}{z}. \tag{22}$$

Its accurate computation is a well-known challenge in numerical analysis [24]. That is because, for small z, it suffers from cancellation error. The ETD-RK4 scheme (21) is expected to suffer from the same problem because it contains the terms

$$\boldsymbol{\alpha} = \frac{\mathbf{L}^{-3}\boldsymbol{\Omega}_1}{\tau^2}, \quad \boldsymbol{\beta} = \frac{\mathbf{L}^{-3}\boldsymbol{\Omega}_2}{\tau^2}, \quad \boldsymbol{\gamma} = \frac{\mathbf{L}^{-3}\boldsymbol{\Omega}_3}{\tau^2} \tag{23}$$

that can be thought of as higher-order analogs of Eq. 22. The cancellation errors in these expressions are even more pronounced, especially when the discretized linear operator matrix $\mathbf{L}$ has eigenvalues close to zero. This vulnerability to cancellation errors in the higher-order ETD and ETD-RK schemes can render them effectively useless [23].

Complex analysis provides a means to combat this cancellation problem. Instead of directly evaluating the function $g(z)$ at points z close to its singularity point $z = 0$, one computes a contour integral

$$g(z) = \frac{1}{2i\pi} \oint \frac{g(t)}{t - z} \mathrm{d}t. \tag{24}$$

This strategy works also when z is not close to the singularity point. If the matrix $\mathbf{L}$ replaces the scalar z, as in Eq. 23, the term $(t - z)^{-1}$ has to be replaced with the resolvent matrix $(t\mathbf{I} - \mathbf{L})^{-1}$, giving rise to

$$g(z) = \frac{1}{2i\pi} \oint (t\mathbf{I} - \mathbf{L})^{-1} g(t) \mathrm{d}t. \tag{25}$$

4.2.3 ETD-RK4 with Lie Group Analysis

The Lie group method [25] provides an alternative approach to deriving a workable ETD-RK4 scheme. A key idea of this method is to approximate the nonlinear term $\mathbf{N}$ in Eq. 3 with an Mth degree polynomial in time,

$$\mathbf{N}(\mathbf{u}(t + \tau), t + \tau) = \sum_{k=0}^{M-1} \frac{\tau^k}{k!} \mathbf{N}_k, \quad \mathbf{N}_k \in \mathbb{R}^{N_{\text{dis}}}, \tag{26}$$

which allows one to solve the resulting ODEs exactly. Specifically, a solution of Eq. 3 with $\mathbf{N}$ in Eq. 26 is

$$\mathbf{u}(t + \tau) = \mathrm{e}^{\tau\mathbf{L}}\mathbf{u}(t) + \int_0^\tau \mathrm{e}^{(\tau - s)\mathbf{L}} \sum_{k=0}^{M-1} \frac{\tau^k}{k!} \mathbf{N}_k \mathrm{d}s \tag{27}$$

or

$$\mathbf{u}(t + \tau) = \mathrm{e}^{\tau\mathbf{L}}\mathbf{u}(t) + \sum_{k=1}^{M} \tau^k \boldsymbol{\phi}_k(\tau\mathbf{L})\mathbf{N}_{k-1}, \tag{28a}$$

where

$$\boldsymbol{\phi}_k(\tau\mathbf{L}) = \int_0^\tau \frac{s^{k-1}}{(k-1)!}\mathrm{e}^{(\tau-s)\mathbf{L}}\mathrm{d}s. \tag{28b}$$

To account for the possibility that the matrix $\mathbf{L}$ is not invertible, the matrices $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$, and $\boldsymbol{\phi}_3$ are modified to

$$\begin{aligned}
\boldsymbol{\phi}_1 &= \tau\mathbf{A} + \mathbf{L}^\dagger(\mathrm{e}^{\tau\mathbf{L}} - \mathbf{I}) \\
\boldsymbol{\phi}_2 &= \frac{\tau^2}{2}\mathbf{A} + (\mathbf{L}^\dagger)^2(\mathrm{e}^{\tau\mathbf{L}} - \mathbf{I} - \tau\mathbf{L}) \\
\boldsymbol{\phi}_3 &= \frac{\tau^3}{3}\mathbf{A} + (\mathbf{L}^\dagger)^3[2\mathrm{e}^{\tau\mathbf{L}} - 2\mathbf{I} - \tau\mathbf{L}(2\mathbf{I} + \tau\mathbf{L})],
\end{aligned} \tag{28c}$$

where $\mathbf{A} = \mathrm{e}^{\tau\mathbf{L}}(\mathbf{I} - \mathbf{L}^\dagger\mathbf{L})$. Hence, in accordance with Eq. 28, the time-stepping from $\mathbf{u}_n = \mathbf{u}(t)$ to $\mathbf{u}_{n+1} = \mathbf{u}(t+\tau)$ in the forth-order ($M = 4$) scheme is carried out as

$$\mathbf{u}_{n+1} = \mathrm{e}^{\tau\mathbf{L}}\mathbf{u}_n + \tau(\mathcal{T}_1 + \mathcal{T}_2 + \mathcal{T}_3). \tag{29a}$$

Here,

$$\begin{aligned}
\mathcal{T}_1 &= (\boldsymbol{\phi}_1 - 3\boldsymbol{\phi}_2 + 4\boldsymbol{\phi}_3)\mathbf{N}_1, \\
\mathcal{T}_2 &= (2\boldsymbol{\phi}_2 - 4\boldsymbol{\phi}_3)(\mathbf{N}_2 + \mathbf{N}_3), \\
\mathcal{T}_3 &= (-\boldsymbol{\phi}_2 + 4\boldsymbol{\phi}_3)\mathbf{N}_4,
\end{aligned} \tag{29b}$$

with

$$\begin{aligned}
\mathbf{N}_1 &= \mathbf{N}(\mathbf{u}_n, t), \quad \mathbf{N}_2 = \mathbf{N}(\mathbf{u}_n^{(2)}, t + \tfrac{\tau}{2}), \\
\mathbf{N}_3 &= \mathbf{N}(\mathbf{u}_n^{(3)}, t + \tfrac{\tau}{2}), \quad \mathbf{N}_4 = \mathbf{N}(\mathbf{u}_n^{(4)}, t + \tau)
\end{aligned} \tag{29c}$$

and

$$\begin{aligned}
\mathbf{u}_n^{(2)} &= \mathrm{e}^{\frac{\tau\mathbf{L}}{2}}\mathbf{u}_n + \tfrac{\tau}{2}\boldsymbol{\phi}_1\big(\tfrac{\tau\mathbf{L}}{2}\big)\mathbf{N}_1, \\
\mathbf{u}_n^{(3)} &= \mathrm{e}^{\frac{\tau\mathbf{L}}{2}}\mathbf{u}_n + \tfrac{\tau}{2}\boldsymbol{\phi}_1\big(\tfrac{\tau\mathbf{L}}{2}\big)\mathbf{N}_2, \\
\mathbf{u}_n^{(4)} &= \mathrm{e}^{\frac{\tau\mathbf{L}}{2}}\mathbf{u}_n^{(2)} + \tau\boldsymbol{\phi}_1\big(\tfrac{\tau\mathbf{L}}{2}\big)\left(-\tfrac{1}{2}\mathbf{N}_2 + \mathbf{N}_3\right).
\end{aligned} \tag{29d}$$

Remark 4 Several of the numerical schemes described above require multiple function evaluations. Their computational cost is negligible under conditions of time regularity, which allows one to precompute certain matrices, e.g., $\boldsymbol{\Psi}_1$ and $\boldsymbol{\Psi}_2$ in Eq. 15 and the contour integral in Eq. 24, prior to the simulation. Even in the absence of time regularity, the scheme remains efficient as long as each chosen time step is small enough to stay in the stability region [26]. Although the benefit of pre-computing these matrices is lost, the cost of their computation does not dominate the overall budget.

5 Simulation results

The simulations reported in this section represent a reservoir with uniform initial pressure $p(\mathbf{x}, 0) = p_{\text{in}}$ and oil saturation $S_{\text{oil}}(\mathbf{x}, 0) = 1$. A production well, located at the center of the square reservoir of size $L_x = L_y = L$, is operated at a constant rate q. Once the pressure $p(\mathbf{x}, t)$ drops below the bubble point p_{bub}, some gas is produced. The values of these and other parameters used in our simulations come from [17]; they are collated in Table 1.

Figure 1 exhibits the temporal evolution of pressure, $p(\mathbf{x}_{\text{w}}, t)$, and flow rates of oil, $Q_{\text{oil}}(\mathbf{x}_{\text{w}}, t)$, and gas,

Table 1 Numerical values of the parameters and other quantities used in our simulations are taken from [17]

Quantity	Value	Value (SI)
Square reservoir size, L	3500 ft	1066.8 m
Reservoir thickness, L_z	100 ft	30.5 m
Top depth, D_{top}	5000 ft	1523.9 m
Longitudinal permeability, k_x	200 md	0.197 μm^2
Transverse permeability, k_y	100 md	0.099 μm^2
Porosity, ϕ	0.25	
Rock compressibility, c_{r}	0	
Oil density, ρ_{oil}	49.1 lbm/ft^3	786.5 kg/m^3
Oil viscosity, μ_{oil}	2 cp	2 mPa·s
Initial pressure, p_{in}	6000 psi	4.137×10^7 Pa
Bottom-hole pressure, p_{bth}	2000 psi	1.379×10^7 Pa
Atmospheric pressure, p_{atm}	14.7 psi	1.013×10^5 Pa
Bubbling pressure, p_{bub}	3400 psi	2.344×10^7 Pa
Number of grid blocks, N	$N_x = N_y = 35$	
Well location, $\mathbf{x}_{\text{w}} = (N_{x_{\text{w}}}, N_{y_{\text{w}}})$	(18, 18)	
Pumping rate, q	2000 STB/day	3.68×10^{-3} m^3/s

The square reservoir, discretized with the equal numbers of grid blocks in each dimension, is used for simplicity; the relevant matrices, such as transmissibility matrix, are readily computable for more complex geometries (see the Appendix)

$Q_{\mathrm{gas}}(\mathbf{x}_{\mathrm{w}}, t)$, at the extraction well $\mathbf{x}_{\mathrm{w}} = (N_{x_{\mathrm{w}}}, N_{y_{\mathrm{w}}})$. These quantities of interest were alternatives computed with the ETD-RK4 scheme (29) and the commercial reservoir simulation software Eclipse. The slight disagreement between the two solutions stems, in part, from the difference in the fluid properties: while our simulations describe them with the closed-form relations in Eq. 7, Eclipse calculates them by interpolating between the fluid properties from a table. Since the discrepancy is negligible for the average reservoir pressure $p_{\mathrm{ave}}(t) = L^{-2} \int p(\mathbf{x}, t)\mathrm{d}\mathbf{x}$ (Fig. 1a), we carry out our assessment of the relative performance of the ETD-RK schemes by focussing on the well-block pressure $p(\mathbf{x}_{\mathrm{w}}, t)$. The latter is also used directly to calculate the oil and gas flow rates in Fig. 1b.

5.1 Selection of the linear operator

As discussed earlier, the multiphase flow (7) do not have a natural stiff linear part $\mathbf{L}$ required by ETD. Instead, selection of $\mathbf{L}$ in Eq. 12 or, more precisely, of $\tilde{\mathbf{u}}$ is arbitrary. We compare the performance of the ETD schemes with three choices of the matrix $\mathbf{L} = \mathbf{D}^{-1}(\tilde{\mathbf{u}})\mathbf{T}(\tilde{\mathbf{u}})$. These are $\mathbf{L}_{\mathrm{high}}$, $\mathbf{L}_{\mathrm{low}}$, and $\mathbf{L}_{\mathrm{med}}$ corresponding, respectively, to $\tilde{\mathbf{u}}$ given by the initial field pressure p_{in}, bottom-hole pressure p_{bth}, and the average field pressure $p_{\mathrm{avg}} = (p_{\mathrm{in}} + p_{\mathrm{bth}})/2$.

All these versions of the matrix $\mathbf{L}$ are penta-diagonal, being the product of the inverse diagonal matrix $\mathbf{D}^{-1}(\tilde{\mathbf{u}})$ and transmissibility matrix $\mathbf{T}(\tilde{\mathbf{u}})$ in two dimensions. Consequently, $\mathbf{L}$ is very sparse,

$$\left[\begin{array}{ccccccccccccccc}
c & b & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
b & a & b & \cdots & d & & & & & & & & & & \vdots \\
0 & b & a & b & \cdots & d & & & & & & & & & \vdots \\
\vdots & & \ddots & \ddots & \ddots & & \ddots & & & & & & & & \vdots \\
\vdots & & \ddots & & \ddots & \ddots & \ddots & & \ddots & & & & & & \vdots \\
\vdots & & & & d & \cdots & b & a & b & \cdots & d & & & & \vdots \\
\vdots & & & & & & \ddots & & \ddots & \ddots & \ddots & & \ddots & & \vdots \\
\vdots & & & & & & & \ddots & & \ddots & \ddots & \ddots & & & \vdots \\
\vdots & & & & & & & & & d & \cdots & b & a & b & 0 \\
\vdots & & & & & & & & & & d & \cdots & b & a & b \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & b & c
\end{array}\right],$$

with the omitted elements being zeros. Values of the constants a, b, c, and d for $\mathbf{L}_{\mathrm{high}}$, $\mathbf{L}_{\mathrm{med}}$ and $\mathbf{L}_{\mathrm{low}}$ are given in Table 2. Of the three matrices, $\mathbf{L}_{\mathrm{high}}$ is the most strongly diagonal. This implies that setting $\mathbf{L} = \mathbf{L}_{\mathrm{high}}$ transfers most of the stiffness to the linear component of the multiphase flow Eq. 11. On the other hand, choosing the constant matrix $\mathbf{L}$ to be either $\mathbf{L}_{\mathrm{med}}$ or $\mathbf{L}_{\mathrm{low}}$ provides less separation of stiffness, i.e., the nonlinear component $\mathbf{N}$ retains much of stiffness, which defeats the purpose of ETD.

We report the accuracy of the alternative ETD schemes in terms of the normalized error over the simulation time horizon $[0, T]$, with $T = 600$ days,

$$\mathcal{E}(\tau) = \frac{|\int_0^T p(\mathbf{x}_{\mathrm{w}}, t)\mathrm{d}t - \int_0^T p^{\mathrm{ex}}(\mathbf{x}_{\mathrm{w}}, t)\mathrm{d}t|}{\int_0^T p^{\mathrm{ex}}(\mathbf{x}_{\mathrm{w}}, t)\mathrm{d}t}. \qquad (30)$$

In lieu of the exact solution, $p^{\mathrm{ex}}(\mathbf{x}_{\mathrm{w}}, t)$, we use a numerical solution obtained via the ETD-RK4 method (29) with the smallest time step $\tau = 0.001$ day as a close estimate. The trapezoidal rule is used to evaluate the quadratures in Eq. 30.

Figure 2 demonstrates the relative performance of the second- and fourth-order ETD, with and without RK. The upper range in these plots (steep increase in the error $\mathcal{E}$) corresponds to the largest time step τ for which the simulations fail to converge. The second-order methods, ETD2, converge regardless of the choice of the linear operator $\mathbf{L}$ (Fig. 2a). The choice of $\mathbf{L}_{\mathrm{high}}$ ensures the method's convergence at much larger time steps τ relative to $\mathbf{L}_{\mathrm{med}}$ and $\mathbf{L}_{\mathrm{low}}$. The importance of choosing an appropriate matrix $\mathbf{L}$ is even more pronounced for higher-order ETD schemes, such as ETD-RK4 (Fig. 2b). The Lie group-based ETD-RK4 method fails to converge if $\mathbf{L} = \mathbf{L}_{\mathrm{med}}$ and $\mathbf{L}_{\mathrm{low}}$. Even when convergent, the higher-order schemes with a suboptimal choice of $\mathbf{L}$ offers little advantage over their second-order counterparts in terms of computational efficiency: the allowable time steps for ETD2 and ETD4 are very similar for $\mathbf{L}_{\mathrm{med}}$ and $\mathbf{L}_{\mathrm{low}}$, while the performance gains provided by $\mathbf{L}_{\mathrm{high}}$ are massive. That is to be expected, since the system is most stiff at its initial state when the pressure gradient close to the well and, hence, the oil and gas flow rates are large. For this reason, it is most appropriate to construct the matrix $\mathbf{L}$ by using $\tilde{\mathbf{u}}$ to be the initial field pressure p_{in}. If the bottom-hole pressure and, hence, the final solution pressure are drastically different from the initial pressure p_{in}, an optimal pressure to evaluate $\mathbf{L}$ would be further from p_{in}. However, $\mathbf{L}_{\mathrm{high}}$ would still be the most optimal choice among the three.

5.2 Comparison with explicit (RK) schemes

Figure 3 provides a comparison of the accuracy and efficiency of the explicit second-order schemes with and without ETD. The normalized error $\mathcal{E}$ grows approximately linearly in the log-log scale for all the scheme considered (Fig. 3a). The second-order RK scheme, RK2, provides baseline; its maximum allowable time step is around $\tau = 0.15$ day. Even without RK, the ETD methods outperform

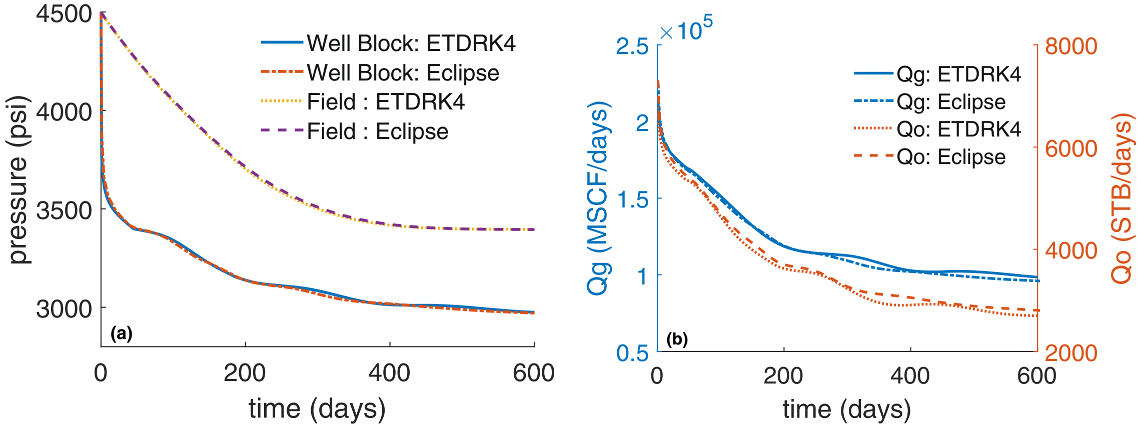

Fig. 1 Quantities of interest computed with the ETD-RK4 scheme (29) and the commercial reservoir simulation software Eclipse. (a) Pressure at the extraction well, $p(\mathbf{x}_{\mathrm{w}}, t)$, and average reservoir pressure $p_{\mathrm{ave}}(t)$. (b) flow rates of oil, $Q_{\mathrm{oil}}(\mathbf{x}_{\mathrm{w}}, t)$, and gas, $Q_{\mathrm{gas}}(\mathbf{x}_{\mathrm{w}}, t)$, at the well

RK2, having the maximum allowable time step of around $\tau = 0.75$ day. The fusion of RK2 and ETD results in the best performance; the largest possible time step of ETD-RK2 is around $\tau = 1.5$ day, i.e., it is ten times faster than RK2 and twice as fast as ETD alone. For any time step size τ, the normalized error $\mathcal{E}$ is lowest for the stand-alone ETD scheme and highest for the RK2 method. The error of the combined scheme, ETD-RK2, lies in between these two.

Direct comparison of the total simulation time is not straightforward because it depends, in part, on the computer architecture. Nonlinearity of the multiphase flow Eqs. 7 requires matrix initialization at every time step. Initialization speed could vary greatly across language platforms and CPU. For this reason, we focus on the solver component of the computation and ignore the initialization process. This metric could heavily favor a scheme that requires a high number of iteration if each iteration is cheap. In the present context, such a favored scheme is RK2. At each time step, the total simulation time of ETD2 and ETD-RK2 is higher than that of RK2 because the former involve matrix multiplication as opposed to matrix-vector multiplication in RK2. The total simulation time of ETD-RK2 with the maximum allowable time step τ is around half of that of RK2 (Fig. 3b). Although the normalized error $\mathcal{E}$ of ETD-RK2 with the maximum allowable time step τ is larger than that of RK2, it is sufficient to accurately capture pressure dynamics.

Figure 3 also provides a comparison of the accuracy and efficiency of the explicit fourth-order methods with and without ETD. Similar to the second-order schemes, the fourth-order methods are robust to the extreme nonlinearity of the pressure profile. The stand-alone RK4 serves as the baseline algorithm, with the maximum allowable time step $\tau \approx 0.3$ day. We compare the performance of two ETD-RK4 variants: one based on the Lie group analysis, ETD-RK4-Lie, and the other based on the contour integration, ETD-RK4-TF. For sufficiently small time step sizes τ, the accuracy of ETD-RK4-Lie and ETD-RK4-TF is nearly identical (Fig. 3a). ETD-RK4-TF fails to converge at the time step size $\tau \approx 1.25$ day, while ETD-RK4-Lie remains stable for τ up to 3.25 day.

At each time step, the total simulation times of both ETD-RK4-Lie and ETD-RK4-TF are higher than that of RK4 because the former include matrix multiplication, as opposed to matrix-vector multiplication used in RK4. The fourth-order scheme requires more time for matrix initialization. However, it has to be done only once in the beginning of the time step loop. Similar to the second-order schemes, the total solver computation time of the scheme of choice (ETD-RK4-Lie) with its maximum allowable time step τ is around half that of RK4. Our numerical experiments reveal that the fourth-order scheme outperforms its second-order counterpart. Maximum allowable time step is larger for each corresponding scheme type (Fig. 3b).

Table 2 Non-zero components of the penta-diagonal matrix $\mathbf{L}$ for the three alternative choices of the reference pressure $\tilde{\mathbf{u}}$

Constant	$\mathbf{L}_{\mathrm{high}}$	$\mathbf{L}_{\mathrm{med}}$	$\mathbf{L}_{\mathrm{low}}$
a	−4.47	−0.0050	$-6.67 \cdot 10^{-4}$
b	1.28	0.0015	$1.90 \cdot 10^{-4}$
c	−3.20	−0.0038	$0.00 \cdot 10^{-4}$
d	1.92	0.0023	$2.86 \cdot 10^{-4}$

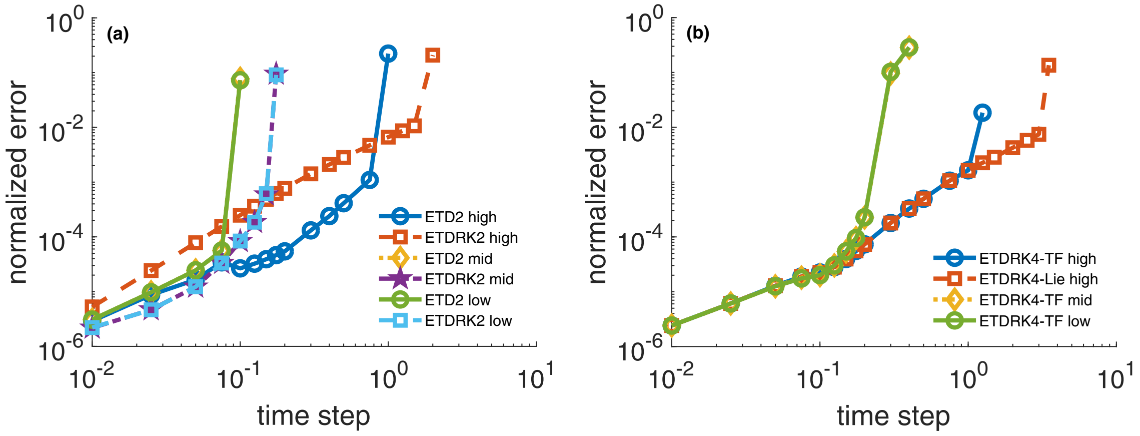

Fig. 2 The normalized error $\mathcal{E}$, defined in Eq. 30, as function of the time step τ, for (a) second-order and (b) fourth-order ETD methods

5.3 Comparison with an implicit scheme

Next, we compare the performance of our ETD-RK algorithms with that of the standard implicit method Eq. 40. One of the advantages of ETD is the absence of tuning parameters. That is in contrast to implicit algorithms, whose accuracy and efficiency typically depend on fine-tuning of several parameters related to Newton iterations. Two main design choices have direct impact on the performance of an implicit computation. The first choice deals with the residual matrix corresponding to each Newton iteration. In order for the iterative process to stop, the infinity (max) norm of the residual matrices $\mathbf{R}_l$ ($l =$ oil, gas) [27],

$$\|\mathbf{R}_l\|_\infty = \max_N |\mathbf{R}_l|, \tag{31}$$

has to be smaller than a specified value ϵ_1 (the first tuning parameter). Two additional criteria specify convergence tolerances ϵ_2 and ϵ_3 (the second and third tuning parameters) for gas saturation and pressure,

$$\max_{m\in[1,N]} \left|(\mathbf{S}_{\text{gas}})_m^{n+1,\nu} - (\mathbf{S}_{\text{gas}})_m^{n+1,\nu-1}\right| \le \epsilon_2 \tag{32}$$

$$\max_{m\in[1,N]} \left|\frac{(\mathbf{p})_m^{n+1,\nu} - (\mathbf{p})_m^{n+1,\nu-1}}{\mathbf{p}_{\text{avg}}^{n+1,\nu}}\right| \le \epsilon_3. \tag{33}$$

Here $(\mathbf{p})_m^{n+1,\nu}$ represents pressure component in $\hat{\mathbf{u}}^{n+1}$ of the mth gridblock in Eq. 40, ν denotes the number of the iteration in the Newton method, and $\mathbf{p}_{\text{avg}}^{n+1,\nu}$ represents the average pressure across the reservoir. The saturation is defined in a similar manner [27].

The second choice determines how adaptive time-stepping is performed. Ideally, the time step Δt should be small in the beginning when changes in the state variables are drastic, and large later on when these changes become less significant. This procedure complicates comparison of

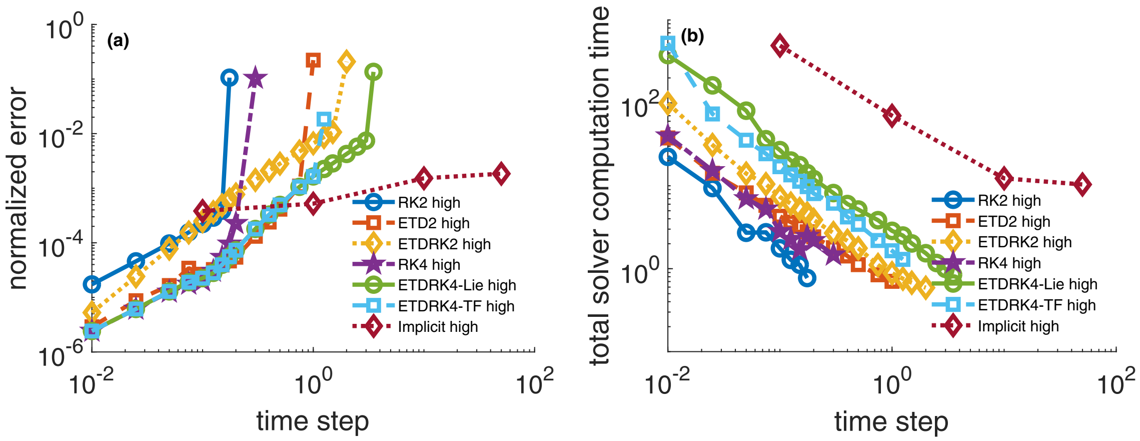

Fig. 3 The accuracy (a) and efficiency (b) of the implicit method and the alternative explicit (RK) second- and fourth-order schemes with and without ETD

the relative performance of the implicit and ETD schemes because the latter typically employs a constant time step τ. The use of a constant time step Δt in the implicit solver would introduce large errors in the beginning of the simulation. An adaptive time-stepping picks an initial time step Δt^0 (the fourth tuning parameter) and computes subsequent time steps from the relation

$$\Delta t^{n+1} = \min[\Delta t^n \mathcal{T}, \Delta t_{\max}], \tag{34a}$$

$$\mathcal{T} = \min_{\mathcal{V}} \left[\min_m \frac{(1+\omega)\eta_{\mathcal{V}}}{|\Delta_{\mathcal{V}}(m)| + \omega\eta_{\mathcal{V}}} \right]. \tag{34b}$$

Here $\Delta_{\mathcal{V}}(m)$ is the observed change in the state variable $\mathcal{V}$ (pressure, saturation) in the mth gridblock during the nth time step; $\eta_{\mathcal{V}}$ is a maximum allowable change of the variable $\mathcal{V}$ at any gridblock during any time step; and $0 \leq \omega \leq 1$. Values of these three tuning parameters and the other four defined above are collated in Table 3.

Figure 3 provides a comparison of the performance of the implicit scheme and the several variants of ETD-RK. In general, the implicit scheme has higher errors $\mathcal{E}$ than the ETD methods do; but, in the regions of high nonlinearity such as when the pressure p approaches the bubbling point p_{bub}, the ETD methods are more accurate than the implicit scheme only if they use relatively small time steps τ (Fig. 3a). The total solver computation time of the implicit scheme is dominated by the Jacobian iteration loop. The solver computation at the same time step is much larger in the implicit scheme. Therefore, even as the time step increases, the total solver computation time of the implicit scheme remains higher than that of the ETD variants (Fig. 3b). In addition, increasing the time step size does not always translates in less total computation time of the implicit scheme. Although there are fewer time steps to go through, each time step might take longer because it is more difficult for the Newton iteration with a larger time step to converge.

Remark 5 As expected, our numerical experiments show that the number of Newton iterations required to meet the convergence criterion increases with the time-step size. While the phase-vanishing scenarios were not explored in this study, one can expect the convergence of the Newton method under such conditions to be similar to other degenerate parabolic problems [20].

Table 3 Values of the tuning parameters in the implicit solver

Parameter	Value
ϵ_1	10^{-3}
ϵ_2	10^{-2}
ϵ_3	10^{-3}
η_s	0.05
η_p	50.0
ω	0.5
Δt^0	1.0

Remark 6 The Newton–Raphson method used in our simulations to benchmark ETD can be replaced with alternative iterative schemes such as Picard iteration or L-scheme [28, 29]. We leave detailed comparison of ETD with these and other implicit schemes for a follow-up study.

Finally, we compare our ETD method with a mixed implicit/explicit scheme IMPES, which is often used in simulations of multiphase flow in porous media. Figure 4 demonstrates that the ETD scheme is significantly more accurate than the IMPES method for relatively small time steps ($\Delta t \leq 0.3$ days). Similar to the implicit scheme, IMPES yields a lower error at large time steps. The IMPES performance depends on many tuning parameters in the way analogous to the implicit scheme discussed in Section 5.2. We used the parameter values collated in Table 3, except for ϵ_2 and η_s which are not relevant in IMPES since the implicit calculation is operated with the pressure matrix only while the saturation matrix is treated explicitly (46).

6 Summary and conclusions

Stiff problems are notoriously hard to solve explicitly because a required time step has to be sufficiently small to capture fast changes in a quantity of interest. Stiff

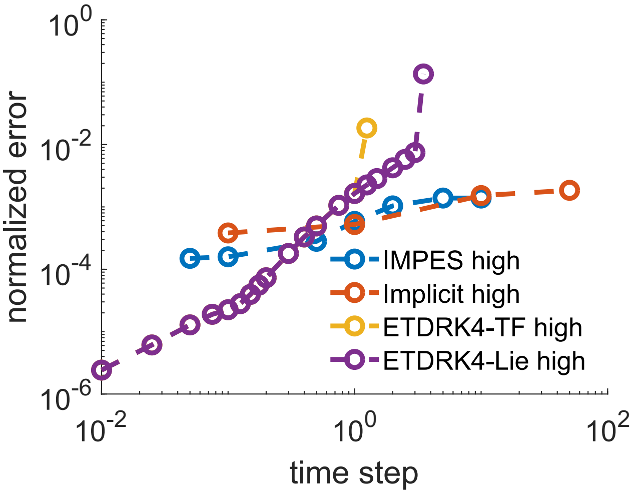

Fig. 4 The accuracy of IMPES, implicit, and selected top-performing ETD schemes

systems of partial differential equations often arise when multiple physical processes are considered. Multiphase flow of compressible fluids in porous media provides an example of such stiff problems.

Exponential time differencing (ETD) is an efficient scheme for stiff problems that can be separated into linear stiff and nonlinear non-stiff parts. We extended the range of applicability of ETD to problems, such as multiphase flow in porous media, that do not exhibit this natural separation of stiffness. Our approach is to isolate a linearized stiff elliptic operator and relegate the rest of the stiffness to the nonlinear term. The selection of the stiff linearized component is crucial to the overall performance, in terms of both accuracy and efficiency, of the resulting ETD algorithms. Our numerical experiments confirmed the optimality of a somewhat intuitive ETD design: the stiff linear operator should be constructed via linearization around the state where the solution changes most rapidly. In the context of multiphase flow, such term should be chosen at the initial field pressure.

We tested several variants of ETD, including its use in conjunction with the Runge-Kutta (RT) method. Their performance was compared with stand-along RK schemes and an implicit algorithm. Our numerical experiments revealed that the fusion of ETD and a fourth-order RK scheme, ETD-RK4, provides the best overall performance in terms of accuracy and computational efficiency.

Although we demonstrated that ETD can be employed to solve nonlinear problems without natural stiffness separation, several research directions remain to be explored. For example, it might be possible to increase the efficiency of ETD-RK4 further by improving its treatment of the singular matrices, e.g., via a combination of the contour integration and the Lie group analysis.

Numerical experiments similar to those reported in [30] can be used to investigate the asymptotic preserving property of our ETD schemes. Such an analysis would verify that a stiff-based scheme, such as ETD, performs well in much less stiff scenarios. Since various variables are involved in the calculation of each matrix in Eq. 11, choosing a single variable that can be used to vary the stiffness of a PDE is a complex task that is left for a follow-up study.

Appendix A: Fully Implicit Method and IMPES

We start by rewriting Eq. 7 as

$$\phi \frac{\partial}{\partial t}\left(\frac{S_{\text{oil}}}{B_{\text{oil}}}\right) + \nabla \cdot \left(\frac{\mathbf{U}_{\text{oil}}}{B_{\text{oil}}}\right) = q_{\text{oil}} \tag{35}$$
$$\phi \frac{\partial}{\partial t}\left(\frac{S_{\text{oil}} R_s}{B_{\text{oil}}} + \frac{S_{\text{gas}}}{B_{\text{gas}}}\right) + \nabla \cdot \left(\frac{\mathbf{U}_{\text{oil}} R_s}{B_{\text{oil}}} + \frac{\mathbf{U}_{\text{gas}}}{B_{\text{gas}}}\right) = q_{\text{gas}}.$$

where R_s is a solubility of the gas phase in the oil phase. A two-dimensional simulation domain is discretized with an $N \times N$ grid, whose cells are indexed alternatively either by (i, j) with $1 \leq i, j \leq N$ or by $k = N(i-1) + j$. Each cell is of size Δx and Δy and thickness h. A finite-volume discretization of these equations yields a vector of unknowns (pressure and saturation) of length $2N^2$,

$$\hat{\mathbf{u}} = \begin{bmatrix} \vdots \\ \begin{pmatrix} p_{\text{oil}} \\ s_{\text{gas}} \end{pmatrix}^k \\ \vdots \end{bmatrix}, \qquad k = 1, \cdots, N^2. \tag{36}$$

Transmissibility of oil/gas (l = oil, gas) between the cells (i, j) and $(i-1, j)$ and between the cells (i, j) and $(i, j-1)$ is defined as

$$\gamma^l_{i-\frac{1}{2},j} = \frac{\kappa \Delta y h \kappa_{\text{rel},l}}{\Delta x \mu_l B_l}, \qquad \gamma^l_{i,j-\frac{1}{2}} = \frac{\kappa \Delta x h \kappa_{\text{rel},l}}{\Delta y \mu_l B_l}, \tag{37a}$$

respectively. We also introduce

$$\beta_{i,j} = R_s \gamma^{\text{oil}}_{i,j} + \gamma^{\text{gas}}_{i,j}. \tag{37b}$$

We define a $2N^2 \times 2N^2$ block penta-diagonal transmissibility matrix $\hat{\mathbf{T}}$. Each component of this matrix, corresponding to the cell (i, j), is

$$\hat{\mathbf{T}}_{ij} = \begin{bmatrix} \vdots \\ \begin{pmatrix} \beta_{i-\frac{1}{2},j} & 0 \\ \gamma^{\text{oil}}_{i-\frac{1}{2},j} & 0 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \beta_{i,j-\frac{1}{2}} & 0 \\ \gamma^{\text{oil}}_{i,j-\frac{1}{2}} & 0 \end{pmatrix} \\ -\begin{pmatrix} \beta_{i-\frac{1}{2},j} + \beta_{i,j-\frac{1}{2}} + \beta_{i,j+\frac{1}{2}} + \beta_{i+\frac{1}{2},j} & 0 \\ \gamma^{\text{oil}}_{i-\frac{1}{2},j} + \gamma^{\text{oil}}_{i,j-\frac{1}{2}} + \gamma^{\text{oil}}_{i,j+\frac{1}{2}} + \gamma^{\text{oil}}_{i+\frac{1}{2},j} & 0 \end{pmatrix} \\ \begin{pmatrix} \beta_{i,j+\frac{1}{2}} & 0 \\ \gamma^{\text{oil}}_{i,j+\frac{1}{2}} & 0 \end{pmatrix} \\ \vdots \\ \begin{pmatrix} \beta_{i+\frac{1}{2},j} & 0 \\ \gamma^{\text{oil}}_{i+\frac{1}{2},j} & 0 \end{pmatrix} \end{bmatrix}. \tag{37c}$$

Furthermore, we introduce a flow-rate vector of length $2N^2$,

$$\hat{\mathbf{Q}} = \begin{bmatrix} \vdots \\ \begin{pmatrix} q_{\text{gas}} \\ q_{\text{oil}} \end{pmatrix}^k \\ \vdots \end{bmatrix}, \qquad k = 1, \cdots, N^2. \tag{38}$$

and discretize the terms

$$\phi \frac{\partial}{\partial t}\left(\frac{S_{\text{oil}}}{B_{\text{oil}}}\right) \quad \text{and} \quad \phi \frac{\partial}{\partial t}\left(\frac{S_{\text{oil}} R_s}{B_{\text{oil}}} + \frac{S_{\text{gas}}}{B_{\text{gas}}}\right)$$

in Eq. 7 as follows. Let d_{11} and d_{12} denote the change in oil-phase volume due to change in pressure and gas-phase saturation, respectively. Similarly, let d_{21} and d_{22} denote the change in gas-phase volume due to changes in pressure and gas-phase saturation, respectively. Then,

$$\Delta_t\left(\frac{\phi S_{\text{oil}}}{B_{\text{oil}}}\right) = d_{21}\Delta p_{\text{oil}} + d_{22}\Delta S_{\text{gas}} \tag{39a}$$

$$\Delta_t\left(\frac{\phi S_{\text{oil}} R_s}{B_{\text{oil}}} + \frac{\phi S_{\text{gas}}}{B_{\text{gas}}}\right) = d_{11}\Delta p_{\text{oil}} + d_{12}\Delta S_{\text{gas}}, \tag{39b}$$

where Δ_t is a suitable discretization of ∂_t, e.g., $\Delta_t X = (X^{n+1} - X^n)/\Delta t$. Finally, we define a $2N^2 \times 2N^2$ compressibility matrix

$$\hat{\mathbf{D}} = \begin{bmatrix} \ddots & & \\ & \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} & \\ & & \ddots \end{bmatrix}. \tag{39c}$$

The fully implicit method results in a system of ODEs,

$$\hat{\mathbf{D}}(\hat{\mathbf{u}}^{n+1})(\hat{\mathbf{u}}^{n+1} - \hat{\mathbf{u}}^n) = \hat{\mathbf{T}}(\hat{\mathbf{u}}^{n+1})\hat{\mathbf{u}}^{n+1} - \hat{\mathbf{Q}}(\hat{\mathbf{u}}^{n+1}), \tag{40}$$

Newton's method can be used to solve Eq. 40. At each iteration, the matrix $\hat{\mathbf{u}}^{n+1}$ is updated via the Jacobian matrix until convergence.

Unlike the fully implicit method that combines the oil and gas equations into a single large matrix equation, IMPES solves Eq. 35 by recasting it into two separate pressure and saturation matrix equations. In order to construct the pressure equation, the saturation is eliminated by rewriting Eq. 35 as

$$\begin{aligned}&\left(1 - \frac{d_{22}}{d_{12}}R_s\right)\nabla\cdot\left(\frac{\mathbf{U}_{\text{oil}}}{B_{\text{oil}}}\right) - \left(\frac{d_{22}}{d_{12}}\right)\nabla\cdot\left(\frac{\mathbf{U}_{\text{gas}}}{B_{\text{gas}}}\right)\\ &+\left(d_{21} - \frac{d_{22}}{d_{12}}d_{11}\right)\Delta p_{\text{oil}} = q_{\text{oil}} + \left(\frac{d_{22}}{d_{12}}\right)q_{\text{gas}}.\end{aligned} \tag{41}$$

Similar to Eq. 40, the matrix equation of the pressure equation can be written as

$$\begin{aligned}\check{\mathbf{D}}^p(\mathbf{p}^{n+1}, \mathbf{S}^n)(\mathbf{p}^{n+1} - \mathbf{p}^n) = \check{\mathbf{T}}(\mathbf{p}^{n+1}, \mathbf{S}^n)\mathbf{p}^{n+1}\\ - \check{\mathbf{Q}}(\mathbf{p}^{n+1}, \mathbf{S}^n).\end{aligned} \tag{42}$$

Hence, the solution for $\mathbf{p}$ at any time step depends on $\mathbf{S}$ at the previous time step, i.e., the pressure and saturation computations are decoupled. Similar to Eq. 37c, the transmissibility matrix of the pressure equation, $\check{\mathbf{T}}$, is the $N^2 \times N^2$ penta-diagonal matrix whose components are

$$\check{\mathbf{T}}_{ij} = \begin{bmatrix} \vdots \\ \left(\gamma^{\text{oil}}_{i-\frac{1}{2},j} - \frac{d_{22}}{d_{12}}\beta_{i-\frac{1}{2},j}\right) \\ \vdots \\ \left(\gamma^{\text{oil}}_{i,j-\frac{1}{2}} - \frac{d_{22}}{d_{12}}\beta_{i,j-\frac{1}{2}}\right) \\ \begin{pmatrix} -\gamma^{\text{oil}}_{i-\frac{1}{2},j} - \gamma^{\text{oil}}_{i,j-\frac{1}{2}} - \gamma^{\text{oil}}_{i,j+\frac{1}{2}} - \gamma^{\text{oil}}_{i+\frac{1}{2},j} \\ \frac{d_{22}}{d_{12}}(\beta_{i-\frac{1}{2},j} + \beta_{i,j-\frac{1}{2}} + \beta_{i,j+\frac{1}{2}} + \beta_{i+\frac{1}{2},j}) \end{pmatrix} \\ \left(\gamma^{\text{oil}}_{i,j+\frac{1}{2}} - \frac{d_{22}}{d_{12}}\beta_{i,j+\frac{1}{2}}\right) \\ \vdots \\ \left(\gamma^{\text{oil}}_{i+\frac{1}{2},j} - \frac{d_{22}}{d_{12}}\beta_{i+\frac{1}{2},j}\right) \end{bmatrix}. \tag{43}$$

The flow-rate vector of the pressure equation,

$$\check{\mathbf{Q}} = \begin{bmatrix} \vdots \\ \left(q_{\text{oil}} - \frac{d_{22}}{d_{12}}q_{\text{gas}}\right)^k \\ \vdots \end{bmatrix}, \qquad k = 1, \cdots, N^2. \tag{44}$$

has length N^2. Finally, we define a $N^2 \times N^2$ compressibility matrix of the pressure equation as

$$\check{\mathbf{D}} = \begin{bmatrix} \ddots & & \\ & \left(d_{21} - \frac{d_{22}}{d_{12}}d_{11}\right) & \\ & & \ddots \end{bmatrix}. \tag{45}$$

Once the pressure is computed at the $(n+1)$st time step, the saturation is explicitly determined by

$$\left(\frac{1}{d_{22}}R_s\right)\nabla\cdot\left(\frac{\mathbf{U}_{\text{oil}}}{B_{\text{oil}}}\right) + \left(\frac{d_{21}}{d_{22}}\right)\Delta p_{\text{oil}} + \Delta S_{\text{gas}} = \frac{q_{\text{oil}}}{d_{22}}. \tag{46}$$

The matrix of saturation equation is formulated as

$$\begin{aligned}\mathbf{S}^{n+1} &= \mathbf{S}^n + \mathbf{T}^{\mathrm{s}}(\mathbf{p}^{n+1}, \mathbf{S}^n)\mathbf{p}^{n+1}\\ &-\mathbf{D}^{\mathrm{s}}(\mathbf{p}^{n+1}, \mathbf{S}^n)(\mathbf{p}^{n+1} - \mathbf{p}^n) + \mathbf{Q}^{\mathrm{s}}(\mathbf{p}^{n+1}, \mathbf{S}^n).\end{aligned} \tag{47}$$

The $N^2 \times N^2$ penta-diagonal transmissibility matrix of the saturation equation, $\mathbf{T}^s$, is defined as

$$\mathbf{T}^{\mathrm{s}}_{ij} = \begin{bmatrix} \vdots \\ \left(\frac{1}{d_{22}}\gamma^{\text{oil}}_{i-\frac{1}{2},j}\right) \\ \vdots \\ \left(\frac{1}{d_{22}}\gamma^{\text{oil}}_{i,j-\frac{1}{2}}\right) \\ \left(-\frac{1}{d_{22}}(\gamma^{\text{oil}}_{i-\frac{1}{2},j}+\gamma^{\text{oil}}_{i,j-\frac{1}{2}}+\gamma^{\text{oil}}_{i,j+\frac{1}{2}}+\gamma^{\text{oil}}_{i+\frac{1}{2},j})\right) \\ \left(\frac{1}{d_{22}}\gamma^{\text{oil}}_{i,j+\frac{1}{2}}\right) \\ \vdots \\ \left(\frac{1}{d_{22}}\gamma^{\text{oil}}_{i+\frac{1}{2},j}\right) \end{bmatrix}, \tag{48}$$

the flow-rate vector of length N^2 as

$$\mathbf{Q}^{\mathrm{s}} = \begin{bmatrix} \vdots \\ \left(\frac{1}{d_{22}}q_{\text{oil}}\right)^k \\ \vdots \end{bmatrix}, \qquad k = 1, \cdots, N^2. \tag{49}$$

and the $N^2 \times N^2$ compressibility matrix of the saturation equation as

$$\mathbf{D}^s = \begin{bmatrix} \ddots & & \\ & \left(\frac{d_{21}}{d_{22}}\right) & \\ & & \ddots \end{bmatrix}. \tag{50}$$

References

1. Keyes, D.E., McInnes, L.C., Woodward, C., Gropp, W., et al.: MUltiphysics simulations: Challenges and opportunities. Int. J. High Perform. Comput. Appl. **27**(1), 4–83 (2012)
2. Miranker, W.L.: Numerical Methods for Stiff Equations. D. Reidel Publishing Co, Dordrecht (2001)
3. Berardi, M., Vurro, M.: The numerical solution of Richards? equation by means of method of lines and ensemble Kalman filter. Math. Comput. Simul. **125**, 38–47 (2016)
4. Dethlefsen, F., Beyer, C., Feeser, V., Köber, R.: Parameterizability of processes in subsurface energy and mass storage. Environ. Earth Sci. **75**, 885 (2016)
5. Yu, X., Moraetis, D., Nikolaidis, N.P., Li, B., Duffy, C., Liu, B.: A coupled surface-subsurface hydrologic model to assess groundwater flood risk spatially and temporally. Environ. Model. Soft. **114**, 129–139 (2019)
6. Bhattacharya, M.C.: An explicit conditionally stable finite difference equation for heat conduction problems. Int. J. Numer. Methods Eng. **21**(2), 239–265 (1985)
7. Cox, S.M., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)
8. Whalen, P., Brio, M., Moloney, J.V.: Exponential time-differencing with embedded Runge-Kutta adaptive step control. J. Comput. Phys. **280**, 579–601 (2015)
9. de la Hoz, F., Vadillo, F.: Numerical simulations of time-dependent partial differential equations. J. Comput. Appl. Math. **295**, 175–184 (2016)
10. Bhatt, A., Moore, B.E.: Structure-preserving exponential Runge-Kutta methods. SIAM J. Sci. Comput. **39**(2), A593–A612 (2017)
11. Macías-Díaz, J.E.: A dynamically consistent method to solve nonlinear multidimensional advection?reaction equations with fractional diffusion. J. Comput. Phys. **366**, 71–88 (2018)
12. Assouline, S., Tartakovsky, D.M.: Unsaturated hydraulic conductivity function based on a fragmentation process. Water Resour. Res. **37**(5), 1309–1312 (2001)
13. Schuetz, J., Kaiser, K.: A new stable splitting for singularly perturbed ODEs. Appl. Num. Math. **107**, 18–33 (2016)
14. Giraldo, F.X., Restelli, M.: A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. J. Comput. Phys. **227**, 3849–3877 (2008)
15. Ascher, U.M., Ruuth, S.J., Spiteri, R.J.: Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. **25**(2):151–167 (1997)
16. Aziz, K., Settari, A.: Petroleum reservoir simulation. Applied Science Publishers (1979)
17. Tchelepi, H., Cremon, M., Delgoshaie, A.H.: Lecture notes in reservoir simulation (2018)
18. Abadpour, A., Panfilov, M.: Method of negative saturations for modeling two-phase compositional flow with oversaturated zones Transp. Porous Media **79**, 197–214, 09 (2009)
19. Golmohammadi, A., Khaninezhad, M.-R.M., Jafarpour, B.: Group-sparsity regularization for ill-posed subsurface flow inverse problems. Water Resour. Res. **51**(10), 8607–8626 (2015)
20. Radu, F.A., Pop, I.S., Knabner, P.: Newton-type methods for the mixed finite element discretization of some degenerate parabolic equations. In: de Castro, A.B., Gómez, D., Quintela, P., Salgado, P. (eds.) Numerical Mathematics and Advanced Applications, pp. 1192–1200. Springer, Heidelberg (2006)
21. Moore-Penrose pseudoinverse (pinv) MATLAB:2020b.9.9.0.14 62360 (r2020b) (2020)
22. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
23. Kassam, A.-K., Trefethen, L.N.: Fourth-order time-stepping for stiff PDEs. SIAM J. Sci Comput. **26**(4), 1214–1233 (2005)
24. Friesner, R.A., Tuckerman, L.S., Dornblaser, B.C., Russo, T.V.: A method for exponential propagation of large systems of stiff nonlinear differential equations. J. Sci Comput. **4**(4), 327–354 (1989)
25. Krogstad, S.: Generalized integrating factor methods for stiff PDEs. J. Comput. Phys. **203**(1), 72–88 (2005)
26. Du, Q., Zhu, W.: Analysis and applications of the exponential time differencing schemes and their contour integration modifications. BIT Numer. Math. **45**, 307–328 (2005)
27. Schlumberger: ECLIPSE Reference Manual (2014)
28. List, F., Radu, F.A.: A study on iterative methods for solving Richards' equation. Comput Geosci. **20**, 341–353 (2016)
29. Mitra, K., Pop, I.S.: A modified L-Scheme to solve nonlinear diffusion problems. Comput. Math. Appl. **77**(6), 1722–1738 (2019)
30. Jin, S.: Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations. SIAM J. Sci. Comput. **21**(2), 441–454 (1999)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.