Comput Geosci (2012) 16:323–334
DOI 10.1007/s10596-011-9273-z

ORIGINAL PAPER

Exponential time integrators for stochastic partial differential equations in 3D reservoir simulation

Sebastian Geiger · Gabriel Lord · Antoine Tambue

Received: 11 February 2011 / Accepted: 14 December 2011 / Published online: 13 January 2012

Abstract The transport of chemically reactive solutes (e.g. surfactants, CO_2 or dissolved minerals) is of fundamental importance to a wide range of applications in oil and gas reservoirs such as enhanced oil recovery and mineral scale formation. In this work, we investigate exponential time integrators, in conjunction with an upwind weighted finite volume discretisation in space, for the efficient and accurate simulation of advection–dispersion processes including non-linear chemical reactions in highly heterogeneous 3D oil reservoirs. We model sub-grid fluctuations in transport velocities and uncertainty in the reaction term by writing the advection–dispersion–reaction equation as a stochastic partial differential equation with multiplicative noise. The exponential integrators are based on the variation of constants solution and solve the linear system exactly. While this is at the expense of computing the exponential of the stiff matrix representing the finite volume discretisation, the use of real Léja point or the Krylov subspace technique to approximate the exponential makes these methods competitive compared to standard finite difference-based time integrators. For the deterministic system, we investigate two exponential time integrators, the second-order accurate exponential Euler midpoint (EEM) scheme and exponential time differencing of order one (ETD1). All our numerical examples demonstrate that our methods can compete in terms of efficiency and accuracy compared with standard first-order semi-implicit time integrators when solving (stochastic) partial differential equations that model mixing and chemical reactions in 3D heterogeneous porous media. Our results suggest that exponential time integrators such as the ETD1 and EEM schemes could be applied to typical 3D reservoir models comprising tens to hundreds of thousands unknowns.

Keywords Advection–dispersion–reaction equations · Fast time stepping · SPDEs · Langevin equation · Sub-grid physics · Mixing and chemical reaction

Mathematics Subject Classifications (2010) 65M22 · 65C30 · 76S05

S. Geiger (✉)
Institute of Petroleum Engineering, Heriot-Watt University, Edinburgh, EH14 4AS UK
e-mail: sebastian.geiger@pet.hw.ac.uk

G. Lord
Department of Mathematics and Maxwell Institute of Mathematical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS UK
e-mail: gabriel@macs.hw.ac.uk

A. Tambue
Department of Mathematics, University of Bergen, Johannes Bruns Gate 12, 5008 Bergen, Norway
e-mail: Antoine.Tambue@math.uib.no

1 Introduction

Advection and diffusion can transport chemically reactive components such as dissolved minerals and gases, surfactants or polymers, over long distances through the highly heterogeneous porous media comprising oil and gas reservoirs. It is hence fundamental to reservoir engineering and includes processes like enhanced oil recovery due to controlled-salinity flooding [1], mineral scale formation [33] or subsurface CO_2 storage [30]. Predicting the spatial spreading and mixing of reactive

solutes in field applications by numerical simulations requires the efficient and accurate numerical solution of at least one advection–dispersion–reaction equation (ADR), which must adequately resolve the wide range in flow velocities and reaction rates intrinsic to 3D oil and gas reservoirs.

There are two fundamental problems with using the ADR to model reactive transport in heterogeneous porous media [11]: First, the ADR treats mechanical dispersion of solutes and diffusive mixing of solutes by an effective Fickian dispersion tensor. This overestimates the amount of mixing and mixing-induced reactions. Second, the actual reaction term is highly uncertain because (a) chemical reaction rates depend on pressure, temperature and composition, which change locally and transiently, and (b) the reactive surface area of the pore space evolves transiently and spatially. Both occur at scales that are commonly not resolved in a typical reservoir simulation, i.e. at a scale smaller than the size of a single grid block. It has recently been shown that by including a stochastic forcing term in the ADR, it is possible to model sub-grid variations in transport velocities causing mechanical dispersion and sub-grid diffusion during miscible single-phase flow with chemical reactions [36, 37]. A stochastic form of the ADR was also used to model trapping of an immiscible fluid phase during multi-phase flow [39, 40]. Here we write the ADR in a more general form that is as a stochastic partial equation with multiplicative noise to model sub-grid fluctuations and uncertainties in chemical reaction rates.

The classical deterministic ADR is usually discretised in space by the full range of spatial discretisations (e.g. finite differences, finite volumes or finite elements), and each method comprises its own body of literature. However, a fundamental challenge remains: How to integrate in time the system of stiff ODEs, representing transport and reaction processes evolving over multiple time scales, in a stable, accurate and efficient way while avoiding non-physical oscillations? The key problem in porous media flow is to overcome the limitations of stability criteria, such as the Courant–Friedrich–Levy criterion, when resolving the huge variation in competing transport and reaction rates. We recently showed that an alternative time integration strategy can be used: The exponential time differencing scheme of order one (ETD1), i.e. $\mathcal{O}(\Delta t)$, allows for the fast and accurate solution of the ADR in 2D [35].

The family of exponential integrators date back to the 1960s (see [16, 26] for history and references). These methods are based on approximating the corresponding integral formulation of the non-linear part of the differential equation, solving the linear part exactly and computing the exponential of a matrix, see, for example, [9, 13, 14, 16, 19] and [16, 26] for a review. Although exponential integrators have the advantage that they solve the linear part exactly in time, this is at the price of computing the exponential of a matrix, a notorious problem in numerical analysis [27]. However, new developments in real fast Léja point and Krylov subspace techniques for computing functions of the matrix exponential has revived interest in these methods. The real fast Léja point technique is based on matrix interpolation polynomials at spectral Léja sequences [2, 3]. The Krylov subspace technique is based on the idea of projecting the operator on a "small" Krylov subspace of the matrix via the Arnoldi process [13, 28, 32]. References for work with the Krylov subspace technique and real fast Léja points can be found in [35].

In this work, we solve the ADR (Eq. 1) with and without a stochastic forcing term for anisotropic and heterogeneous 3D porous media that exhibits high Péclet numbers, i.e. solute transport that is advection-dominated. For the deterministic case, we examine flows using two exponential integrators: Exponential time differencing of order $\mathcal{O}(\Delta t)$ (ETD1) and the exponential Euler midpoint integrators (EEM) of order $\mathcal{O}(\Delta t^2)$ [7]. For the stochastic case, we examine an exponential integrator [34] based on the ETD1 scheme, which we expect has order $\mathcal{O}(\Delta t^{1/2})$ for the system considered here. In all cases, we use a finite volume discretisation in space. The aim of this paper is to investigate these integrators, compare their performances in terms of efficiency and accuracy to a standard semi-implicit integrator for solving a non-linear ADR in highly anisotropic and heterogeneous porous media with high Péclet number flows where sub-grid flow variations and changes in reaction rates are modelled by stochastic forcing.

2 Model problem

For simplicity, we assume the movement of a chemically reactive solute in a single phase. Our model problem is to find the unknown concentration of the solute C that satisfies the following stochastic ADR equation:

$$\phi(\underline{x})\frac{\partial C}{\partial t} = \nabla\cdot(\mathbf{D}(\underline{x})\nabla C) - \nabla\cdot(\underline{q}(\underline{x})C) + R(\underline{x}, C) + \sigma G(C)\frac{\partial W}{\partial t}, \quad (1)$$

where $(\underline{x}, t) \in \Omega \times [0, T]$. We discuss the stochastic term below, but note for now that when $\sigma = 0$, we have the standard deterministic model. Here we take Ω to be an open domain of $\mathbb{R}^3$ and solve over a finite time

interval $[0, T]$. ϕ is the porosity (void fraction) of the rock, and $\mathbf{D}$ is the symmetric dispersion tensor, which models small-scale mixing and spreading of solutes. R is a reaction function, which models the chemical interaction between different solutes and/or between solutes and pore space. A simple model for $R(C)$ is the classical Langmuir sorption isotherm $R(C) = (\lambda\gamma C) / (1 + \lambda C)$. The formulation for $\mathbf{D}$ is given further below. The velocity $\underline{\mathrm{q}}$ is given by Darcy's law as

$$\underline{\mathrm{q}} = -\frac{\mathbf{k}(\underline{\mathrm{x}})}{\mu}\nabla p, \tag{2}$$

where p is the fluid pressure, μ is the fluid viscosity and $\mathbf{k}$ the permeability tensor of the porous medium. For simplicity, we take $\mathbf{k}$ to be

$$\mathbf{k} = \begin{pmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_3 \end{pmatrix} \tag{3}$$

with $k_1 > 0, k_2 > 0, k_3 > 0$. Assuming that rock and fluids are incompressible and sources or sinks are absent, mass conservation is given by $\nabla \cdot \underline{\mathrm{q}} = 0$. From this we can formulate the elliptic pressure equation, which allows us to compute the pressure field in the porous medium

$$\nabla \cdot \left[\frac{\mathbf{k}(\underline{\mathrm{x}})}{\mu}\nabla p\right] = 0. \tag{4}$$

For $\sigma \neq 0$, we have a stochastic ADR which should be interpreted as an integral equation and here we understand the stochastic integral in the Ito sense. We use the representation of the Wiener process $W(x, t)$ as

$$W(x, t) = \sum_{i,j,k\in\mathbb{N}} \sqrt{\nu_{i,j,k}}\, e_{ijk}(x)\beta_{ijk}(t), \tag{5}$$

see [10, 29], where $(\nu_{ijk}, e_{ijk})_{i,j,k\in\mathbb{N}}$ are eigenvalues and eigenfunctions for a given covariance operator Q that we take here to be exponential with correlation lengths (ℓ_1, ℓ_2, ℓ_3) in the x-, y- and z-directions. The β_{ijk} are independent and identically distributed standard Brownian motions and so the noise is white in time. We take $G(C) = C(1 - C)$ as phenomenological form of noise, it has the property that concentration remains in $[0, 1]$ and variations from noise are small for C close to either 0 or 1. Interpretation of the stochastic integral in the Ito sense from a modelling view can be thought of that the pore-scale process acts on a time scale that is faster than the reaction time scales.

3 Finite-volume space discretisation and exponential integrators schemes for ADR

3.1 Space discretisation

As in [35], we use the classical finite volume method with a structured mesh $\mathcal{T}$. First, we solve the pressure equation (Eq. 4) and then obtain the velocity field from Eq. 2. This provides the integral of the velocity $\{q_{i,j}\}_{i\in\mathcal{T}}$ at each edge j of a control volume i. Integrating Eq. 1 over i using the divergence theorem and the flux approximations, we obtain the following equation:

$$\phi_i V_i \frac{\mathrm{d}C_i(t)}{\mathrm{d}t} = -\sum_{j}^{\text{edges of i}} \left[F_{i,j}(t) + q_{i,j}C_j(t)\right] + V_i\, R(C_i(t)) + \sigma V_i\, G(C_i(t))\frac{\mathrm{d}W_i(t)}{\mathrm{d}t} \quad \forall \quad i \in \mathcal{T}. \tag{6}$$

Here, $C_i(t)$ is the approximation of C at time t at the centre of the control volume $i \in \mathcal{T}$, $F_{i,j}(t)$ is the approximation of the diffusive flux at time t at edge j and $q_{i,j}C_i(t)$ is the approximation of the advective flux at time t at edge j. Note that $F_{i,j}(t)$ already contains C_j. $V_i\, R(C_i(t))$ is the approximation of the integral of the reaction term over the ith control volume of volume V_i and ϕ_i is the mean value of the porosity ϕ in the control volume i. We apply standard upwind weighting to the flux term $q_{i,j}C_j$. For $\sigma \neq 0$, the stochastic forcing term is treated in a similar way to the reaction term. We let h denote the maximum mesh size and use this to indicate our spatial discretisation. We can rewrite Eq. 6 in the standard way as the following non-linear system of equations for all control volumes $i \in \mathcal{T}$

$$\frac{\mathrm{d}\underline{\mathrm{C}}_h(t)}{\mathrm{d}t} = \mathbf{L}\underline{\mathrm{C}}_h(t) + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h, t) + \sigma\underline{\mathrm{G}}(\underline{\mathrm{C}}_h)\frac{\mathrm{d}\underline{\mathrm{W}}_h(t)}{\mathrm{d}t}, \quad t \in [0, T]. \tag{7}$$

Here $\mathbf{L}$ is the stiffness matrix coming from the approximations of the advective and diffusive fluxes, $\underline{\mathrm{C}}_h(t)$ is the concentration vector at all control volumes at time t and the term $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h, t)$ comes from the boundary conditions and reaction term. For the case of stochastic forcing when $\sigma \neq 0$, we project $W(x, t)$ onto the centre of the finite volume by P_h (see [22, 23, 34]) so that $\underline{\mathrm{W}}_h(t) = P_h \sum_{i,j,k,\in[1,\ldots,J]} \sqrt{\nu_{ijk}}\, e_{ijk}(x)\beta_{ijk}(t)$, and we have also projected onto a finite set of eigenfunctions e_{ijk} of the covariance operator Q.

3.2 Time discretisations

First, we discuss the implementation for the deterministic system when $\sigma = 0$. The standard time-stepping schemes commonly used for the ADR are the semi-implicit Euler and implicit Euler schemes. In [35], we showed, for several 2D applications, that the exponential time differencing scheme of order $\mathcal{O}(\Delta t)$ (ETD1) requires at least ten times less computational cost compared to implicit time integrators to reduce the numerical error to a certain value. Here we only use the semi-implicit scheme for comparisons. Recall that given the initial data $\underline{\mathrm{C}}_h^0 = \underline{\mathrm{C}}^0$ and a sequence of time steps τ_n, the semi-implicit Euler scheme for Eq. 7 is

$$\frac{\underline{\mathrm{C}}_h^{n+1} - \underline{\mathrm{C}}_h^n}{\tau_n} = \mathbf{L}\underline{\mathrm{C}}_h^{n+1} + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_n), \qquad \tau_n = t_{n+1} - t_n. \tag{8}$$

We introduce the ETD1 scheme for the ADR (Eq. 1) using the variation of constants. This allows us to write the exact solution of Eq. 7 as

$$\underline{\mathrm{C}}_h(t_n) = e^{t_n \mathbf{L}}\underline{\mathrm{C}}^0 + \int_0^{t_n} e^{(t_n - s)\mathbf{L}}\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(s), s)\mathrm{d}s, \quad t_n \in [0, T].$$

We can construct the corresponding solution at t_{n+1} as

$$\underline{\mathrm{C}}_h(t_{n+1}) = e^{\tau_n \mathbf{L}}\underline{\mathrm{C}}_h(t_n) + \int_0^{\tau_n} e^{(\tau_n - s)\mathbf{L}}\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n + s), t_n + s)\mathrm{d}s. \tag{9}$$

Note that the expression in Eq. 9 is still an exact solution. The idea behind exponential time differencing is to approximate $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n + s), t_n + s)$ by a suitable polynomial [9, 19]. We consider the simplest case where $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n + s), t_n + s)$ is approximated by the constant $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n), t_n)$. The corresponding ETD1 scheme is given by

$$\underline{\mathrm{C}}_h^{n+1} = e^{\tau_n \mathbf{L}}\underline{\mathrm{C}}_h^n + \tau_n \varphi_1(\tau_n \mathbf{L})\underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_n) \tag{10}$$

where $\varphi_1(\mathbf{G}) = \mathbf{G}^{-1}\left(e^{\mathbf{G}} - \mathbf{I}\right) = \left(e^{\mathbf{G}} - \mathbf{I}\right)\mathbf{G}^{-1}$ for any invertible matrix $\mathbf{G}$ and $\varphi_1(\mathbf{G}) = \sum_{i=1}^{\infty} \mathbf{G}^{i-1}/i!$ when $\mathbf{G}$ is singular. We also consider the EEM scheme [7], also called the exponential Rosenbrock–Euler method [15], which is of order $\mathcal{O}(\Delta t^2)$ for non-autonomous problems. The scheme uses the φ_1 function

$$\underline{\mathrm{C}}_h^{n+1} = \underline{\mathrm{C}}_h^n + \tau_n\, \varphi_1\left(\tau_n\, (\mathbf{L} + \partial_C \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_{n+1/2}))\right) \times (\mathbf{L}\underline{\mathrm{C}}_h^n + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_{n+1/2})), \tag{11}$$

where $t_{n+1/2} = (t_{n+1} + t_n)/2$. In our deterministic simulations, we used the EEM scheme as the reference solution, i.e. treat it as the "exact" solution when comparing the ETD1 and semi-implicit schemes. Throughout the simulations, we use a constant time step $\tau_n = \Delta t$. We note that the EEM and ETD1 schemes are equivalent for linear advection diffusion problems, and both give the same solution up to the tolerance specified for the evaluation of the exponential function φ_1.

For the stochastic ADR when $\sigma \neq 0$, we apply the stochastic exponential time differencing method, introduced in [21] for a Fourier–Galerkin approximation in space and developed further in [17, 18, 20]. Convergence for finite element discretisation and implementation for finite volumes is considered in [22] for a semi-implicit method and exponential integration in [23, 24]. These methods are based on the mild solution [10, 29] of the stochastic ADR (Eq. 1) and the system of stochastic differential equations Eq. 7. Given the solution at time t_n, the solution at time t_{n+1} is given by

$$\underline{\mathrm{C}}_h(t_{n+1}) = e^{\Delta t\, \mathbf{L}}\underline{\mathrm{C}}_h(t_n) + \int_{t_n}^{t_{n+1}} e^{(t_{n+1} - s)\, \mathbf{L}}\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(s), s)\mathrm{d}s + \int_{t_n}^{t_{n+1}} e^{(t_{n+1} - s)\, \mathbf{L}}\underline{\mathrm{G}}(\underline{\mathrm{C}}_h(s))\mathrm{d}W(s).$$

A numerical scheme may be constructed approximating $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(s), s)$ by $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n), t_n)$ and $e^{(t_{n+1} - s)\, \mathbf{L}}\underline{\mathrm{G}}(\underline{\mathrm{C}}_h(s))$ by $e^{\Delta t \mathbf{L}}\underline{\mathrm{G}}(\underline{\mathrm{C}}_h(t_n))$. This leads to

$$\underline{\mathrm{C}}_h^{n+1} = e^{\Delta t\, \mathbf{L}}\underline{\mathrm{C}}_h^n + \Delta t \varphi_1(\Delta t\, \mathbf{L})\underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_n) + e^{\Delta t\, \mathbf{L}}\underline{\mathrm{G}}(\underline{\mathrm{C}}_h^n)\Delta W_h^n \tag{12}$$

where $\Delta W_h^n = P_h \sum_{i \in [1, \ldots, J]^3} \sqrt{\nu_i} e_i(x) \Delta \beta_i^n$ and $\Delta \beta^n = \beta(t_n + 1) - \beta(t_n)$ are standard Brownian increments, independent identically distributed $\mathcal{N}(0, \Delta t)$ random variables. Convergence of the scheme is proved in [24].

In the absence of chemical reactions, i.e. $\gamma = 0$, and sub-grid heterogeneities, i.e. $\sigma = 0$, the ADR (Eq. 1) is solved exactly within the given tolerance of $\varepsilon = 10^{-6}$ of the EEM and ETD1 scheme, which give identical results in this case.

4 Efficient computation of the action of φ_1

It is well-known that a standard Padé approximation for a matrix exponential is not an efficient method for large-scale problems [4, 27, 32]. Hence, we focus on the real fast Léja point and the Krylov subspace techniques to evaluate the action of the exponential matrix function $\varphi_1(\Delta t\, \mathbf{L})$ on a vector $\underline{\mathrm{v}}$, instead of computing the full exponential function $\varphi_1(\Delta t\, \mathbf{L})$ as in a standard Padé approximation. For details on the real fast Léja point technique, see [13, 32], and for the Krylov subspace

technique, see [2, 3, 6]. We give a brief summary below. Note that for the EEM scheme, we need to compute at the n^{th} step the action of $\varphi_1\left(\Delta t\ (\mathbf{L} + \partial_C \underline{\mathbf{N}}(\underline{\mathbf{C}}_h^n, t_{n+1/2}))\right)$ in the same way as the action of $\varphi_1(\Delta t\, \mathbf{L})$.

4.1 Krylov space subspace technique

The Krylov subspace technique approximates the action of the exponential matrix function $\varphi_1(\Delta t \mathbf{L})$ on a vector $\underline{\mathrm{v}}$ by projection onto a small Krylov subspace $K_m = \text{span}\left\{\underline{\mathrm{v}}, \mathbf{L}\underline{\mathrm{v}}, \ldots, \mathbf{L}^{m-1}\underline{\mathrm{v}}\right\}$ [32]. The approximation is formed using an orthonormal basis of $\mathbf{V}_m = \left[\underline{\mathrm{v}}_1, \underline{\mathrm{v}}_2, \ldots, \underline{\mathrm{v}}_m\right]$ of the Krylov subspace K_m and of its completion $\mathbf{V}_{m+1} = \left[\mathbf{V}_m, \underline{\mathrm{v}}_{m+1}\right]$. The basis is found by Arnoldi iteration [12] which uses stabilised Gram–Schmidt to produce a sequence of vectors that span the Krylov subspace. Let $\underline{\mathrm{e}}_i^j$ be the ith standard basis vector of $\mathbb{R}^j$. We approximate $\varphi_1(\Delta t \mathbf{L})\underline{\mathrm{v}}$ by

$$\varphi_1(\Delta t \mathbf{L})\underline{\mathrm{v}} \approx \|\underline{\mathrm{v}}\|_2 \mathbf{V}_{m+1}\varphi_1(\Delta t \overline{\mathbf{H}}_{m+1})\underline{\mathrm{e}}_1^{m+1} \tag{13}$$

with

$$\overline{\mathbf{H}}_{m+1} = \begin{pmatrix} \mathbf{H}_m & \underline{0} \\ 0, \cdots, 0, h_{m+1,m} & 0 \end{pmatrix} \quad \text{where}$$

$$\mathbf{H}_m = \mathbf{V}_m^T \mathbf{L} \mathbf{V}_m = [h_{i,j}].$$

The coefficient $h_{m+1,m}$ is recovered in the last iteration of Arnoldi's iteration [13, 32, 35]. For a small Krylov subspace (i.e. m is small), a standard Padé approximation can be used to form $\varphi_1(\Delta t \overline{\mathbf{H}}_{m+1})$, but an efficient way used in [32] is to recover $\varphi_1(\Delta t \overline{\mathbf{H}}_{m+1})\underline{\mathrm{e}}_1^{m+1}$ directly from the Padé approximation of the exponential of a matrix related to $\mathbf{H}_m$. In our implementation, we use the function `phiv.m` of the Matlab package Expokit [32], which uses the efficient technique specified above. It performs adaptive time stepping to achieve a set tolerance.

4.2 Real fast Léja point technique

For a given vector $\underline{\mathrm{v}}$, real fast Léja points approximate $\varphi_1(\Delta t \mathbf{L})\underline{\mathrm{v}}$ by $P_m(\Delta t \mathbf{L})\underline{\mathrm{v}}$, where P_m is an interpolation polynomial of degree m of φ_1 at the sequence of points $\{\xi_i\}_{i=0}^m$ called spectral real fast Léja points. These points $\{\xi_i\}_{i=0}^m$ belong to the spectral focal interval $[\alpha, \beta]$ of the matrix $\Delta t\mathbf{L}$, i.e. the focal interval of the smaller ellipse containing all the eigenvalues of $\Delta t \mathbf{L}$. This spectral interval can be estimated by the well-known Gershgorin circle theorem [38]. It has been shown that as the degree of the polynomial increases and hence the number of Léja points increases, superlinear convergence is achieved [6], i.e.

$$\lim_{m\to\infty} \|\varphi_1(\Delta t \mathbf{L})\underline{\mathrm{v}} - P_m(\Delta t \mathbf{L})\underline{\mathrm{v}}\|_2^{1/m} = 0, \tag{14}$$

where $\|.\|_2$ is the standard Euclidian norm. For a real interval $[\alpha, \beta]$, a sequence of real fast Léja points $\{\xi_i\}_{i=0}^m$ is defined recursively as follows. Given an initial point ξ_0, usually $\xi_0 = \beta$, the sequence of fast Léja points is generated by

$$\prod_{k=0}^{j-1} |\xi_j - \xi_k| = \max_{\xi\in[\alpha,\beta]} \prod_{k=0}^{j-1} |\ \xi - \xi_k\ | \qquad j = 1, 2, 3, \cdots. \tag{15}$$

We use the Newton's form of the interpolating polynomial P_m given by

$$P_m(z) = \varphi_1\left[\xi_0\right] + \sum_{j=1}^{m} \varphi_i\left[\xi_0, \xi_1, \cdots, \xi_j\right] \prod_{k=0}^{j-1} (z - \xi_k) \tag{16}$$

where the divided differences $\varphi_1[\bullet]$ are defined recursively by

$$\begin{cases} \varphi_i\left[\xi_j\right] = \varphi_i(\xi_j) \\ \varphi_i\left[\xi_j, \xi_{j+1}, \cdots, \xi_k\right] := \dfrac{\varphi_i\left[\xi_{j+1}, \xi_{j+2}, \cdots, \xi_k\right] - \varphi_i\left[\xi_j, \xi_{j+1}, \cdots, \xi_{k-1}\right]}{\xi_k - \xi_j}. \end{cases} \tag{17}$$

An algorithm to compute the action of the exponential matrix function $\varphi_i(\Delta t \mathbf{L})$ on a vector $\underline{\mathrm{v}}$ can be found in [35] where the standard way is used to compute the divided differences. Due to cancelation errors, this standard way cannot produce accurate divided differences with magnitude smaller than machine precision. Here we used the techniques described in [5, 25] to compute the divided differences efficiently and accurately. We use the Taylor expansion with 16 terms during the computation of the finite differences (Eq. 17).

[31] has shown that Léja points for the interval $[-2, 2]$ assure optimal accuracy; thus, for the spectral focal interval $[\alpha, \beta]$ of the matrix $\Delta t\mathbf{L}$, it is convenient to interpolate, by a change of variables, the function $\varphi_i(c + \gamma\xi)$ of the independent variable $\xi \in [-2, 2]$ with $c = (\alpha + \beta)/2$ and $\gamma = (\beta - \alpha)/4$. It can be shown [25]

that the divided differences of a function $f(c+\gamma\xi)$ of the independent variable ξ at the points $\{\xi_i\}_{i=0}^m \subset [-2,2]$ are the first column of the matrix function $f(\mathbf{L}_m)$, where

$$\mathbf{L}_m = c\mathbf{I}_{m+1} + \gamma\widehat{\mathbf{L}}_m, \quad \widehat{\mathbf{L}}_m = \begin{pmatrix} \xi_0 & & & & \\ 1 & \xi_1 & & & \\ & 1 & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & \xi_m \end{pmatrix}$$

From this, the divided differences of $\varphi_i(c+\gamma\xi)$ of the independent variable $\xi \in [-2,2]$ at the points $\{\xi_i\}_{i=0}^m \subset [-2,2]$ is $\varphi_1(\mathbf{L}_m)e_1^{m+1}$ where e_1^{m+1} is the first standard basis vector of $\mathbb{R}^{m+1}$. Taylor expansion of order $p=16$ with scaling and squaring is used in [5, 25] to compute $\varphi_1(\mathbf{L}_m)e_1^{m+1}$. In practice, the real fast Léja points is computed once in the interval $[-2,2]$ and reused at each time step during the computation of the divided differences. We use the efficient algorithm of [2] to compute the real fast Léja points in $[-2,2]$.

5 Numerical examples

To analyse the efficiency of the ETD1 and EEM schemes compared to standard semi-implicit time integrators and demonstrate that they can be used to solve an ADR (Eq. 1) with stochastic forcing, we consider three different examples. We use the upper 20 layers (Examples 1 and 3) and upper 40 layers (Example 2) of the highly heterogeneous SPE 10 benchmark case (Fig. 1), which represents a braided fluvial North Sea oil field with seven orders of magnitude permeability variation [8]. We consider two cases for dispersion. In all examples, we take a uniform dispersion tensor of $\mathbf{D} = 10^{-6} \times \mathbf{I}_3$. In Example 1, we additionally take $\mathbf{D}$ as function of the velocity $\mathbf{q}$

$$\mathbf{D} = \alpha_T \|\mathbf{q}\|\delta_{i,j} + (\alpha_L - \alpha_T)q_i q_j/\|\mathbf{q}\|, \quad 1 \le i,j \le 3, \quad (18)$$

where α_T and α_L are the longitudinal and the transverse dispersivity, respectively. In Example 3, we also include the stochastic forcing term with $\sigma \in \{0, 5\times10^{-4}, 10^{-3}\}$, i.e. we increase the influence of the noise term from simulation to simulation, to model sub-grid fluctuations in velocity and reaction rate.

All our tests were performed on a workstation with a 3-GHz Intel processor and 8 GB RAM. Our code was implemented in Matlab 7.10. In contrast to our earlier 2D simulations [35], we now use large time steps, very large Péclet numbers and an iterative solver for linear systems. Further, in contrast to the simulations presented in [7] for the EEM scheme used in conjunction with a 2D finite element method, we use the Krylov subspace technique and large Péclet number flows. The resulting domain is $\Omega = [0, L_1] \times [0, L_2] \times [0, L_3]$, the finite volume mesh $\mathcal{T}$ has a spatial discretisation $\Delta x = 20$ ft, $\Delta y = 10$ ft and $\Delta z = 2$ ft. Dimensions are $L_1 = 1{,}200$ ft, $L_2 = 2{,}200$ ft and $L_3 = 20$ ft in Examples 1 and 3 and $L_3 = 40$ ft in Example 2.

The matrix $\mathbf{L}$ in Eq. 7 is sparse with size $264{,}000 \times 264{,}000$ and 1,810,400 non-zero elements for Example 1 and $528{,}000 \times 528{,}000$ with 3,647,200 non-zero elements for Example 2. For semi-implicit time integration, this linear systems is solved at each time step with a variant of the iterative Krylov solver, the bi-conjugate gradients stabilised method as implemented in Matlab. We use $\varepsilon = 10^{-6}$ as the absolute tolerance

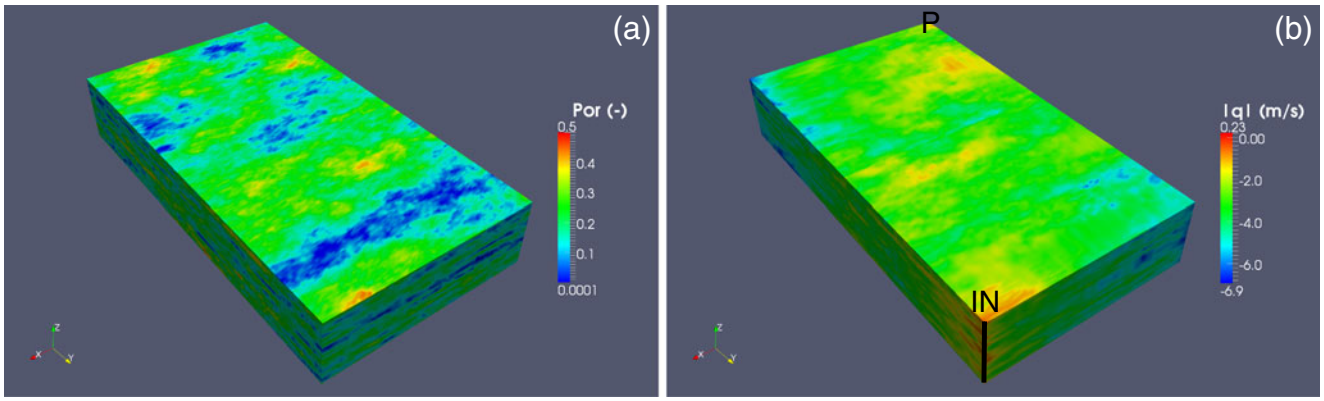

Fig. 1 Porosity field (**a**) for the first upper 20 layers of the SPE 10 model [8] and simulated velocity field, shown by the magnitude of the velocity vector, i.e. $\log_{10}(\|\mathbf{q}\|)$ (**b**). The injector (*IN*) is located along the vertical edge of the *lower left model corner*; the producer (*P*) is located diagonally opposite along the vertical edge of the *upper right model corner*. Note that the vertical height is exaggerated tenfold

error and $m = 8$ for the Krylov subspace dimension. To our knowledge, there is no rigorous theory that allows us to predict the optimal value for m a priori. For small m, a penalty can arise from an increase in the number of iterations necessary to achieve a given tolerance, especially if Δt is large, but less time is spent in the orthogonalisation process and the required memory is lower. When m increases, the total number of iterations decreases, but a penalty occurs due to the additional time spent in the orthogonalisation process and the corresponding increase in memory requirements.

For pressure and concentration, we take the Dirichlet boundary condition

$$\Gamma_D = \{\{0\} \times \{0\} \times [0, L_3]\} \cup \{\{L_1\} \times \{L_2\} \times [0, L_3]\},$$

and homogenous Neumann boundary conditions elsewhere such that

$$p = \begin{cases} 3{,}998.96 \text{ psi} & \text{in} \quad \{0\} \times \{0\} \times [0, L_3] \\ 7{,}997.92 \text{ psi} & \text{in} \quad \{L_1\} \times \{L_2\} \times [0, L_3] \end{cases}$$
$$-k\,\nabla p(x,t) \cdot \underline{\mathrm{n}} = 0 \quad \text{in} \quad \Gamma_N = \partial\Omega \backslash \Gamma_D.$$

This models a fixed-pressure injector and producer pair located at two diagonally opposite edges of the model, i.e. at $\{0\} \times \{0\} \times [0, L_3]$ and $\{L_1\} \times \{L_2\} \times [0, L_3]$, respectively.

For the concentration, we take

$$C = 0 \quad \text{in} \quad \{\{0\} \times \{0\} \times [0, L_3]\} \times [0, T]$$
$$C = 1 \quad \text{in} \quad \{\{L_1\} \times \{L_2\} \times [0, L_3]\} \times [0, T]$$
$$-(\mathbf{D}\nabla C)(x,t) \cdot \mathbf{n} = 0 \quad \text{in} \quad \Gamma_N \times [0, T]$$
$$C_0 = 0 \quad \text{in} \quad \Omega \quad \text{(initial solution)}$$

where $\mathbf{n}$ is the unit outward normal vector to Γ_N. For the reaction function, we use the Langmuir sorption isotherm with $\lambda = 1$, $\gamma = 10^{-3}$ in Examples 1 and 2 and $\gamma = 0$ or $\gamma = 10^{-4}$ in Example 3. In other words, we model reactions in Example 3 either by explicitly defining a reaction term and adding noise to it or just by stochastic noise. The dynamic viscosity μ is 0.3 cp in all cases.

In the legends of Fig. 2, "Krylov ETD1" denotes results from the ETD1 scheme with the Krylov subspace technique, "Léja ETD1" denotes results from the ETD1 scheme with the real fast Léja point technique, "Semi-implicit" denotes results from the standard semi-implicit scheme and "Krylov EEM" denotes results from EEM scheme the Krylov subspace technique.

We set a tolerance of 10^{-6} for the Krylov, Léja point methods as well as for the iterative solver for the standard semi-implicit scheme. This ensures errors are from the discretisation rather than from the linear algebra. We discuss varying the tolerance further below.

5.1 Example 1

We compare the concentration fields at time $T = 4{,}096$ s for the first 20 upper layers of the SPE 10 model for the solution of the ADR with a spatially homogeneous dispersion tensor and with chemical reactions

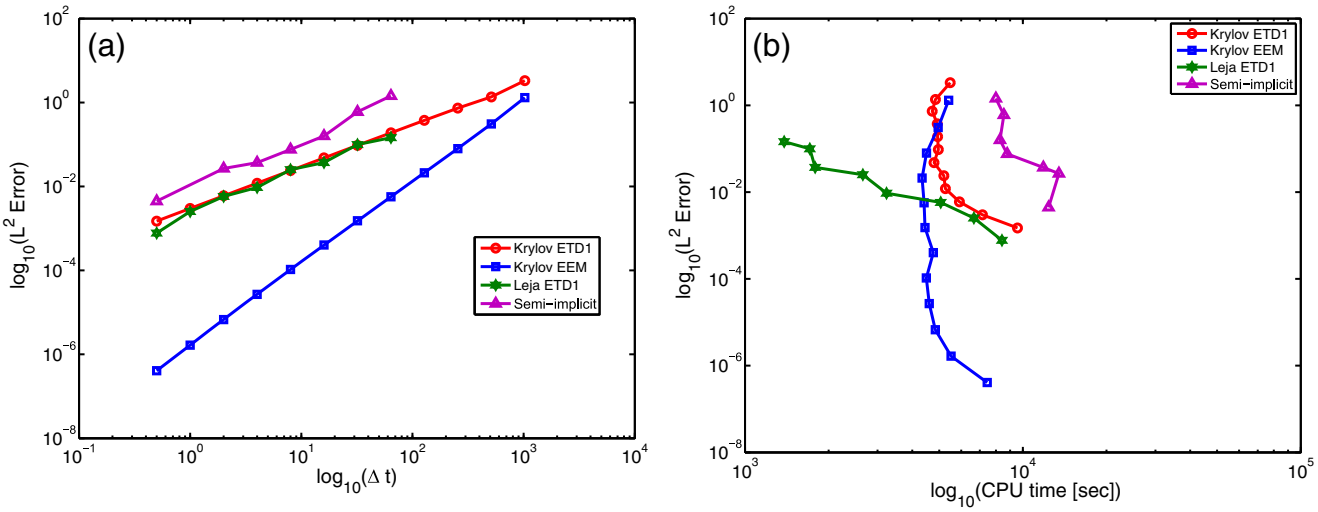

Fig. 2 L^2 error as a function of size of the time step (**a**) and CPU time as a function of the L^2 error (**b**) for Example 1. The maximum Péclet number is 1.7×10^6. The ADR Eq. 1 was solved with a spatially homogeneous dispersion tensor and with chemical reactions represented by a Langmuir adsorption isotherm

modelled by a Langmuir adsorption isotherm with $\gamma = 10^{-3}$. Figure 2a shows the corresponding L^2 error as a function of the size of the time step. It demonstrates that the ETD1 scheme is more accurate compared to the standard semi-implicit method. It further shows that the EEM scheme is also more accurate compared to the semi-implicit and ETD1 schemes. All three time integrators, semi-implicit, ETD1 and EEM, exhibit convergence that is in agreement with theory: The temporal order of the semi-implicit and ETD1 scheme is $\mathcal{O}(\Delta t)$ while the temporal order of the EEM scheme is $\mathcal{O}(\Delta t^2)$; hence, the smallest error for a given time-step size is obtained with the EEM scheme. Figure 2b demonstrates the efficiency of ETD1 and EEM schemes compared to the standard semi-implicit method by plotting the L^2 error as a function of the CPU time. It shows that the real Léja point technique is computationally more efficient than the Krylov subspace technique in the ETD1 scheme because the L^2 error is less for the same CPU time. A slight improvement in efficiency can be obtained with the EEM scheme compared to the ETD1 scheme with the Krylov subspace technique. Note that the near-vertical CPU time for both Krylov subspace methods implies that the L^2 error does not change as a function of CPU time when smaller (or larger) time-step sizes increase computational cost; there is an optimum time-step size for which CPU time and L^2 error are smallest. This is explained by the fact that each time step is subdivided into smaller sub-steps to reach the given fixed tolerance ε in the function `phiv.m` of the package Expokit [32], providing a limit for the efficiency of this method (see also Fig. 3b). We also observed that the CPU time for the ETD1 scheme with the real Léja technique appears to be proportional to the size of the time step for the given fixed tolerance. This indicates that the best efficiency is reached for large time-step sizes. Generally, these results imply that the best efficiency of the ETD1 and EEM schemes is obtained for the largest time steps, although at the cost of accuracy.

Figure 3 shows the results for the same 20 layers of the SPE 10 model when solving the ADR with a spatially heterogeneous dispersion tensor (Eq. 18) and with chemical reactions modelled by a Langmuir adsorption isotherm with $\gamma = 10^{-3}$. Figure 3a depicts the L^2 error as a function of the size of the time step, and Fig. 3b depicts the cross-plot of L^2 error versus CPU time. This indicates that, like in Fig. 2, the EEM scheme is more accurate than the ETD1 and semi-implicit schemes as it is of order $\mathcal{O}(\Delta t^2)$. However, for this case, the EEM scheme is significantly less efficient than the ETD1 scheme with the Léja point technique. Likewise, the ETD1 scheme with the Krylov subspace technique is less efficient than the Léja point technique. In both Krylov subspace methods, the L^2 error is independent of the CPU time, i.e. spending higher computational costs by varying the time-step size does not increase the solution accuracy. It hence appears that a heterogeneous and anisotropic dispersion tensor reduces the efficiency of the ETD1 and EEM schemes. However, both schemes are still more accurate than the semi-implicit time integrator. As explained above, this is due

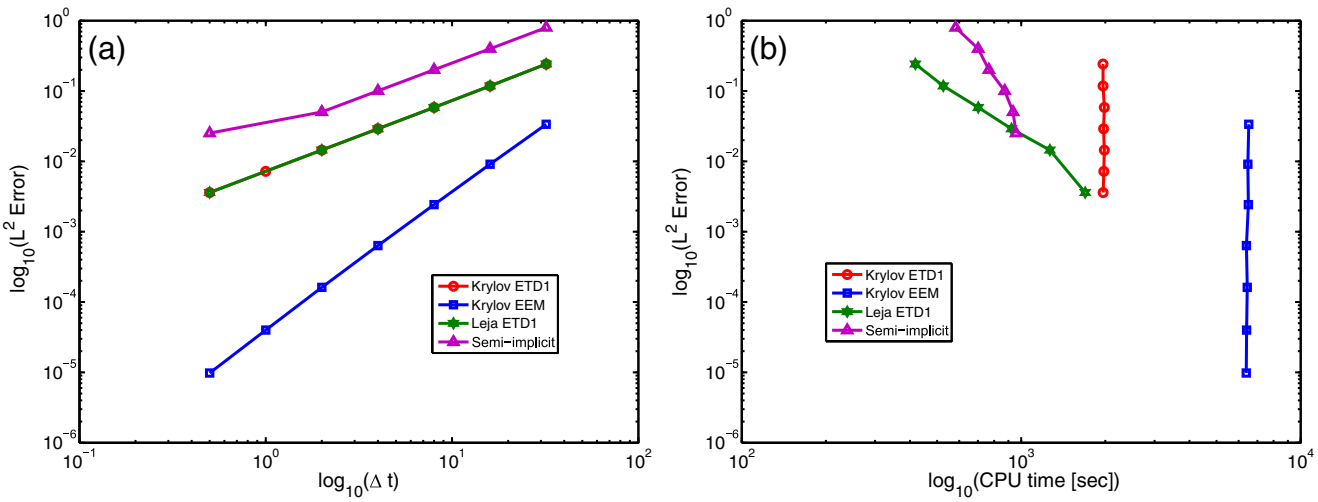

Fig. 3 L^2 error as a function of size of the time step (**a**) and CPU time as a function of the L^2 error (**b**) for Example 1. The maximum Péclet number is 2.4×10^6. The ADR Eq. 1 was solved with a spatially heterogeneous dispersion tensor (see Eq. 18) and with chemical reactions represented by a Langmuir adsorption isotherm with $\gamma = 10^{-3}$

to the time step being subdivided into smaller sub-steps to reach a given tolerance ε in the function `phiv.m` of the package Expokit [32]. The ETD1 method with the Léja point technique still shows scalability, i.e. the CPU time decreases with decreasing size of the time step for a fixed small tolerance.

5.2 Example 2

We consider the concentration field at time $T = 4{,}096$ s for the first 40 upper layers of the SPE 10 model for the solution of the ADR with a spatially homogeneous dispersion tensor and chemical reactions modelled by a Langmuir adsorption isotherm with $\gamma = 10^{-3}$. As before, ETD1 and semi-implicit schemes are of order $\mathcal{O}(\Delta t)$ while the EEM scheme is of $\mathcal{O}(\Delta t^2)$ (Fig. 4a). Like in the previous two examples, Fig. 4b shows that the subdivision of the time step into smaller sub-steps in the function `phiv.m` of the package Expokit [32] to reach our given tolerance reduces the efficiency of the Krylov subspace-based techniques EEM and ETD1 for large time steps. Further analysis [34] also showed that the ETD1 scheme with the Léja point technique is only more efficient than the standard semi-implicit scheme for large time steps. For smaller time steps, the ETD1 scheme with the Léja point technique becomes less efficient because more Léja points are needed for the interpolation. This is probably due to the lack of accuracy in the computation of the divided differences (Eq. 17). We can recover the efficiency by adding more terms in the Taylor expansion during the computation of the finite differences (Eq. 17) or by scaling the matrix $\mathbf{L}_m$ to smaller entries when using in the algorithm of [7] to compute the finite differences. Scaling the matrix $\mathbf{L}_m$ renders the "squaring" procedure less efficient, but as the finite differences are computed only once per time step, this will not increase the overall computational cost greatly.

5.3 Example 3

Figure 5 shows simulated concentration fields in the upper 20 layers of the SPE 10 model [8] after $T = 32{,}768$ s for a single realization of the noise path. We used the Krylov subspace technique to evaluate the matrix exponentials in Eq. 12. We took as eigenfunctions e_{ijk} of Q the eigenfunctions of the Laplacian with periodic boundary conditions on Ω and correlation in space corresponding to correlation lengths in x, y and z of $\ell_1 = 24$, $\ell_2 = 60$ and $\ell_3 = 40$. In Fig. 5, the strength of the stochastic forcing term increases from top to bottom, so that $\sigma = 0$ (a, d), $\sigma = 5 \times 10^{-4}$ (b, e) and $\sigma = 10^{-3}$ (c, f). The strength of the Langmuir adsorption reaction term increases from left to right, i.e. $\gamma = 0$ (a, b, c) and $\gamma = 10^{-4}$ (d, e, f). We observe, as expected, that the increase in reaction rate retards breakthrough of the solute. The concentration fronts become significantly sharper, i.e. less dispersed, for the form of stochastic forcing we have taken to model subgrid heterogeneities. Sharp concentration fronts are clearly seen as we increase the intensity of the stochastic forcing (e.g. (c) and (f)). This can be explained by

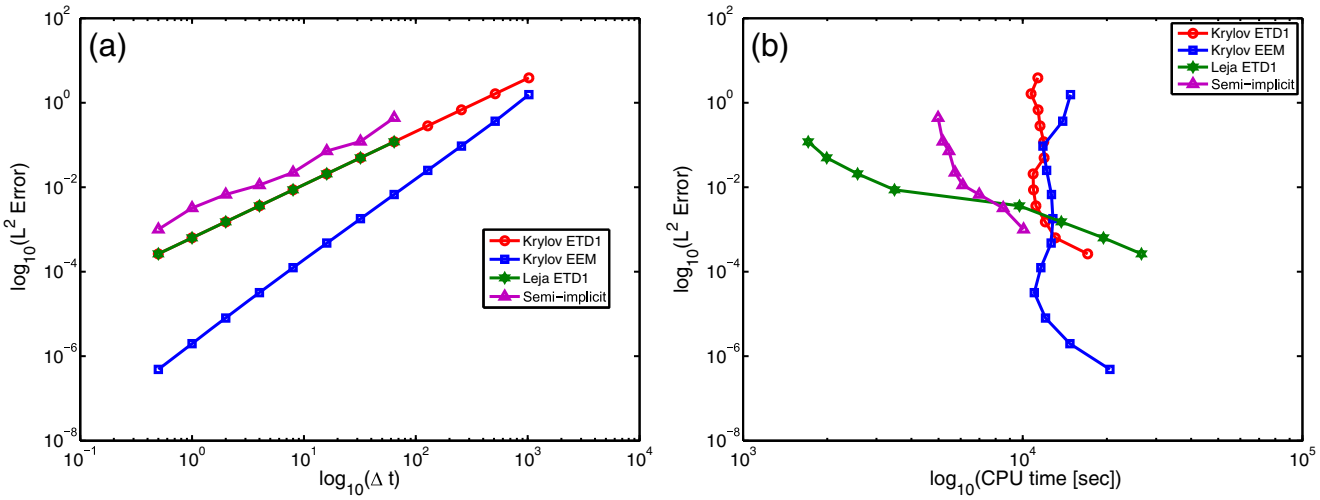

Fig. 4 L^2 error as a function of size of the time step (**a**) and CPU time as a function of the L^2 error (**b**) for Example 2. The maximum Péclet number is 3.2×10^6. The ADR Eq. 1 was solved with a spatially homogeneous dispersion tensor and with chemical reactions represented by a Langmuir adsorption isotherm with $\gamma = 10^{-3}$

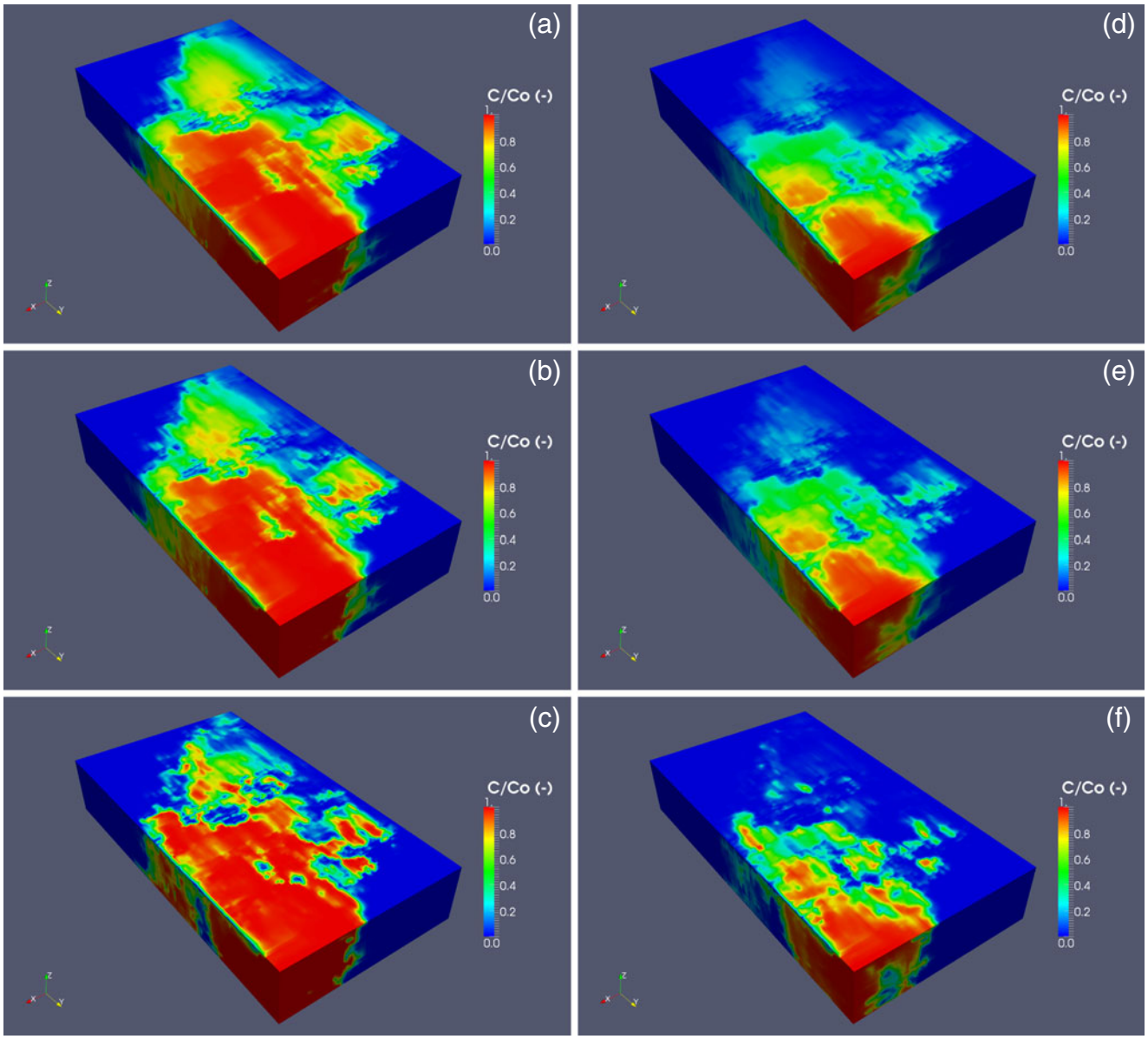

Fig. 5 Simulated concentration fields in the upper 20 layers of the SPE 10 model [8] after $T = 32{,}768$ s. The strength of the stochastic forcing term increases from top to bottom, i.e. $\sigma = 0$ (**a**, **d**), $\sigma = 5 \times 10^{-4}$ (**b**, **e**) and $\sigma = 10^{-3}$ (**c**, **f**). The strength of the reaction term increases from *left* to *right*, i.e. $\gamma = 0$ (**a**–**c**) and $\gamma = 10^{-4}$ (**d**–**f**). An increase in reaction rate retards breakthrough of the solute. An increase in stochastic forcing causes sharper fronts because, on average, more solutes react at intermediate concentrations. Note that the vertical height is exaggerated tenfold

our choice of the phenomenological model for sub-grid heterogeneity in our stochastic forcing term: It retards breakthrough of low concentrations because, on average, more solutes react at intermediate concentrations; this also promotes longer range higher concentrations because once concentrations have reached a critical threshold, they react more rapidly. As a consequence, for larger noise intensity, we find less dispersed regions of higher concentrations at further distances from the point of injector compared to lower noise or zero noise intensity.

6 Conclusions

We have applied several exponential time integrators to solve a non-linear advection–dispersion–reaction problem in highly heterogeneous 3D porous media

based on the SPE 10 test case [8]. We considered a deterministic case where the reaction term is prescribed explicitly and a stochastic case where we model sub-grid variations in transport velocities and chemical reaction rates using additive noise. We have presented and analysed two exponential time integrators of order $\mathcal{O}(\Delta t)$ (ETD1 scheme) and order $\mathcal{O}(\Delta t^2)$ (EEM scheme) for the deterministic case. Here we have computed the matrix exponential using either real fast Léja point techniques (ETD1) or with a Krylov subspace technique (ETD1 and EEM). In the stochastic case, we have implemented a stochastic exponential time differencing scheme of order $\mathcal{O}(\Delta t^{1/2})$. The spatial discretisation was achieved by standard upwind-weighted finite volume method, and the resulting matrices contained several 100,000 unknowns. For the applied boundary conditions, solute transport is dominated by advection (Péclet numbers larger than 10^6). Our analysis showed that the first- and second-order exponential time integration schemes perform well compared to the standard first-order semi-implicit scheme in terms of efficiency and accuracy in the deterministic case. Exponential time integrators are further well-suited to solve advection–dispersion–reaction problems with stochastic noise, which is an emergent technology to model uncertain and fluctuating sub-grid physics during porous media flow [36, 37, 39, 40]. The results we have presented have been for a fixed small tolerance on the linear algebra. For large step sizes, this leads to a reduced performance of the ETD1 and EEM schemes with the Krylov subspace technique. This is caused by the function `phi.m` of the Expokit package [32], which creates internal time subdivisions to satisfy a given tolerance. If a large time step is chosen, this subdivision can dominate the calculation and increase the CPU time, rendering the Krylov-based techniques less efficient. For large step sizes, such as we consider here, we can consider varying the tolerance with the step size. In Fig. 6, we vary the tolerance for the Krylov method so that $\varepsilon = 10^{-2}\Delta t$ and recompute the errors and CPU time for the ETD1 method in Example 2. For comparison, we also show the Léja point calculation from Fig. 3 with a fixed tolerance of $\varepsilon = 10^{-6}$. It now seems that the Krylov-based method is competitive with the Léja point method. However, with larger tolerances in the linear algebra, we note that some of the solutions take non-physical values of concentration, i.e. slightly negative or slightly larger than one. These are not present if the tolerance in the Krylov method is fixed and small, i.e. $\varepsilon = 10^{-6}$. Given the small difference in efficiency, we observe in Fig. 6 that this suggests a preference in the Léja point implementation here.

In general, our results suggest that exponential integrators can readily be applied to typical 3D reservoir models comprising tens to hundreds of thousands grid cells to simulate solute mixing and chemical reactions where sub-grid processes are represented by stochastic forcing.

Acknowledgements We thank the U.K. Engineering and Physical Research Science Council, Grant EP/H025022/1, for financial support.

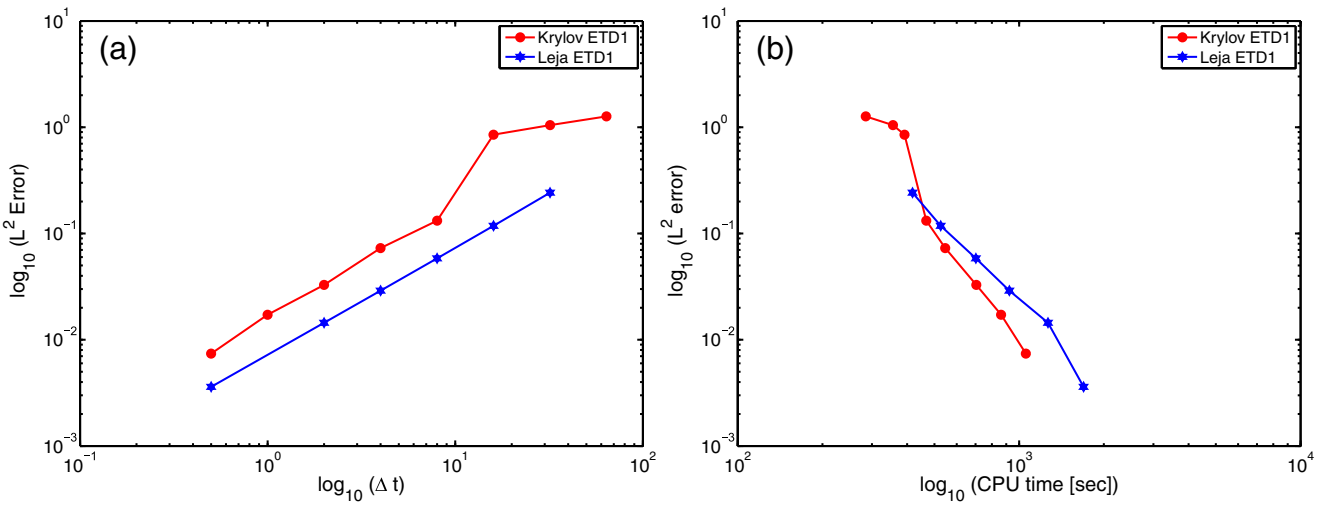

Fig. 6 Comparison of the Krylov method with a variable tolerance of $\varepsilon = 10^{-2}\Delta t$ and Léja method with a fixed tolerance of $\varepsilon = 10^{-6}$ for the ETD1 scheme used in Example 1. The Krylov method is more efficient compared to Fig. 3 where the tolerance for was fixed but produced non-physical results, i.e. concentrations slightly below zero or slightly above one due to the larger tolerances in the linear algebra

References

1. Austad, T., Strand, S., Madland, M.V., Puntervold, T., Korsnes, R.I.: Seawater in chalk: An EOR and compaction fluid. SPE Reserv. Evalu. Eng. **11**(4), 648–654 (2008)
2. Baglama, J., Calvetti, D., Reichel, L.: Fast Leja points. Electron. Trans. Numer. Anal. **7**, 124–140 (1998)
3. Bergamaschi, L., Caliari, M., Vianello, M.: The RELPM exponential integrator for FE discretizations of advection–diffusion equations. Computational Science—ICCS Proceedings, pp. 434–442 (2004)
4. Berland, H., Skaflestad, B., Wright, W.: A Matlab package for exponential integrators. ACM Trans. Math. Softw. **33**(1), Article No. 4 (2007)
5. Caliari, M.: Accurate evaluation of divided differences for polynomial interpolation of exponential propagators. Computing **80**(2), 189–201 (2007)
6. Caliari, M., Vianello, M., Bergamaschi, L.: Interpolating discrete advection diffusion propagators at Leja sequences. J. Comput. Appl. Math. **172**(1), 79–99 (2004)
7. Caliari, M., Vianello, M., Bergamaschi, L.: The LEM exponential integrator for advection–diffusion–reaction equations. J. Comput. Appl. Math. **210**(1–2), 56–63 (2007)
8. Christie, M.A., Blunt, M.J.: Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE Reserv. Evalu. Eng. **4**(4), 308–317 (2001)
9. Cox, S.N., Matthews, P.C.: Exponential time differencing for stiff systems. J. Comput. Phys. **176**(2), 430–455 (2002)
10. Da Prato, G., Zabczyk, J.: Stochastic Equations in Infinite Dimensions. Encyclopedia of Mathematics and its Applications, vol. 44. Cambridge University Press, Cambridge (1992)
11. Dentz, M., Le Borgne, T., Englert, A., Bijeljic, B.: Mixing, spreading, and reactions in heterogeneous media: A brief review. J. Contam. Hydrol. **120–121**, 1–17 (2011). doi:10.1016/j.jconhyd.2010.05.002
12. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins University Press, Baltimore (1996)
13. Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **34**(5), 1911–1925 (1997)
14. Hochbruck, M., Lubich, C. and Selhofer, H.: Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. **19**(5), 1552–11574 (1998)
15. Hochbruck, M., Ostermann, A. and Schweitzer, J.: Exponential Rosenbrock-type methods. SIAM J. Numer. Anal. **47**(1), 786–803 (2009y)
16. Hochbruck, M., Ostermann, A.: Exponential integrators. Acta Numer. **19**, 209–286 (2010)
17. Jentzen, A.: Pathwise numerical approximations of SPDEs. Potential Anal. **31**(4), 375–404 (2009)
18. Jentzen, A., Kloeden, P.E.: Overcoming the order barrier in the numerical approximation of SPDEs with additive space–time noise. Proc. R. Soc. A **465**(2102), 649–667 (2009)
19. Kassam, K.A., Trefethen, L.N.: Fourth-order time stepping for stiff PDEs. SIAM J. Comput. **26**(4), 1214–1233 (2005)
20. Kloeden, P.E., Lord, G.J., Neuenkirch, A., Shardlow, T.: The exponential integrator scheme for stochastic partial differential equations: Pathwise error bounds. J. Comput. Appl. Math. **235**, 1245–1260 (2011). doi:10.1016/j.cam.2010.08.011
21. Lord, G.J., Rougemont, J.: A numerical scheme for stochastic PDEs with Gevrey regularity. IMA J. Numer. Anal. **24**(4), 587–604 (2004). doi:10.1093/imanum/24.4.587
22. Lord, G.J., Tambue, A.: A modified semi-implicit Euler–Maruyama scheme for finite element discretization of SPDEs. ArXiv:1004.1998v1 (2010)
23. Lord, G.J., Tambue, A.: Stochastic exponential integrators for a finite element discretization of SPDEs. ArXiv:1005.5315v1 (2010)
24. Lord, G.J., Tambue, A.: Stochastic exponential integrators for finite element discretization of SPDEs for multiplicative and additive noise. ArXiv:1103.1986v2 (2010)
25. McCurdy, A., Ng, K.C., Parlett, B.N.: Accurate computation of divided differences of the exponential function. Math. Comput. **43**(186), 501–528 (1984)
26. Minchev, B., Wright, W.: A review of exponential integrators for first order semi-linear problems. Preprint 2/2005, Norwegian University of Science and Technology, Trondheim Norway. http://www.math.ntnu.no/preprint/numerics/2005/N2-2005.pdf (2005)
27. Moler, C., Van Loan, C.F.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Rev. **45**(1), 3–49 (2003)
28. Niesen, J. and Wright, W.: A Krylov subspace algorithm for evaluating the $\phi-$functions appearing in exponential integrators. ACM Trans. Math. Softw. (2012, in press)
29. Prévôt, C., Röckner, M.: A Concise Course on Stochastic Partial Differential Equations. Springer, New York (2007). ISBN-10: 3540707808
30. Pruess, K., Xu, T.F., Apps, J., Garcia, J.: Numerical modeling of aquifer disposal of CO_2. SPE J. **8**(1), 49–60 (2003)
31. Reichel, L.: Newton interpolation at Leja points. BIT **30**(2), 332–346 (1990)
32. Sidje, R.B.: Expokit: A software package for computing matrix exponentials. ACM Trans. Math. Softw. **24**(1), 130–156 (1998)
33. Sorbie, K.S., MacKay, E.J.: Mixing of injected, connate and aquifer brines in waterflooding and its relevance to oilfield scaling. J. Pet. Sci. Eng. **27**(1–2), 85–106 (2000)
34. Tambue, A.: Efficient numerical methods for porous media flow. Ph.D. thesis, Department of Mathematics, Heriot-Watt University (2010)
35. Tambue, A., Lord, G.J., Geiger, S.: An exponential integrator for advection-dominated reactive transport in heterogeneous porous media. J. Comput. Phys. **229**(10), 3957–3969 (2010)
36. Tartakovsky, A.M.: Langevin model for reactive transport in porous media. Phys. Rev. E **82**(2), 026,302 (2010). doi:10.1103/PhysRevE.82.026302
37. Tartakovsky, A.M., Tartakovsky, D.M., Meakin, P.: Stochastic Langevin model for flow and transport in porous media. Phys. Rev. Lett. **101**(4), 044,502 (2008). doi:10.1103/PhysRevLett.101.044502
38. Thomas, J.W.: Numerical Partial Differential Equations: Finite Difference Methods. Springer, New York (1995)
39. Tyagi, M., Jenny P. Lunati, I., Tchelepi, H.A.: A Lagrangian, stochastic modeling framework for multiphase flow in porous media. J. Comput. Phys. **227**, 6696–6714 (2008). doi:10.1016/j.jcp.2008.03.030
40. Tyagi, M., Tchelepi, H.A., Jenny, P.: A stochastic trapping model for non-wetting phase ganglias in two phase flow through porous media. In: 11th European Conference on the Mathematics of Oil Recovery. Bergen, Norway (2008)