An exponential integrator for advection-dominated reactive transport in heterogeneous porous media

A. Tambue*,[a], G. J. Lord[b], S. Geiger[c]

[a]*Department of Mathematics and the Maxwell Institute for Mathematical Sciences, Heriot Watt University, Edinburgh EH14 4AS, U.K.*
[b]*Department of Mathematics and the Maxwell Institute for Mathematical Sciences, Heriot Watt University, Edinburgh EH14 4AS, U.K.*
[c]*Institute of Petroleum Engineering and the Edinburgh Collaborative of Subsurface Science and Engineering, Heriot Watt University, Edinburgh EH14 4AS, U.K.*

Abstract

We present an exponential time integrator in conjunction with a finite volume discretisation in space for simulating transport by advection and diffusion including chemical reactions in highly heterogeneous porous media representative of geological reservoirs. These numerical integrators are based on the variation of constants solution and solving the linear system exactly. This is at the expense of computing the exponential of the stiff matrix comprising the finite volume discretisation. Using real Léja points or a Krylov subspace technique to approximate the exponential makes these methods competitive compared to standard finite difference-based time integrators. We observe for a variety of example applications that numerical solutions with exponential methods are generally more accurate and require less computational cost. They hence comprise an efficient and accurate method for simulating

*Corresponding author
Email addresses: `at150@hw.ac.uk` (A. Tambue), `gabriel@ma.hw.ac.uk` (G. J. Lord), `sebastian.geiger@pet.hw.ac.uk` (S. Geiger)

non-linear advection dominated transport in geological formations.

Key words: Exponential integration, Léja points, Krylov subspace, advection-diffusion equation, fast time integrators, porous media

1. Introduction

Advection and diffusion can transport chemically reactive components such as dissolved minerals, colloids, or contaminants, over long distances through the highly heterogeneous porous media comprising geological formations. It is hence a fundamental process in many geo-engineering applications, including oil and gas recovery from hydrocarbon reservoirs, groundwater contamination and sustainable use of groundwater resources, storing greenhouse gases (e.g, CO_2) or radioactive waste in the subsurface, or mining heat from geothermal reservoirs. One of the fundamental challenges is to forecast these processes accurately because the permeability in heterogeneous porous and fractured media typically varies over orders of magnitude in space and possibly time (e.g, [1, 2]). This causes highly variable flow fields where local transport can be dominated entirely by either advection (Péclet number larger than one) or diffusion (Péclet number less than one), leading to macroscopic mixing and "anomalous transport" that is characterised by early breakthrough of solutes or contaminants and long tailing at late time [3]. Chemical reaction rates and equilibrium constants can vary in a similar manner, giving rise to complex mixing-induced reaction patterns at the macro-scale because chemical reactions rates can dominate locally over transport rates or vice versa (e.g, [4, 5, 6]).

Predicting the spatial spreading and mixing of reactive solutes in field applications hence requires the efficient and accurate numerical solution of advection-diffusion-reaction equations (ADR) which resolve the wide range in flow velocities and reaction rates. This is particularly important because the exact spatial distribution of the permeability field and reaction rates is commonly unknown and therefore a large number of simulations must be run to quantify the uncertainty of the transport behaviour [7], for example to forecast the possible arrival of highly toxic contaminants at a groundwater well and design adequate remediation schemes.

The ADR can be 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 (e.g, [8, 9]). 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. Common methods include implicit or adaptive time-stepping (e.g, [12, 13]) and operator splitting techniques (e.g, [10, 11]). Comparatively new methods are streamline-based simulations where transport is computed along the time-of-flight [14, 15], adaptive mesh refinement to focus the computational effort around the moving fronts and resolve them accurately [16], or event-based simulations where only those regions are updated where an event (i.e. chemical reaction or transport) occurs [17, 18].

The family of exponential integrators date back to the 1960's (see [19] and [20] for history and detailed references). These methods are based on approximating the corresponding integral formulation of the non-linear part of the differential equation and solving the linear part exactly and computing the exponential of a matrix. Sidje [24] used the Krylov subspace technique and Padé approximation to solve the linear system of ODEs based on variation of constants. Cox and Matthews [32] developed the family of exponential time differencing methods for solving non-linear stiff ODEs. They present the instability issue for computing non-diagonal matrix exponential functions, the so called φ-functions. Kassam and Trefethen [20] used a fourth order exponential time differencing method and the contour integral technique for computing the matrix exponential functions to solve the Kuramoto-Sivashinsky and Allen-Cahn PDEs in one dimension. Berland et al. [33] used a Padé approximation to compute the matrix exponential of φ-functions and provided a package for exponential integrators which is efficient in one dimension.

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 [36]. However, new developments in real fast Léja points and Krylov subspace techniques for computing functions of the matrix exponential has revived interest in these methods. The real fast Léja points technique is based on matrix interpolation polynomials at spectral Léja sequences [21, 22]. 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 [23, 24].

In two and three dimensions, the real fast Léja points technique [25, 26, 22, 27] and Krylov subspace technique [25, 26] have been used to implement the matrix exponential of φ-functions efficiently in linear advection diffusion equations. The real fast Léja points technique is also used for the exponential Euler-Midpoint integrator scheme for solving non-linear ADRs [28] and for the exponential Rosenbrock-type integrators for solving semi-linear parabolic PDEs [29]. Simulations have been carried out for homogeneous media with constant dispersion tensors, uniform velocity fields, and low Péclet number flows using finite difference methods or finite element methods for spatial discretisations. In contrast to previous work, we consider heterogeneous media, the exponential time differencing method of order one with the finite volume discretisation in space and examine high Péclet number flows.

The aim of this paper is to investigate the exponential time differencing method of order one (ETD1) and compare its performance in terms of efficiency and accuracy to standard semi-implicit and fully implicit schemes for solution of non-linear ADRs in highly heterogeneous porous media with largely varying Péclet number flows, that is situations where transport is locally dominated either by diffusion or advection. We use 2D simulations and finite volume discretisations to demonstrate the efficiency and the accuracy of the exponential scheme ETD1. In the implementation of the ETD1 scheme we also compare the efficiency of the real fast Léja points technique with the Krylov technique for computing matrix exponential.

The paper is organised as follows. In the next section we present the mathematical and numerical formulations of ADR. Then we discuss the exponential time differencing stepping schemes for ADR and implementation of the exponential time differencing of order one (ETD1) using the real fast

Léja points and Krylov space techniques. This is followed by two sets of tests in 2D from homogeneous porous media with exact solutions. This allows us to examine the ETD1 scheme as well as test feasibilty for large systems. We then consider heterogeneous porous media where we take first a deterministic permeability field and then a random permeability field. In these examples we see that the ETD1 method using the real Léja points technique is efficient and competitive compared to standard finite difference time ingegrators. Finally, the discussions and conclusions are given.

2. Mathematical and numerical formulations

2.1. Model problem

Our model problem is to find the unknown concentration of the solute C that satisfies the following advection-diffusion-reaction equation (ADR):

$$\phi(\underline{x})\frac{\partial C}{\partial t} = \nabla \cdot (\mathbf{D}(\underline{x})\nabla C) - \nabla \cdot (\underline{q}(\underline{x})C) + R(\underline{x}, C) \quad (\underline{x}, t) \in \Omega \times [0, T] \,. \tag{1}$$

Here we take Ω to be an open domain of $\mathbb{R}^2$ 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. For simplicity we take $\mathbf{D}$ to be

$$\mathbf{D} = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \tag{2}$$

with $D_1 > 0, D_2 > 0$. The term R is a reaction function. Possible reaction mechanisms can be adsorption, for example as described by a Langmuir isotherm, biodegradation or radioactive decay. The velocity $\underline{q}$ is given by Darcy's law as

$$\underline{q} = -\frac{k(\underline{x})}{\mu}\nabla p, \tag{3}$$

where p is the fluid pressure, μ is the fluid viscosity, and k the permeability of the porous medium. 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

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

2.2. Space discretisation

We use the classical finite volume method with a structured mesh $\mathcal{T}$ [30]. First, we solve the pressure equation (Eq. (4)) and then obtain the velocity field from Eq. (3). 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 used in [30] we obtain the following equation

$$\phi_i V_i \frac{dC_i(t)}{dt} = - \sum_{j}^{\text{edges of i}} \left[F_{i,j}(t) + q_{i,j} C_j(t)) \right] + V_i \, R(C_i(t)) \quad \forall \quad i \in \mathcal{T}. \tag{5}$$

Here, $C_i(t)$ is the approximation of C at time t at the center 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. $V_i \, R(C_i(t))$ is the approximation of the integral of the reaction term over the ith control volume of area V_i and ϕ_i is the mean value of the porosity ϕ in the control volume i. We apply standard upwind weighting [31, 30] to the flux term $q_{i,j}C_j$.

We let h denote the maximum mesh size and use this to indicate our spatial discretisation. We can rewrite Eq. (5) in the standard way [31] as the following non-linear system of equations for all control volumes $i \in \mathcal{T}$

$$\frac{d\underline{\mathrm{C}}_h(t)}{dt} = \mathbf{L}\underline{\mathrm{C}}_h(t) + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h, t), \quad t \in [0, T]\,. \tag{6}$$

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. In section 4, we also examine the effects of putting the approximation of the advective flux in the non-linear term $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h, t)$.

2.3. Standard time discretisation

We briefly describe two standard time-stepping schemes, the implicit Euler scheme and the semi implicit Euler scheme. Later we use these for comparison with the exponential scheme of order one, ETD1. Given the initial data $\underline{\mathrm{C}}_h^0 = \underline{\mathrm{C}}^0$, the implicit Euler scheme for Eq. (6) is

$$\frac{\underline{\mathrm{C}}_h^{n+1} - \underline{\mathrm{C}}_h^n}{\Delta t} = \mathbf{L}\underline{\mathrm{C}}_h^{n+1} + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^{n+1}, t_{n+1}) \tag{7}$$

and the semi implicit scheme is

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

where $\Delta t = t_{n+1} - t_n$ is the fixed time-step. For the implicit Euler method we have to solve a non-linear algebraic equation of the form

$$\underline{\mathrm{f}}(\underline{\mathrm{X}}) = (\mathbf{I} + \Delta t\mathbf{L})\underline{\mathrm{X}} + \Delta t\underline{\mathrm{N}}(\underline{\mathrm{X}}, t_n) - \underline{\mathrm{C}}_h^n = \underline{0}$$

at each time-step. For brevity we denote $\underline{\mathrm{C}}_h^{n+1}$ as $\underline{\mathrm{X}}$. We use Newton's method and a variant of Newton's method designed for semi-linear problems [31]. We solve the linear systems using the standard solver in Matlab$^{\mathrm{TM}}$at each iteration in the exact Newton's method. For the variant of Newton's method, the Jacobian of $\underline{\mathrm{f}}$, $\mathbf{J}(\underline{\mathrm{X}})$, is approximated by its constant linear part so that $\mathbf{J}(\underline{\mathrm{X}}) \approx \mathbf{I} + \Delta t\mathbf{L}$. The corresponding quasi Newton's iteration is then given by

$$\begin{aligned}
\underline{\mathrm{X}}_{k+1} &= \underline{\mathrm{X}}_k - (\mathbf{I} + \Delta t\mathbf{L})^{-1}\, \underline{\mathrm{f}}(\underline{\mathrm{X}}_k) \\
&= (\mathbf{I} + \Delta t\mathbf{L})^{-1}\, (\Delta t\underline{\mathrm{N}}(\underline{\mathrm{X}}_k, t_n) - \underline{\mathrm{C}}_h^n).
\end{aligned}$$

This is equivalent to a fixed point method to solve the equivalent equation $(\mathbf{I} + \Delta t\mathbf{L})^{-1}\, \underline{\mathrm{f}}(\underline{\mathrm{X}}) = \underline{0}$. The approximation of the Jacobian by its constant linear part allows us to compute the matrix factorisation only once and to reuse this at each time-step. In the quasi-exact Newton's method and the semi implicit Euler scheme we solve the linear systems using either an LU-decomposition or the standard solver in Matlab$^{\mathrm{TM}}$.

3. Exponential Time Differencing scheme of order one for ADR

3.1. Review of the Exponential Time Differencing methods

We introduce the exponential time differencing stepping scheme of order one (ETD1) for the ADR problem (Eq. (1)) using the variation of constants. This allows us to write the exact solution of Eq. (6) as

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

where s is the integration time. Then, given the exact solution at the time t_n, we can construct the corresponding solution at t_{n+1} as

$$\underline{\mathrm{C}}_h(t_{n+1}) = e^{\Delta t\mathbf{L}}\underline{\mathrm{C}}_h(t_n) + e^{\Delta t\mathbf{L}} \int_0^{\Delta t} e^{-s\mathbf{L}}\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n + s), t_n + s)ds. \qquad (9)$$

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 [32, 20]. We consider the simplest case where $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n + s), t_n + s)$ is approximated by the constant $\underline{\mathrm{N}}(\underline{\mathrm{C}}_h(t_n), t_n)$ and for simplicity consider a constant time-step $\Delta t = t_{n+1} - t_n$. The corresponding ETD1 scheme is given by

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

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 $\mathbf{G}$.

Note that the ETD1 scheme in Eq. (10) can be rewritten as

$$\underline{\mathrm{C}}_h^{n+1} = \underline{\mathrm{C}}_h^n + \Delta t \varphi_1(\Delta t \mathbf{L})(\mathbf{L}\underline{\mathrm{C}}_h^n + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^n, t_n)). \tag{11}$$

This new expression has the advantage that it is computationally more efficient as only one matrix exponential function needs to be evaluated at each step.

3.2. 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 [24, 33, 36]. Here we focus on the real fast Léja points 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. The details of the real fast Léja points technique and [23, 24] for the Krylov subspace technique are given in [27, 21, 22]. We give a brief summary below.

3.2.1. Real fast Léja points 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 [34]. It has been shown that as the degree of the polynomial increases and hence the number of Léja points increases, convergence is achieved [27], i.e.

$$\lim_{m\to\infty} \|\varphi_1(\Delta t\mathbf{L})\underline{\mathrm{v}} - P_m(\Delta t\,\mathbf{L})\underline{\mathrm{v}}\|_2 = 0, \tag{12}$$

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

$$\prod_{k=0}^{j-1} |\xi_j - \xi_k| = \max_{\xi\in[\alpha,\beta]} \prod_{k=0}^{j-1} \mid \xi - \xi_k \mid \quad j = 1, 2, 3, \cdots . \tag{13}$$

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

$$P_m(z) = \varphi_1\,[\xi_0] + \sum_{j=1}^{m} \varphi_1\,[\xi_0, \xi_1, \cdots, \xi_j] \prod_{k=0}^{j-1} (z - \xi_k) \tag{14}$$

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

$$\begin{cases} \varphi_1\,[\xi_j] = \varphi_1(\xi_j) \\ \\ \varphi_1\,[\xi_j, \xi_{j+1}, \cdots, \xi_k] := \dfrac{\varphi_1\,[\xi_{j+1}, \xi_{j+2}, \cdots, \xi_k] - \varphi_1\,[\xi_j, \xi_{j+1}, \cdots, \xi_{k-1}]}{\xi_k - \xi_j}. \end{cases} \tag{15}$$

Algorithm 1 : Compute $\varphi_1(\Delta t\,\mathbf{L})\underline{\mathrm{v}}$ with real fast Léja points. Error e_m is controlled to a presrcribed tolerance *tol* so that $e_m^{\text{Léja}} < tol$.

```
Input: $\mathbf{L}, \underline{\mathrm{v}}, \Delta t, tol, z$ {matrix, vector, time-step, tolerance, number of Léja points to be generated }
$[\alpha, \beta]$ = getfocal($\mathbf{L}$) {get the focal interval using the Gershgorin circle theorem [34]}
$\xi$ = getLeja($\alpha, \beta, z$) {generate $z$ fast Léja points from (Eq. (13)).}
$d_0 = \varphi_1(\xi_0)$.
$\underline{\mathrm{w}}_0 = \underline{\mathrm{v}}$, $\underline{\mathrm{p}}_0 = d_0\underline{\mathrm{w}}_0$, $m = 0$ {initialisation}
while $e_m^{\text{Léja}} = |d_m| \times \|\underline{\mathrm{w}}_m\|_2 > tol$ do
    $\underline{\mathrm{w}}_{m+1} = (\Delta t\mathbf{L} - \xi_m\mathbf{I})\underline{\mathrm{w}}_m$
    $m = m + 1$
    $d_m = \varphi_1(\xi_m)$
    for $i = 1, \cdots, m$ do
        $d_m = \dfrac{d_m - d_{i-1}}{\xi_m - \xi_{i-1}}$ {compute the next divided difference $d_m$}
    end for
    $\underline{\mathrm{p}}_m = \underline{\mathrm{p}}_{m-1} + d_m\underline{\mathrm{w}}_m$
end while
Output: $\underline{\mathrm{p}}_m$
```

We summarise in Algorithm 1 the steps for computing $\varphi_1(\Delta t\, \mathbf{L})\underline{\mathrm{v}}$. In our implementation we estimate the focal interval for $\mathbf{L}$ only once and precompute a sufficiently large number z of Léja points using the efficient algorithm of Baglama et al. [21] for a focal interval of $\Delta t\mathbf{L}$.

The data is passed as input parameters during each call of the algorithm and scaled by Δt. We observed the same convergence problems as described by Caliari et al. [27], that is problems arising from round-off errors during the computation of the divided differences (Eq. (15)) and from the large capacity of the spectral focal interval $[\alpha, \beta]$. We were able to resolve this issue by reducing the time-step size or by using an algorithm for minimising rounding errors from the divided differences [35] when computing Eq. (15). Note that although it is advised in [27] to compute the divided differences in quadruple precision we did not find this necessary.

3.2.2. Krylov space subspace technique

The main idea of the Krylov subspace technique is to approximate 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 $K_m = \text{span}\left\{\underline{\mathrm{v}}, \mathbf{L}\underline{\mathrm{v}}, \ldots, \mathbf{L}^{m-1}\underline{\mathrm{v}}\right\}$ [24]. The approximation is formed using an orthonormal basis of $\mathbf{V}_m = [\underline{\mathrm{v}}_1, \underline{\mathrm{v}}_2, \ldots, \underline{\mathrm{v}}_m]$ 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 [37] which uses stabilised Gram-Schmidt to produce a sequence of vectors that span the Krylov subspace (see Algorithm 2).

Let $\underline{\mathrm{e}}_i^j$ be the i^{th} standard basis vector of $\mathbb{R}^j$. We approximate $\varphi_1(\Delta t\, \mathbf{L})\underline{\mathrm{v}}$

by

$$\varphi_1(\Delta t\,\mathbf{L})\underline{\mathrm{v}} \;\approx\; \|\underline{\mathrm{v}}\|_2\mathbf{V}_{m+1}\varphi_1(\Delta t\,\overline{\mathbf{H}}_{m+1})\underline{\mathrm{e}}_1^{m+1} \qquad (16)$$

with

$$\overline{\mathbf{H}}_{m+1} = \begin{pmatrix} \mathbf{H}_m & \underline{0} \\ 0,\cdots,0,h_{m+1,m} & 0 \end{pmatrix} \qquad \text{where} \qquad \mathbf{H}_m = \mathbf{V}_m^T\mathbf{L}\mathbf{V}_m = [h_{i,j}].$$

The coefficient $h_{m+1,m}$ is recovered in the last iteration of Arnoldi's iteration.

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 a efficient way used in [24] 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 $\mathbf{H}_m$ [24]. In our implementation we use the function `phiv.m` of the package Expokit [24], which allows us to compute the forward ETD1 solution using the previous solution while controlling the local error at each iteration for a given tolerance. The function `phiv.m` takes the time step Δt, the matrix $\mathbf{L}$, the vectors $\underline{\mathrm{u}}$ and $\underline{\mathrm{v}}$, the dimension of the Krylov subspace m, and the desired tolerance as the input and provides $\underline{\mathrm{u}} + \Delta t\varphi_1(\Delta t\,\mathbf{L})(\mathbf{L}\underline{\mathrm{u}} + \underline{\mathrm{v}})$ as the output. This method is accurate for a symmetric matrix with negative eigenvalues but can be less efficient on very large non-symmetric matrices [23, 24].

4. Numerical experiments

To analyse the convergence and efficiency of the ETD1 method for solving ADRs, we apply it to a variety of porous media flow problems and compare it to our standard time-stepping methods implicit Euler and semi implicit schemes introduced in Section 2.3. We consider the following four problems:

Algorithm 2 : Arnoldi's algorithm

```
Initialise: $\underline{v}_1 = \frac{\underline{v}}{||\underline{v}||_2}$ {normalisation}
for $j = 1 \cdots m$ do
    $\underline{w} = \mathbf{L}\underline{v}_j$
    for $i = 1 \cdots j$ do
        $h_{i,j} = \underline{w}^T \underline{v}_i$ {compute inner product to build elements of the matrix $\mathbf{H}$}
        $\underline{w} = \underline{w} - h_{i,j}\underline{v}_i$ {Gram–Schmidt process}
    end for
    $h_{j+1,j} = ||\underline{w}||_2$
    $\underline{v}_{j+1} = \frac{\underline{w}}{||\underline{w}||_2}$ {normalisation}
end for
```

1. A linear ADR without reaction term, a heterogeneous dispersion tensor, and a non-uniform velocity field representing moderate Péclet number flows, for which an analytical solution exists [38].

2. A non-linear ADR in homogeneous media where transport is controlled equally by advection and diffusion (i.e, Péclet number is 1) for which an analytical solution exists [28].

3. A non-linear ADR for a deterministic permeability field representing a highly idealised fractured porous media. Here transport is entirely dominated by advection (high Péclet number flow).

4. A non-linear ADR for a stochastically generated permeability field where transport is locally dominated by either advection or diffusion.

In the two latter applications we use the classical Langmuir isotherm to model the sorption of the transported species onto the rock surface, i.e.

$$R(C) = \frac{\lambda \beta C}{1 + \lambda C}.$$

The parameter λ is an adsorption constant and β the maximum amount of the solute that can be adsorbed. We take $\lambda = \beta = 1$ in this work.

For the sake of simplicity we assume that the porosity ϕ is constant in all applications. In all cases we take our domain to be rectangular $\Omega = [0, L_1) \times [0, L_2)$ but use both, uniform and non-uniform, rectangular meshes. The time has been normalised by the average flow rate and the domain length in the direction of flow such that the mean of the concentration has traveled through the entire domain at $T = 1$. In each application example, the matrix $\mathbf{L}$ is pentadiagonal. For a grid size $N_x \times N_y$, the corresponding matrix has the size $N_x N_y \times N_x N_y$ with $5 \times N_x N_y - 2 \times N_x - 6$ non-zero elements.

For pressure, we take the Dirichlet boundary $\Gamma_D^1 = \{0, L_1\} \times [0, L_2]$ and Neumann boundary $\Gamma_N^1 = (0, L_1) \times \{0, L_2\}$ such that

$$\begin{aligned} p &= \begin{cases} 1 & \text{in} \quad \{0\} \times [0, L_2] \\ \\ 0 & \text{in} \quad \{L_1\} \times [0, L_2] \end{cases} \\ -k\, \nabla p(x,t) \cdot \underline{\mathrm{n}} &= 0 \quad \text{in} \quad \Gamma_N^1. \end{aligned}$$

For concentration, we take the Dirichlet boundary $\Gamma_D = \{0\} \times [0, L_2]$ and Neumann boundary $\Gamma_N = \{(0, L_1] \times \{0, L_2\}\} \cup \{\{L_1\} \times [0, L_2]\}$ such that

$$\begin{aligned} C &= 1 \quad \text{in} \quad \Gamma_D \times [0, T] \\ -(\mathbf{D}\nabla C)(x,t) \cdot \underline{\mathrm{n}} &= 0 \quad \text{in} \quad \Gamma_N \times [0, T] \\ C_0 &= 0 \quad \text{in} \quad \Omega \quad \text{(initial solution)} \end{aligned}$$

where $\underline{\mathrm{n}}$ is the unit outward normal vector to Γ_N (or Γ_N^1).

We report below the local or grid Péclet number $\mathrm{Pe}_{\mathrm{loc}} = \max_i \mathrm{Pe}_i$ where Pe_i is computed over each control volume as

$$\mathrm{Pe}_i := \frac{\max\limits_{j \text{ edge of } i} |q_{i,j}|}{\|\mathbf{D}_i\|_\infty}.$$

$\mathbf{D}_i$ is the mean value of the diffusion matrix over the control volume i and $q_{i,j}$ is the integral of the velocity over the edge j for the control volume i.

For applications where we do not have an analytic solution we estimate the global error by

$$\|C_h(t) - C_h^{\Delta t}(t)\|_{L^2(\Omega)} \approx 2\|C_h^{\Delta t}(t) - C_h^{\Delta t/2}(t)\|_{L^2(\Omega)}.$$

where $C_h^{\Delta t}(t)$ is the approximation of the solution at time t found with time-step Δt. Unless explicitly stated, the tolerance used for Newton's method and the ETD1 schemes is 10^{-6} and the Krylov space dimension used is $m = 6$. The tests were performed on a standard PC with a 3 GHz processor and 2GB RAM. Our code was implemented in Matlab 7.7. In the legends of all of our graphs we use the following notation

- "Implicit with Newton" denotes results from the implicit Euler with standard Newton method.

- “Implicit with Newton V” denotes results from the implicit Euler with the variant of Newton method.

- “Léja ETD1” denotes results from ETD1 with real fast Léja points for matrix exponential.

- “Krylov ETD1” denotes results from ETD1 with Krylov subspace for matrix exponential.

- “Semi implicit” denotes results from the semi-implicit scheme.

4.1. Homogeneous porous media without reaction term

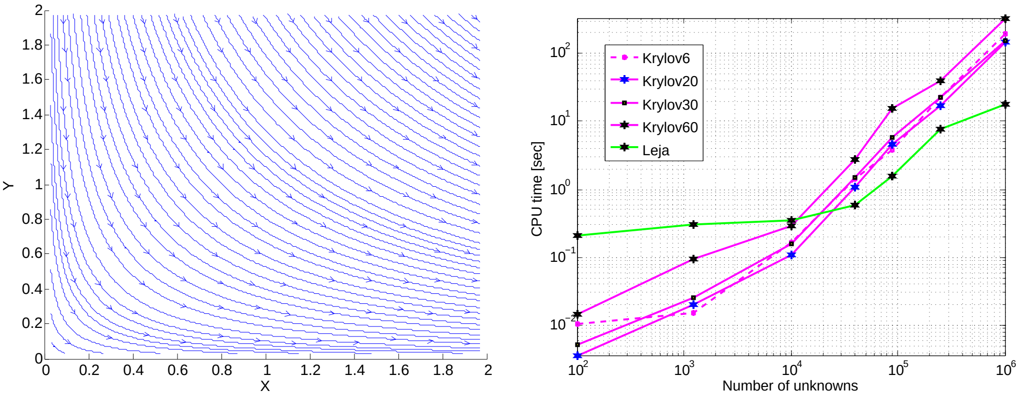
We use this problem to examine the scaling of the ETD1 method for problems with different numbers of unknowns and analyse the convergence in space by comparing it to an exact solution [38]. Since the ADR does not contain a reaction term, the problem is linear. The domain is defined as $\Omega = [L_0, L_1) \times [L_0, L_1)$, $L_0 = 0.01$, $L_1 = 2$. The initial time is given as $t_0 = 0.01$. This is necessary because the exact solution is not defined at the origin and at $t = 0$. The dispersion tensor $\mathbf{D}$ is heterogeneous and its coefficients are given by

$$\begin{cases} D_1(x,y) = D_0 u_0^2 x^2 \quad (x,y) & \in \Omega \\ D_1(x,y) = D_0 u_0^2 y^2 \quad (x,y) & \in \Omega. \end{cases}$$

The velocity field (Fig. 1(a)) is given explicitly by

$$\begin{cases} \underset{\sim}{q} = (q_x, q_y)^T \\ q_x(x,y) = u_0 x \quad (x,y) & \in \overline{\Omega} \\ q_y(x,y) = -u_0 y \quad (x,y) & \in \overline{\Omega} \end{cases} \tag{17}$$

where $D_0 = 0.1$ and $u_0 = 2$. The local Péclet number ranges from 21 to 2 as the grid is refined. Initial and boundary conditions are taken according to the exact solution [38], assuming an instantaneous release at a point (x_0, y_0), $x_0 = 1.5$, $y_0 = 1.5$. We take a fixed time-step of $\Delta t = 1/3000$.



(a) (b)

Figure 1: Numerical examples for the linear advection-diffusion problem in homogeneous porous media given in [38] (a) shows the streamlines, (b) shows the CPU time as a function of number of unknowns required to evaluate the expression $\varphi_1(\Delta t\mathbf{L})(\mathbf{L}\underline{\mathrm{C}}_h^0 + \underline{\mathrm{N}}(\underline{\mathrm{C}}_h^0, T_0))$. A standard PC with a 3 GHz processor and 2GB RAM was used for the simulations. The number in the four Krylov curves in (b) denotes the dimension m of the subspace taken.

Figure 1(a) shows the streamlines which indicate direction of flow. Figure 1(b) shows the CPU time needed to compute single time-step using ETD1 with real Léja points and Krylov techniques as a function of the number of unknowns. The number of the real fast Léja points used to achieve the given tolerance are 6 for 100 unknowns, and increases to 69 as the grid is refined.

In Figure 1(b) we show that good values for the dimension of the Krylov subspace are $m = 20$ and $m = 6$, but $m = 20$ appears to be a slightly better value for this specific example. To our knowledge, there is no rigorous theory that allows us to predict the optimal value for m a priori. For example, the default value used in [24] is $m = 30$ but we observe that this is not the optimal value for our specific example. When m increases, the total number of iterations decreases but a penalty occurs due to the additional time spent in the orthogonalisation process in Algorithm 2 and the corresponding increase in memory requirements. 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. Since the memory on the PC used in this work is limited to 2 GB, the values of Δt in our application examples are generally small, and we require to compute the action of the matrix exponential function φ_1 on a vector to reach the final time T over 3000 times, we have chosen $m = 6$ as the optimal value for the Krylov subspace dimension in all our applications.

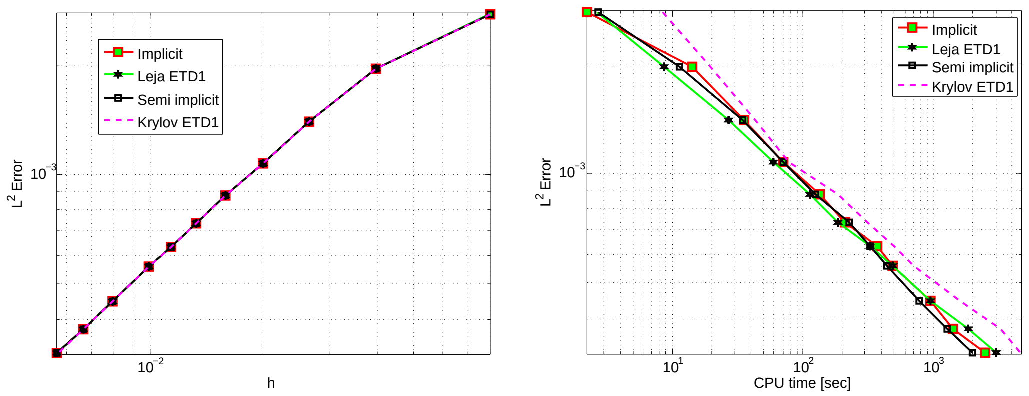
For 10^4 and more unknowns, that is for problem sizes that become representative for real reservoir simulations, the computation of the matrix exponential with real fast Léja points is more efficient than the Krylov technique by a factor of approximately 10, regardless of the Krylov subspace dimension m. Similar results were obtained by [25, 26] for constant dispersion tensor, constant velocity, and low Péclet number flows. Once the matrix size is greater or equal to 10^4, the CPU time increases linearly with the number of unknowns (Figure 1(b)). The time to evaluate a matrix with 10^6 unknowns using 69 real Léja points is 18 seconds. These results suggest that the ETD1 is a scalable solver and is hence probably applicable to large-scale problems with several million of unknowns that are encountered in 3D reservoir simulations.

Figure 2(a) shows a convergence of order $\mathcal{O}(h)$ for the spatial discretisation with fixed time step $\Delta t = 1/3000$. The error in the L^2 norm is computed at time $T = 1$. Figure 2(b) shows the L^2 error as a function CPU time, which is depicted in Figure 2(a).

The efficiency for solving this linear ADR problem is roughly similar for all methods, that is approximately the same computational cost is required to reduce the numerical error by a certain increment. Although Figure 1(b) indicates that for small number of unknowns the Krylov technique requires significantly less computational effort than the real Léja point method to compute one step with one vector $\underline{\mathrm{v}}$, Figure 2(b) shows that over the course of an entire simulation, which involves many individual time-steps, the local error control reduces this efficiency, therefore Krylov and Léja point methods are comparable. We recall that the Krylov subspace implementation is known to be efficient for symmetric matrices. Here we observe good convergence even for highly nonsymmetric matrices $\mathbf{L}$.

4.2. Homogeneous porous media with a non-linear reaction term

We now evaluate the ETD1 method for a non-linear ADR problem where the non-linear reaction term is given by $R(C) = -\gamma C^2(1 - C)$. We take $\gamma = 100$, use a constant velocity of $\underline{\mathrm{q}} = [-0.01, -0.01]^T$, and the dispersion tensor has the entries $D_1 = D_2 = 10^{-4}$. The domain is $\Omega = [0, 1) \times [0, 1)$, which we discretise with $h = \Delta x = \Delta y = 10^{-2}$. The local Péclet number for the flow is 1, that is transport is controlled equally by advection and diffusion. The initial condition and boundary conditions are defined with



(a)

(b)

Figure 2: (a) Convergence of the L^2 norm at $T = 1$ as a function of the grid size h. (b) The L^2 norm at $T = 1$ as a function of CPU time. Both plots are for the linear ADR in homogeneous porous media without a reaction term (Problem 1) with fixed $\Delta t = 1/3000$. Recall here that the Krylov subspace dimension is fixed to be $m = 6$.

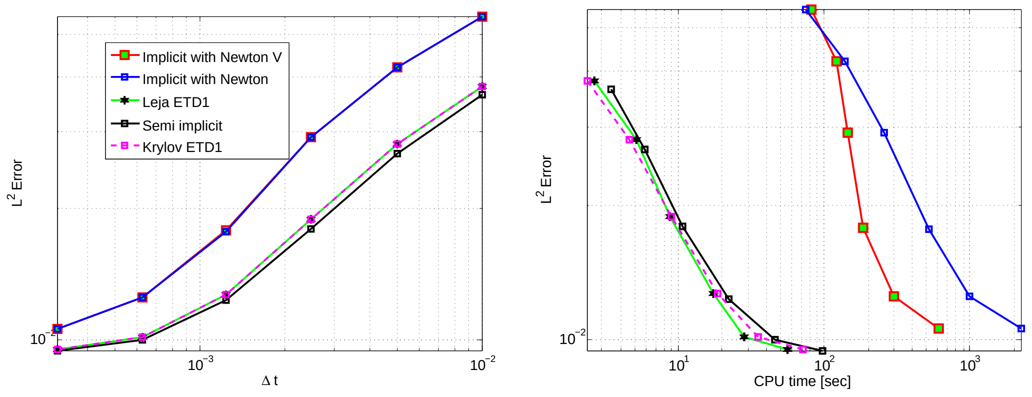
respect to the exact solution [28] given by

$$C(x, y, t) = (1 + \exp\left(a(x + y - bt) + a(b - 1)\right))^{-1} \qquad (18)$$

where $a = \sqrt{\gamma/\left(4 \times 10^{-4}\right)}$ and $b = -0.02 + \sqrt{\gamma \times 10^{-4}}$.

Figure 3(a) shows the convergence as a function of the chosen time-step Δt, measuring the error at the final time $T = 1$. The semi-implicit time-stepping method and the ETD1 methods have similar error constants. All schemes have the same rate of convergence $\mathcal{O}(\Delta t)$.

Figure 3(b) shows the L^2 error as a function of CPU time, which is given in Figure 3(a). Again, the computational effort to reduce the numerical by a certain error is approximately equivalent for both, Léja and Krylov subspace techniques. They are also similar to a semi-implicit time integrator. However, all three methods, ETD1 with Léja points and Krylov subspace technique and semi-implicit time-stepping, outperform the implicit time-stepping methods. Those require about 10 times more computational costs to obtain the same numerical error. If the advective component of the flux is included in the non-linear part rather than the linear part for the ETD1 scheme, then the error constant worsens. In this case the graph representing the error would lie between that of the ETD1 or semi-implicit error and implicit error in Figure 3(a).



(a)

(b)

Figure 3: (a) Convergence of the L^2 norm at $T = 1$ as a function of Δt. (b) The L^2 norm at $T = 1$ as a function of CPU time. Both are for the the non-linear ADR in homogeneous porous media (Problem 2).

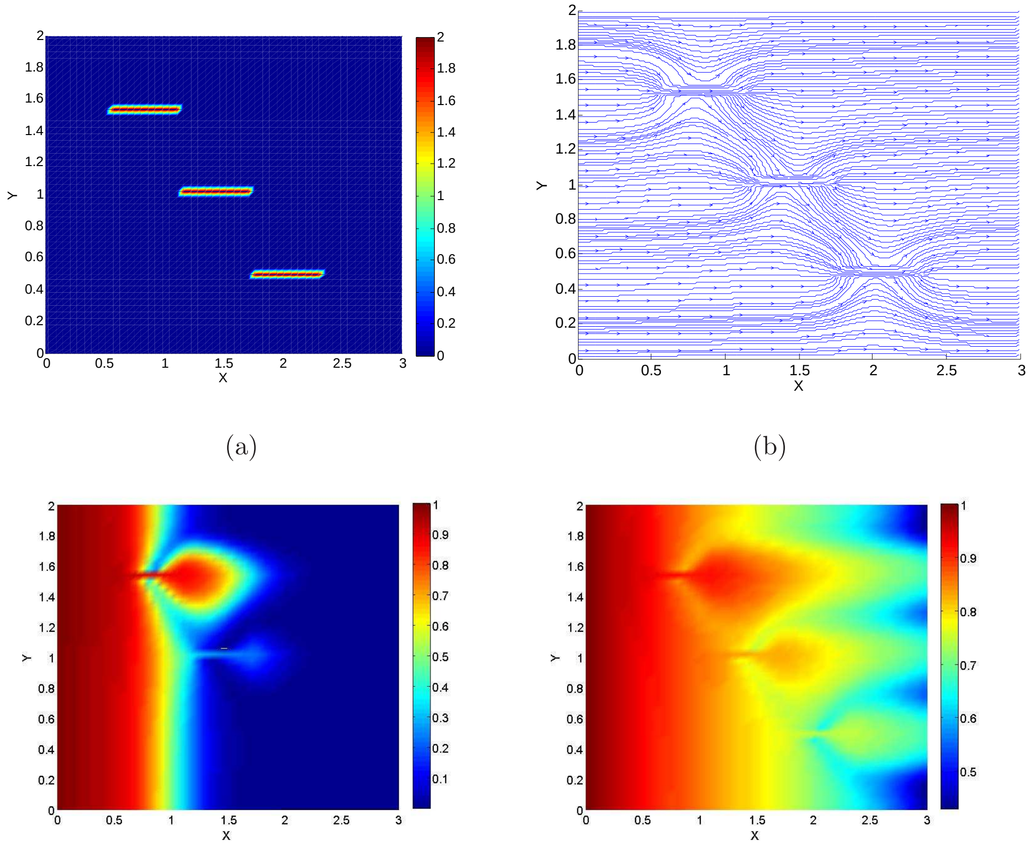
4.3. Deterministic heterogeneous porous media and non-linear reaction

We now test the ETD1 method for a porous media with three parallel high-permeability streaks. This could represent, for example, transport in a highly idealised fracture pattern. The permeability of the three parallel streaks is 100 times greater than the permeability of the surrounding domain (Figure 4(a)). Hence flow is diverted from the lower-permeability rocks into the high-permeability matrix (Figure 4(b)). The advection rates increase towards the high-permeability streaks and are highest in them. This is clearly visible by the closer spacing of the streamlines in the high–permeability streaks (Figure 4(b)).

For the non-linear reaction term we now take the Langmuir sorption isotherm. The domain is given by $\Omega = [0, 2) \times [0, 3)$ and discretised in space with $\Delta x = 3/50$ and $\Delta y = 1/25$. The dispersion tensor is anisotropic with $D_1 = 10^{-3}$, $D_2 = 10^{-4}$. The viscosity is $\mu = 0.1$. The maximum local Péclet number is 2975.4.

Figure 4(c) shows the concentration at $t = 0.3$ and Figure 4(d) the concentration at $T = 1$. Again, the flow-focusing due to the high-permeability streaks is clearly visible.

Figure 5(a) shows the convergence at the final time $T = 1$ in the L^2 norm for varying time-steps Δt. All schemes show convergence rates of $\mathcal{O}(\Delta t)$. There is now a distinct difference between ETD1 method with Krylov or Léja point technique and the implicit and semi-implicit integrators. The EDT1 methods displays a clear improvement in the error constant. Figure 5(b) depicts the L^2 error at $T = 1$ as a function of CPU time. The ETD1 based schemes are significantly more accurate and computationally more efficient than (semi-)implicit schemes. They require between 10 and 100 times less computational effort to achieve the same reduction in numerical error. The Léja point method has also a small computational advantage over the Krylov subspace technique.



(a)

(b)

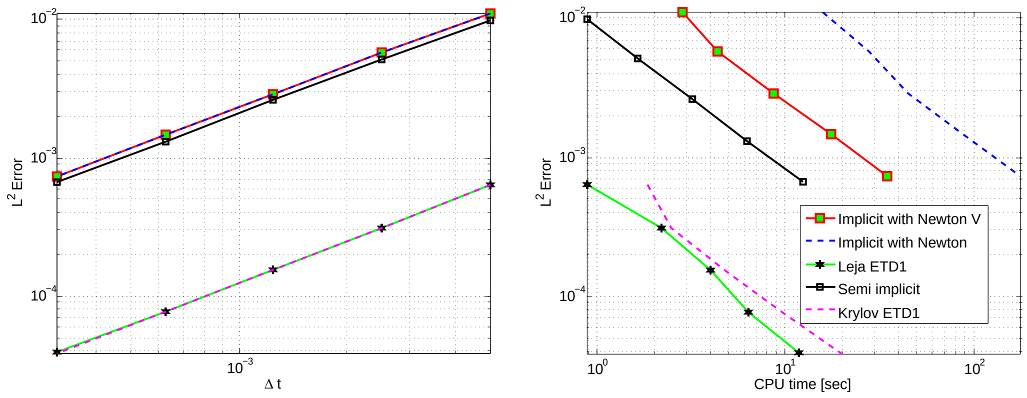
(c)

(d)

Figure 4: Numerical experiments for the non-linear ADR problem in a deterministic heterogeneous porous media (Problem 3). (a) shows the log of permeability field, (b) shows the velocity streamlines, (c) shows the concentration at $t = 0.3$ and (d) shows the concentration field at $T = 1$.

4.4. Stochastic heterogeneous porous media with non-linear reaction

We finally apply the ETD1 method to a stochastically generated permeability field. Stochastic permeability fields are commonly used to represent the unknown heterogeneity in the subsurface. We use the Karhunen-Loeve numerical expansion [39] to generate the random permeability field from



(a)

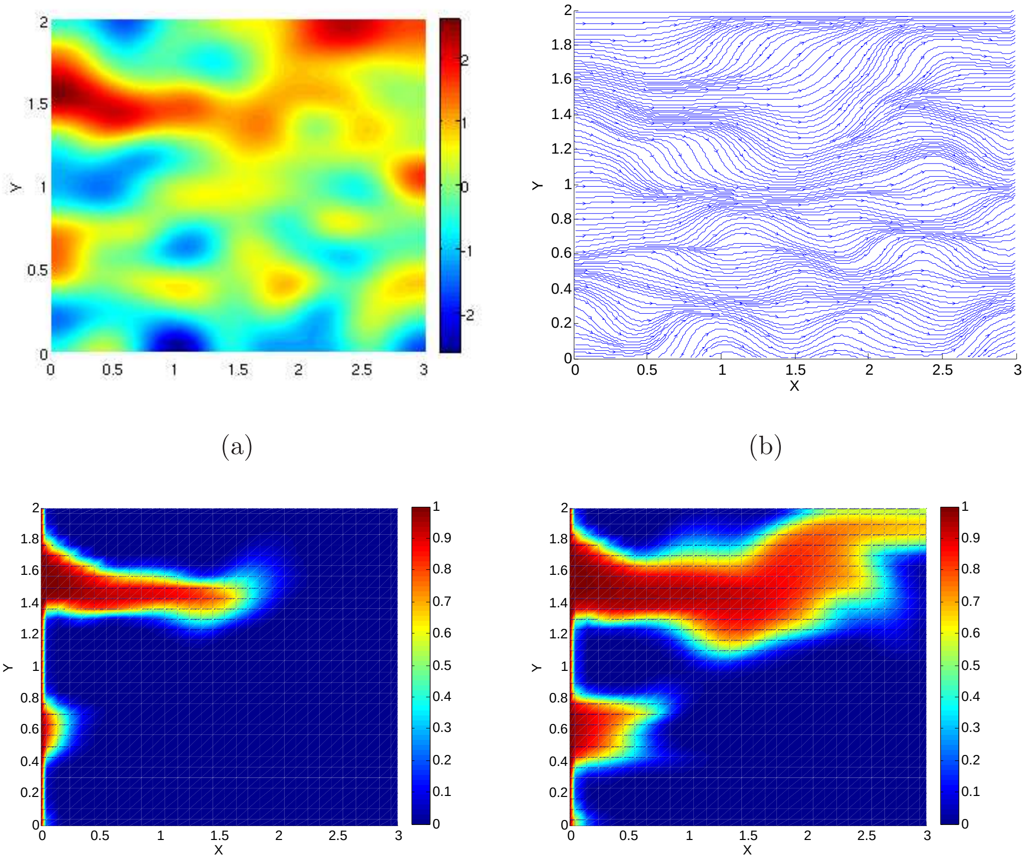
(b)

Figure 5: (a) Convergence of the L^2 norm at $T = 1$ as a function of Δt. (b) The L^2 norm at $T = 1$ as a function of CPU time. Both plots are for the non-linear ADR in a deterministic heterogeneous porous media (Problem 3). Although all time integrators display a convergence rate of $\mathcal{O}(\Delta t)$, there is clear improvement in the error constant. Hence the ETD1 schemes are significantly more efficient than (semi-)implicit methods, with the real Léja point method being most efficient.

a log-normal distribution with an exponentially decaying space correlation. The correlation in the field is given by

$$Q((x_1, y_1); (x_2, y_2)) = \frac{1}{4b_1 b_2} \exp\left(-\frac{\pi}{4} \left[\frac{(x_2 - x_1)^2}{b_1^2} + \frac{(y_2 - y_1)^2}{b_2^2} \right] \right),$$

where b_1 and b_2 are the spatial correlation lengths in x-direction and y-direction, respectively, and given by $b_1 = 0.4$ and $b_2 = 0.2$. We used the first 30 terms in the Kahunen-Loeve numerical expansion. We used the same stochastically generated permeability field to evaluate all time integrators. The domain is given by $\Omega = [0, 3) \times [0, 2)$ with $\Delta x = 1/10$ and $\Delta y = 1/15$. The dispersion tensor has the entries $D_1 = 10^{-3}$, $D_2 = 10^{-4}$ and the viscosity $\mu = 1$. The maximum local Péclet number is $Pe_{loc} = 1649.3$.



(a)

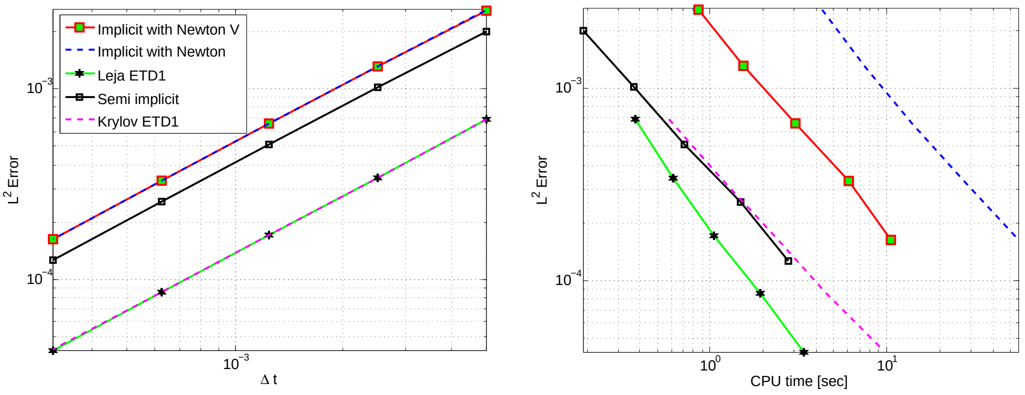
(b)

(c)

(d)

Figure 6: Numerical experiments for non-linear ADR problem in a stochastic heterogeneous porous media (a) shows the log of permeability field, (b) shows the velocity streamlines, (c) shows the concentration at $t = 0.2$ and (d) shows the concentration at $T = 1$.

Figure 6(a) shows the log of the permeability field, which varies over 6 orders of magnitude ranging from 10^{-3} to 10^{3}. Figure 6(b) shows the corresponding streamlines, which show how flow is focused into regions of high permeability. Advection rates are significantly higher in regions of high-permeability, reflected by the close streamline spacing, compared to regions of low permeability. Figure 6(c) shows the concentration at $t = 0.2$ and Figure 6(d) the concentration at $T = 1$. Both show flow flow-focusing into the high-permeability regions.



(a) (b)

Figure 7: (a) Convergence of the L^2 norm at $T = 1$ as a function of Δt. (b) The L^2 norm at $T = 1$ as a function of CPU time. Both plots are for the non-linear ADR in a stochastically generated porous media (Problem 4).

Figure 7(a) shows the convergence of the L^2 norm at $T = 1$ as a function of Δt. As in all our previous applications, all schemes have similar convergence rates of $\mathcal{O}(\Delta t)$, but there is a clear improvement in the error constant for the ETD1 schemes. Figure 7(b) shows the L^2 error as a function of CPU time. The ETD1 methods clearly outperform the implicit time-integrators, with the Léja point based scheme being slightly more efficient than the Krylov subspace based methods. The latter shows a similar performance as the semi-implicit method. Table 1 compares the CPU time necessary to perform 3200 steps of the ETD1 integration using the real Léja point method and the Krylov subspace technique. We analysed how many Léja points are required for the first step for different spatial discretisations ranging from 100×100

grid points to 500 × 2000 grid points. For the largest problem with 10^6 unknowns only 10 Léja points are required. The total CPU time necessary to find the solution at final time $T = 1$ is 5293.3 seconds.

For the Krylov subspace method (with $m = 6$) the total CPU time required to find the solution at the final time $T = 1$ is 14693 seconds. We observe that the real Léja point method seems more efficient than the Krylov implementation, taking approximately half the CPU time. We have tested several values for m and due to the reasons discussed previously, we do not think that the Krylov subspace technique will be more efficient than the real Léja point method for other values of m. Nevertheless, this example demonstrates that ETD1 methods are probably applicable to large-scale 3D reservoir simulations with several million unknowns.

N_x	N_y	$M_{\text{Léja}}$	CPU time [sec] Léja	CPU time [sec] Krylov ($m = 6$)
100	100	6	21	54
200	200	7	123	316
100	1000	7	430	889
200	1000	8	911	2349
100	3000	10	1589	2715
500	2000	10	5293	14693

Table 1: CPU time for the real Léja points and Krylov subspace methods used in Problem 4 as a function of various grid sizes. N_x is the number of subdivisions in the x direction and N_y the number in the y direction. Table shows the number of Léja points used for the first step $M_{\text{Léja}}$ and the CPU time to perform 3200 steps of the ETD1 method using the Léja point method and Krylov subpace technique (with $m = 6$).

5. Concluding remarks

We have developed an exponential time integrator of order one (ETD1) where the matrix exponential is computed with either real fast Léja points techniques or a Krylov subspace technique. We have applied it to a variety of linear and non-linear advection-diffusion-reaction problems in homogeneous as well as highly heterogeneous porous media where the spatial discretisation was achieved by standard upwind-weighted finite volume meshes on non-uniform rectangular grids. The largest problems comprised 10^6 unknowns. We compared the performance of the ETD1 method to standard semi-implicit and implicit time integrators. Transport in our example applications was advection as well as diffusion dominated. All our numerical examples demonstrate that the ETD1 scheme is highly competitive compared to standard time integrators. This competitiveness comprises two components: efficiency and accuracy. Generally, the ETD1 method requires at least 10 times less computational cost compared to implicit time integrators to reduce the numerical error to a certain value. Semi-implicit time integrators perform at best similar to our ETD1 method. The real fast Léja points technique is on average equivalent or more efficient than the Krylov subspace technique. A similar observation was made in Martinez et al. [25] and Bergamaschi et al. [26] for example applications with constant dispersion tensors, uniform velocity fields, and low Péclet number flows, where the spatial discretisation was achieved by finite difference and finite element space discretisation. A single computation of ETD1 with real fast Léja points requires a few seconds on a standard PC, even with our uncompiled Matlab code. Importantly, the CPU time scales linearly with the number of unknowns, which implies that

ETD1 could be readily applied to large-scale 3D reservoir simulations with several million of unknowns. It hence may become a viable alternative to other scalable solvers such as hierarchical algebraic multigrid methods [40] or multi-scale methods (e.g, [41, 42, 43]) which are commonly used for large-scale simulations of flow and transport in heterogeneous porous media.

6. Acknowledgements

We thank three anonymous reviewers for their positive and constructive comments and Prof. E. V. Vorozhtsov for editorial handling. This work was supported, in part, by the initiative "Bridging the Gaps between Mathematics and Engineering" at Heriot-Watt University. S. Geiger thanks the Edinburgh Collaborative of Subsurface Science and Engineering, a joint research institute of the Edinburgh Research Partnership in Engineering and Mathematics, for financial support.

References

[1] M. A. Christie and M. J. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques, *SPE Reserv. Eval. Eng.* 4(4) (2001) 308–317.

[2] S. K. Matthäi and M. Belayneh, Fluid flow partitioning between fractures and a permeable rock matrix, *Geophys. Res. Lett.* 31(7) (2004) L07602 doi:10.1029/2003GL019027.

[3] B. Berkowitz, A. Cortis, M. Dentz, and H. Scher, Modeling non-Fickian transport in geological formations as a continuous time random walk, *Rev. Geophys.* 44(2) (2006) RG2003 doi:10.1029/2005RG000178.

[4] O. Cirpka and P. Kitanidis, Characterization of mixing and dilution in heterogeneous aquifers by means of local temporal moments, *Water Resour. Res.* 36(5) (2000) 1221–1236.

[5] A. M. Tartakovsky, G. Redden, P. L. Lichtner, T. D. Scheibe, and P. Meakin, Mixing-induced precipitation: experimental study and multiscale numerical analysis, *Water Resour. Res.* 44(6) (2008) W06S04 doi:10.1029/2006WR005725.

[6] A. M. Tartakovsky, G. D. Tartakovsky, and T. D. Scheibe. Effects of incomplete mixing on multicomponent reactive transport, *Adv. Water Resour.* 31(11) (2009) 1674–1679 .

[7] M. Christie, V. Demyanov, and D. Ebras, Uncertainty quantification for porous media flows, *J. Comput. Phys.* 217(1) (2006) 143–158.

[8] D. A. Knoll, L. Chacon, L. G. Margolin and V. A. Mousseau, On balanced approximations for time integration of multiple time scale systems, *J. Comput. Phys.* 185(2) (2003) 583–611.

[9] D. L. Ropp, J. N. Shadid, and C. C. Ober, Studies of the accuracy of time integration methods for reaction-diffusion equations, *J. Comput. Phys.* 194(2) (2004) 544–577.

[10] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5(3) (1968) 506–517.

[11] C. I. Steefel and K. T. B. MacQuarrie, Approaches to modeling of reactive transport in porous media, in: P. C. Lichtner, C. I. Steefel, and

E. H. Oelkers (Eds.), Reviews in Mineralogy and Geochemistry Volume 34, Mineralogical Society of America, Chantilly, 1996, pp. 83–129.

[12] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, Implicit explicit methods for time dependent partial differential equations, *SIAM J. Numer. Anal.* 32(3) (1995) 797–823.

[13] M. G. Gerritsen and L. J. Durlofsky, Modeling fluid flow in oil reservoirs, *Annu. Rev. Fluid Mech.* 37 (2005) 211–238.

[14] M. R. Thiele, R. P. Batycky and F. Orr, Simulating flow in heterogeneous systems using streamtubes and streamlines, *SPE Reservoir Eng.* 11(1) (1996) 5–12.

[15] G. DiDonato and M. J. Blunt, Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs, *Water Resour. Res.* 40(4) (2004) W04203 doi:10.1029/2003WR002772.

[16] R. D. Hornung and J. A. Trangenstein, Adaptive mesh refinement and multilevel iteration for flow in porous media, *J. Comput. Phys.* 136(2) (1997) 522–545.

[17] H. Karimabadi, J. Driscoll, Y. A. Omelchenko, and N. Omidi, A new asynchronous methodology for modeling of physical systems: breaking the curse of courant condition, *J. Comput. Phys.* 205(2) (2005) 755–775.

[18] Y. A. Omelchenko and H. Karimabadi, Self-adaptive time integration of flux-conservative equations with sources, *J. Comput. Phys.* 216(1) (2006) 179–194.

[19] B. Minchev and W. Wright, A review of exponential integrators for first order semi-linear problems, *Preprint 2/2005, Norwegian University of Science and Technology, Trondheim Norway 2005*, available at, http://www.math.ntnu.no/preprint/numerics/2005/N2-2005.pdf.

[20] A. K. Kassam and L. N.Trefethen, Fourth-order time stepping for stiff PDES, *SIAM J. Comput.* 26(4) (2005) 1214–1233.

[21] J. Baglama, D. Calvetti, and L. Reichel, Fast Léja points, *Electron. Trans. Num. Anal.* 7 (1998) 124–140.

[22] L. Bergamaschi, M. Caliari, and M. Vianello, The RELPM exponential integrator for FE discretizations of advection-diffusion equations, in: M. Bubak, G. D. Van Albada, P. Sloot (Eds.), Lecture Notes in Computer Sciences Volume 3039, Springer Verlag, Berlin Heidelberg, 2004, pp. 434-442.

[23] M. Hochbruck and C. Lubich, On Krylov subspace approximations to the matrix exponential operator, *SIAM J. Numer. Anal.* 34(5) (1997) 1911–1925.

[24] R. B. Sidje, Expokit: A software package for computing matrix exponentials, *ACM Trans. Math. Software* 24(1) (1998) 130–156.

[25] A. Martinez, L. Bergamaschi, M. Caliari, and M. Vianello, A massively parallel exponential integrator for advection-diffusion models, *J. Comput. Appl. Math.* 231(1) (2009) 82–91.

[26] L. Bergamaschi, M. Caliari, A. Martinez, and M. Vianello, Comparing Léja and Krylov approximations of large scale matrix exponentials, *Comput. Sci. – ICCS* 3994 (2006) 685–692.

[27] M. Caliari, M. Vianello, and L. Bergamaschi, Interpolating discrete advection diffusion propagators at Léja sequences, *J. Comput. Appl. Math.* 172(1) (2004) 79–99.

[28] M. Caliari, M. Vianello, and L. Bergamaschi. The LEM exponential integrator for advection–diffusion–reaction equations. *J. Comput. Appl. Math.* 210(1-2) (2007) 56–63.

[29] M. Caliari and A. Ostermann, Implementation of exponential Rosenbrock-type integrators, *Appl. Numer. Math.* 59(3-4) (2009) 568–581.

[30] R. Eymard, T. Gallouet, and R. Herbin, Finite volume methods, in: P. G. Ciarlet, J. L. Lions (Eds.), Handbook of Numerical Analysis Volume 7, North-Holland, Amsterdam, 2000, pp. 713–1020.

[31] P. Knabner and L. Angermann, Numerical methods for elliptic and parabolic partial differential equations solution, Springer Verlag, Berlin, 2003.

[32] S. M. Cox and P. C. Matthews, Exponential time differencing for stiff systems, *J. Comput. Phys.* 176(2) (2002) 430–455.

[33] H. Berland, B. Skaflestad, and W. Wright, A matlab package for exponential integrators, *ACM Trans. Math. Software* 33(1) (2007) Article No. 4.

[34] J. W. Thomas, Numerical partial differential equations: finite difference methods, Springer Verlag, Berlin Heidelberg New York, 1995.

[35] M. Caliari, Accurate evaluation of divided differences for polynomial interpolation of exponential propagators, *Computing* 80(2) (2007) 189–201.

[36] C. Moler and C. Van Loan, Ninteen Dubious Ways to Compute the Exponential of a Matrix, Twenty–Five Years Later, *SIAM Review* 45(1) (2003) 3–49.

[37] G. H. Golub and C. F. Van Loan, Matrix computations, third ed., Johns Hopkins University Press, Baltimore, 1996.

[38] C. Zoppou and J. H. Knight, Analytical solution of a spatially variable coefficient advection-diffusion equation in up to three dimensions, *Appl. Math. Model.* 23(9) (1999) 667–685.

[39] R. G. Ghanem and P. D. Spanos, Stochastic finite elements, Springer Verlag, New York, 2003.

[40] K. Stüben, A review of algebraic multigrid, *J. Comput. Appl. Math.* 128(1-2) (2001) 281–309.

[41] T. Hou and X. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *J. Comput. Phys.* 134(1) (1997) 169–189.

[42] Y. Efendiev, T. Hou, and X. Wu, Convergence of a nonconformal mul-

tiscale finite element method, *SIAM J. Numer. Anal.* 37(3) (2000) 888–910.

[43] P. Jenny amd H. A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, *J. Comput. Phys.* 187(1) (2003) 47–67.