Computational Geosciences
https://doi.org/10.1007/s10596-024-10293-y

ORIGINAL PAPER

A comparison study of spatial and temporal schemes for flow and transport problems in fractured media with large parameter contrasts on small length scales

Wansheng Gao[1] · Insa Neuweiler[1] · Thomas Wick[2]

Received: 27 June 2023 / Accepted: 25 April 2024

Abstract
In this work, various high-accuracy numerical schemes for transport problems in fractured media are further developed and compared. Specifically, to capture sharp gradients and abrupt changes in time, schemes with low order of accuracy are not always sufficient. To this end, discontinuous Galerkin up to order two, Streamline Upwind Petrov-Galerkin, and finite differences, are formulated. The resulting schemes are solved with sparse direct numerical solvers. Moreover, time discontinuous Galerkin methods of order one and two are solved monolithically and in a decoupled fashion, respectively, employing finite elements in space on locally refined meshes. Our algorithmic developments are substantiated with one regular fracture network and several further configurations in fractured media with large parameter contrasts on small length scales. Therein, the evaluation of the numerical schemes and implementations focuses on three key aspects, namely accuracy, monotonicity, and computational costs.

Keywords Transport problems · Fractured media · Discontinuous Galerkin · Continuous Galerkin · Finite differences · Space-time

1 Introduction

Numerical modeling of flow and scalar transport, such as the transport of dissolved substances or heat, in fractured media, is challenging. One of the major challenges is the contrast of length scales between a large domain and small fractures embedded. While in some cases the fast time scale, determined by the backbone of the fracture network, is the only relevant one, a larger range of time scales matters in many applications, for example, related to contaminant transport [1–5]. This is enforced by the large storage capacity in the matrix compared to the fracture space. In such cases, high-quality solutions are needed.

The great contrast of hydraulic parameters between the highly permeable fractures and the surrounding less permeable matrix also causes challenges [6]. The velocity field in a complex fracture network is highly heterogeneous and reveals preferential flow paths. As a consequence, the corresponding transport process is prone to very sharp gradients at the interface of fractures and the matrix material. Abrupt changes can also occur in time, for example in case of changing inflow over boundaries [5]. Therefore, both the space and time discretization are required to be robust. A particular problem in the case of strong advective transport, which may occur only locally, is spurious oscillations of the numerical solutions, or so-called over- and undershoots. Although these may remain limited, they are nonphysical and can cause large problems, for example, if chemical reactions should be considered or for inverse modeling.

Space discretization methods for numerical models for transport in fractured media can be broadly categorized into

Insa Neuweiler and Thomas Wick are both contributed equally to this work.

✉ Wansheng Gao
gao@hydromech.uni-hannover.de

Insa Neuweiler
neuweiler@hydromech.uni-hannover.de

Thomas Wick
thomas.wick@ifam.uni-hannover.de

[1] Institute of Fluid Mechanics and Environmental Physics in Civil Engineering, University of Hannover, Appelstr. 9a, Hannover 30167, Germany

[2] Institute of Applied Mathematics, University of Hannover, Welfengarten 1, Hannover 30167, Germany

Published online: 13 May 2024

two groups: hybrid-dimensional models [7] and continuum models [8]. In the former approach, fractures are explicitly represented in one dimension lower than the surrounding matrix. Continuum models present both fractures and the surrounding matrix in the same dimension. Especially when the separation of fractures and the surrounding matrix is not so clear and separate treatment of interfacial fluxes should be avoided, also a full dimensional representation of the fracture domain can be beneficial (for example [9]). However, if meshes are too coarse, detailed information on a fracture network, such as connections, can be lost. How to properly refine the mesh and meanwhile maintain the major topology of the underlying fracture network is a non-trivial task for continuum models [10].

The finite volume method (FVM) has gained considerable popularity for space discretization of transport problems because of its local conservation property and ease of implementing numerical fluxes as well as limiters that avoid non-monotonic behavior of the solutions. However, it is not straightforward to construct high-order FV schemes [11]. The finite element method (FEM) allows for accurate solution approximation either by increasing the polynomial degree or refining the mesh, namely the hp refinement [12]. It is well-known that the standard version of the FEM called the continuous Galerkin (CG) method can easily produce instabilities and spurious oscillations for advection-dominated problems [13].

To deal with instabilities and artificial oscillations, facilitating upwinding or stabilized FEMs in continuous Galerkin schemes can be used, such as the Streamline Upwind Petrov-Galerkin (SUPG) method [13], Galerkin least-squares (GLS) [14], finite incremental calculus (FIC) [15] and flux-corrected transport methods (FCT) [16]. The main idea behind SUPG is adding artificial diffusion in the streamline direction. A rigorous analysis of SUPG is presented in [17, 18]. A great amount of SUPG applications can be found in advection-diffusion problems [19, 20] and Navier-Stokes problems [21, 22]. The main issue of SUPG is choosing the stabilization parameter. It should be large enough to ensure the monotonic solution and meanwhile small enough to not introduce excessive artificial diffusion. An optimal choice is usually unknown in practice. SUPG has been used for modeling transport in fractured media. For example, in the study of [23], spurious oscillations were found when modeling transport in fractured media with standard FEMs. The solutions showed monotonicity conservation when the SUPG method was used.

Another alternative is the Discontinuous Galerkin (DG) method [24, 25], which can be considered as the combination of FVM and FEM. It maintains the element-wise conservative property and offers great flexibility in defining interelementary numerical fluxes. Furthermore, inherited from the hp-FEM [12, 26], the DG method can also permit very accurate approximating solutions. Due to the flexibility of handling interelementary fluxes and upwinding schemes, which are often beneficial for stability, schemes can be set up in a straightforward manner. To date, a number of DG methods have been developed. The primal approaches, e.g. the upwind DG method and the interior penalty (IP) methods, explicitly define internal numerical fluxes by the unknown quantities. One does not need to introduce new variables, thus the approach is easy to implement. However, the resulting system matrix is relatively dense because the DoFs of neighbor elements also contribute to the system matrix, causing challenges for the linear system solver. In more recent variants of DG methods, e.g. the local DG method [27] and the hybridizable DG method [28], auxiliary variables indicating numerical fluxes are introduced in the variational formulation. These approaches can alleviate the dense system matrix, but additional DoFs are introduced. Altogether, compared to the conventional Galerkin approach, the family of DG methods is computationally more costly because of the large number of DoFs and a denser system matrix. DG methods for space discretization have been studied for modeling transport in highly heterogeneous media or fracture media with full dimensional resolution of fractures, for example in [9] for purely advective transport.

In the work of [29], DG has also been used for space discretization of fractures in a fractured medium, where the lumped mixed finite element method with an upwind FVM was used for discretizing the matrix domain. High-order adaptive time integration techniques using the method of lines were applied. In their validation simulations, spurious oscillations were mostly generated when using a standard FE method for the discretization in the matrix domain, but were not generated in the fractures. Improved solutions in terms of numerical dissipation could be demonstrated when comparing DG to standard finite volume schemes for the transport in the fractures. In the study of [30], slope limiters were implemented to diminish spurious oscillations when solving transport problems in fractured media using the enriched Galerkin method and the interior penalty method, which was developed in the paper, with the third-order strong stability Runge-Kutta method for time discretization.

Concerning the time discretization approach, the low-order implicit Euler method produces monotonic solutions, but is prone to significant numerical dissipation [31]. The Crank-Nicolson (CN) method, recognized for its second-order accuracy and unconditional stability, is widely used in diffusion problems. However, when applied to advection-diffusion problems, it may exhibit non-monotonic behavior unless the time step is appropriately refined. This implies a constraint on the time step size, dictated by both the element size and the velocity field. This constraint can pose challenges in scenarios involving focused channels with high flow velocities or when an extremely fine mesh is necessary to resolve the fracture zone. Explicit schemes, such as explicit

Runge-Kutta methods, can yield highly accurate solutions, but are subject to the Courant-Friedrich-Levy (CFL) condition. Consequently, for transport problems in fractured media, an accurate monotonic-preserving time discretization method is often preferable. The time Discontinuous Galerkin (TDG) method offers such an alternative [32–35]. Similar to the DG method in space, the TDG method allows for easy adjustment of the polynomial degree while incorporating numerical fluxes into two adjacent temporal subintervals, enhancing the stability of the temporal scheme.

Most space-time approaches to date are based on the Cartesian product of the spatial and the temporal subdomains, e.g. see [36, 37]. Thus this method is also known as the tensor-product TDG method [38, 39]. Several applications can be found in [40–45]. The TDG method is also computationally demanding compared to finite difference methods (FDM). The extra costs of TDG can partly be compensated by using coarser discretization or a suitable solver strategy, e.g. a fixed-point iteration method used in this work (see Appendix B).

There are also a few works jointly using the DG method both in space and time. In the work of [46], detailed proofs of stability and error estimates, for solving advection-diffusion equations, are given. In the work of [47], the etching process of a slit is simulated by using this space-time scheme. Space-time DG method for flow and transport simulations in porous media was tested by [48]. They also considered heterogeneous permeability distributions, however, not fractured media. Therefore, the size of the grid cells did not vary too much. Different interior penalty methods and mesh densities are compared qualitatively. All IP schemes produce similar solutions, with local over- and undershoots near the steep concentration fronts detected. However, the overall solution is stable.

Overall, numerical solutions of advection-dominated transport in fractures often suffer from stability issues, prompting the focus on stability-preserving numerical schemes both in space and time. For space discretization, DG has in many studies been found to be of advantage for capturing moving sharp fronts. However, the method is computationally costly. Stabilized CG methods such as SUPG, which are more common and already implemented in many codes are less costly. Both methods allow for upwinding, but how much this improves spurious oscillations is unclear. For time discretization, also the TDG method is advantageous for stability, but they are computationally more costly than more common finite difference time integration schemes. They should be beneficial for spurious oscillations compared to higher-order explicit schemes. So far no systematical comparison of these methods has been done for transport problems in complex fracture networks.

In the current work, we apply the full space-time FE formulation for solving transport problems in fractured media. The goal is to compare DG methods for space and/or time discretization to advanced CG methods that are more often implemented in existing software.

With this comparison, we want to answer the following questions:

- For which conditions are DG methods for space discretization beneficial compared to more common stabilized CG discretization schemes to model transport in fractured media?
- For which conditions are TDG methods for time integration beneficial to model transport in fractured media when compared to standard time integration schemes?
- What are the computational costs for schemes that produce more accurate solutions?

The criteria for comparison are accuracy, monotonicity (spurious oscillations), and computational costs.

For space discretization, we consider the symmetric interior penalty Galerkin (SIPG) version of DG methods, where the upwind advective flux and the average diffusive flux are imposed on interelementary interfaces. The TDG method is used for the time discretization. By properly choosing jump terms at temporal interelementary interfaces, the transient problem is solved in a sequential manner, as usual time-stepping methods do (e.g. the Backward Euler (BE) method and the Crank-Nicolson method). For efficiency, a fixed-point iteration method is proposed in this work, which can greatly reduce the computation times for TDG schemes with the polynomial degree ≥ 1. It should be noted that we compare these schemes, but do not consider different DG schemes.

The paper is organized as follows. Section 2 briefly introduces the scalar transport process which is coupled with the flow process via flow velocities and a mass balance for the fluid. In Section 3 the variational formulations of DG, TDG, and SUPG are performed. Section 4 demonstrates the workflow of numerical algorithms implemented in this work. Section 5 presents the convergence study for the validation purpose. Three numerical examples in fractured media are shown in Section 6. The last two sections provide discussion and conclusions of the results.

2 Problem statement

The steady-state groundwater flow equation for confined aquifers in the mixed formulation [49–51] reads

$$\begin{aligned} \boldsymbol{q} + \boldsymbol{K}\nabla h &= 0, \\ -\nabla \cdot \boldsymbol{q} &= Q, \end{aligned} \tag{1}$$

where $h \in \mathbb{R}$ is the hydraulic pressure head. $\boldsymbol{K} \in \mathbb{R}^{dim \times dim}$

is the hydraulic conductivity coefficient, which is a second-order tensor with dim components, where dim is the spatial dimension. $\boldsymbol{K}$ is often highly heterogeneous and sometimes anisotropic in practice [52, 53]. In fractured media, the value of $\boldsymbol{K}$ in fractures is often orders of magnitude greater than that in the surrounding matrix. $\boldsymbol{q}$ is the Darcy flux and Q is the source or sink term such as pumping and injecting wells. The flow velocity $\boldsymbol{a} \in \mathbb{R}^{dim}$ then equals the Darcy flux divided by the porosity ϕ:

$$\boldsymbol{a} = \frac{\boldsymbol{q}}{\phi}. \tag{2}$$

A general transient scalar transport equation takes the form

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{a}u) - \nabla \cdot (\boldsymbol{D}\nabla u) = f \quad \forall\{x, t\} \in \Omega \times I, \tag{3}$$

where $I := [t_0, t_0 + T]$ is the time interval with the end time $T > 0$. Equation 3 is subject to the boundary and initial conditions

$$u = g_D, \quad x \in \Gamma_D, \tag{3a}$$

$$\boldsymbol{n} \cdot \boldsymbol{D}\nabla u = g_N, \quad x \in \Gamma_N, \tag{3b}$$

$$u = u_0, \quad t = t_0. \tag{3c}$$

Here, $u\,(x, t) \in \mathbb{R}$ is the scalar unknown variable, which can be for example concentration or temperature. $\boldsymbol{a} \in \mathbb{R}^{dim}$ is the advection coefficient namely the flow velocity defined in Eq. 2. $\boldsymbol{D} \in \mathbb{R}^{dim\times dim}$ refers to the diffusion (or heat conductivity) coefficient. $f \in \mathbb{R}$ denotes the source or sink term. Γ_D and Γ_N represent the Dirichlet boundary and the Neumann boundary, where $\boldsymbol{n}$ is the unit normal vector exterior to the boundary, g_D and g_N are the boundary values respectively defined on Γ_D and Γ_N. u_0 denotes the initial condition in the initial time step t_0. Moreover, we assume that the domain boundary $\partial\Omega$ is subdivided into two disjoint sets Γ_D and Γ_N: $\partial\Omega = \Gamma_D \cup \Gamma_N, \Gamma_D \cap \Gamma_N = \varnothing$.

3 Numerical methods

In this section, we introduce both DG and SUPG discretization methods in space and the TDG method. The methods are taken from the literatures as described in Section 1 and are here outlined for completeness.

3.1 Semi-discretization in space with DG

3.1.1 Partitions and finite element space

Let $\mathcal{T}_h = \{K\}$ be the mesh of the spatial domain Ω, where $K \in \mathcal{T}_h$ is a single element. We define the boundaries of the single element K by ∂K and there are three possible locations of ∂K:

1. on an outer boundary where the Dirichlet boundary is defined
2. on an outer boundary where the Neumann boundary is defined
3. on an interelementary interface shared with another element.

We indicate the three cases above by ∂K_D, ∂K_N and ∂K_{int} respectively. Additionally, we introduce the superscripts + and − that denote two adjacent elements K^+ and K^- that share an interelementary interface. The set consisting of all interelementary interfaces is then defined as:

$$\Gamma_{int} = \left\{\partial K^+_{int} \cap \partial K^-_{int} : \forall K^+, K^- \in \mathcal{T}_h\right\}. \tag{4}$$

We define two operators that act on an interelementary interface F:

1. Jump operator: $[\![u]\!] = u^+ - u^-$
2. Mean operator: $\{\!\{u\}\!\} = \frac{u^+ + u^-}{2}$,

in which u^+ and u^- are two functions respectively defined on the two sides of F, namely F^+ and F^- shown in Fig. 1.

Let $Q_{p_s}(K)$ be the space of tensor-product polynomials of degree $p_s \in \mathbb{N}_0$ defined on a finite element K, the finite element space of the spatial part is defined as:

$$\begin{aligned}\mathcal{D}_{p_s}(\mathcal{T}_h) : &= \{v \in L^2(\Omega) \mid \\ &\qquad v\,|_K \in Q_{p_s}(K) \text{ for } K \in \mathcal{T}_h\} \\ &= \{\varphi_1, \cdots, \varphi_{n_s}\}, \dim(\mathcal{D}_{p_s}(\mathcal{T}_h)) = n_s,\end{aligned} \tag{5}$$

where n_s is the degree of freedom (DoF) in space. The corresponding shape functions $\{\varphi_1, \cdots, \varphi_{n_s}\}$ are constructed from $Q_{p_s}(K)$, which are obtained from $\hat{Q}_{p_s}(\hat{K})^{dim}$ transformations of polynomials in $\hat{Q}_{p_s}(\hat{K})$, so-called isoparametric finite elements, and where

$$\hat{Q}_{p_s}(\hat{K}) = \operatorname{span}\Big\{\prod_{j=1}^{dim} x_j^{\alpha_j} \mid \alpha_j \in \mathbb{N}_0, \alpha_j \leq p_s\Big\}. \tag{6}$$

We also refer to [37] for more details.

For the DG discretization, there are neither continuity constraints between two adjacent elements nor strong constraints on the Dirichlet boundary. We denote the discrete space for the DG discretization by $\mathcal{D}^{DG}_{p_s}$. Obviously, the space $\mathcal{D}^{DG}_{p_s}$ is piecewise discontinuous. Throughout this work, we use a uniform degree of polynomials for all elements and introduce the notation DG(p_s) which indicates the DG method with polynomials of degree p_s in space.

We make the solution ansatz in space

$$u_h(x,t) = \sum_{j=1}^{n_s} \varphi_j(x)\, c_j(t)\,, \tag{7}$$

where n_s is the DoF in space and $c_j(t)$ is the time-dependent unknown coefficient at the spatial node j. The shape function $\varphi_j(x)$ is only defined on an element K, to which the nodal index j belongs. A nodal index j does only belong to one element. An example of index numbering of two adjacent elements is shown in Fig. 1. For every fixed time t, the solution approximation $u_h(x,t)$ is a piecewise discontinuous function defined on $\mathcal{D}_{p_s}^{DG}(\mathcal{T}_h)$.

We then apply the standard Galerkin approach for the spatial test function $v_h(x)$, which is also discontinuous and defined as a set of shape functions: $v_h = \{\varphi_i\}$ for $i = 1, 2, \ldots, n_s$.

3.1.2 Variational formulation

Incorporating the solution approximation u_h, multiplying the governing (3) by the spatial test function $v_h(x)$ and integrating over the domain and integrating by parts yields:

$$\begin{aligned}&\left(v_h, \frac{\partial u_h}{\partial t}\right)_{\mathcal{T}_h} - (\nabla v_h, \boldsymbol{a} u_h)_{\mathcal{T}_h} + \\ &(\nabla v_h, \boldsymbol{D}\nabla u_h)_{\mathcal{T}_h} + \\ &\sum_{K\in\mathcal{T}_h}\sum_{F\in\partial K}\Big\{\langle v_h, \boldsymbol{n}\cdot\boldsymbol{a}u_h\rangle_F - \\ &\langle v_h, \boldsymbol{n}\cdot\boldsymbol{D}\nabla u_h\rangle_F\Big\} \\ &= (v_h, f)_{\mathcal{T}_h}\,,\end{aligned} \tag{8}$$

where $(\cdot,\cdot)_{\mathcal{T}_h} = \int_{\mathcal{T}_h}\cdot\cdot\, dx$ and $\langle\cdot,\cdot\rangle_F = \int_F \cdot\cdot\, dx$ denotes face integral on element boundaries.

Assembling the volume terms $(\cdot,\cdot)_{\mathcal{T}_h}$ is done in a standard manner and will here not be outlined further. It is worth stressing that by assembling volume terms $(\cdot,\cdot)_{\mathcal{T}_h}$ yields a block diagonal matrix. The most essential part of the DG method is how to treat the face terms $\langle\cdot,\cdot\rangle_F$. As the next steps, the use of an upwinding method for the advection term and the interior penalty Galerkin (IPG) methods for the diffusion term will be outlined separately.

3.1.3 Advection term

By using the classical upwind method to obtain stable solutions, the consistent formulation of the face integral of the advection part reads:

$$\begin{aligned}&\sum_{K\in\mathcal{T}_h}\sum_{F\in\partial K}\langle v_h, \boldsymbol{n}\cdot\boldsymbol{a}u_h\rangle_F \\ &= \sum_{K\in\mathcal{T}_h}\Bigg\{\sum_{F\in\partial K_N}\langle v_h, \boldsymbol{n}\cdot\boldsymbol{a}u_h\rangle_F + \\ &\sum_{F\in\partial K_{D_{out}}}\langle v_h, \boldsymbol{n}\cdot\boldsymbol{a}u_h\rangle_F + \\ &\sum_{F\in\partial K_{D_{in}}}\langle v_h, \boldsymbol{n}\cdot\boldsymbol{a}g_D\rangle_F\Bigg\} + \\ &\sum_{F\in\Gamma_{int}}\left\langle [\![v_h]\!], \boldsymbol{n}^{\mathrm{up}}\cdot\boldsymbol{a}u_h^{\mathrm{up}}\right\rangle_F\,,\end{aligned} \tag{9}$$

where for the element boundary ∂K_D it is distinguished between the inflow part and the outflow part:

$$\partial K_D = \begin{cases}\partial K_{D_{out}}, & \text{if } \boldsymbol{n}\cdot\boldsymbol{a} > 0 \\ \partial K_{D_{in}}, & \text{if } \boldsymbol{n}\cdot\boldsymbol{a} < 0\end{cases}. \tag{10}$$

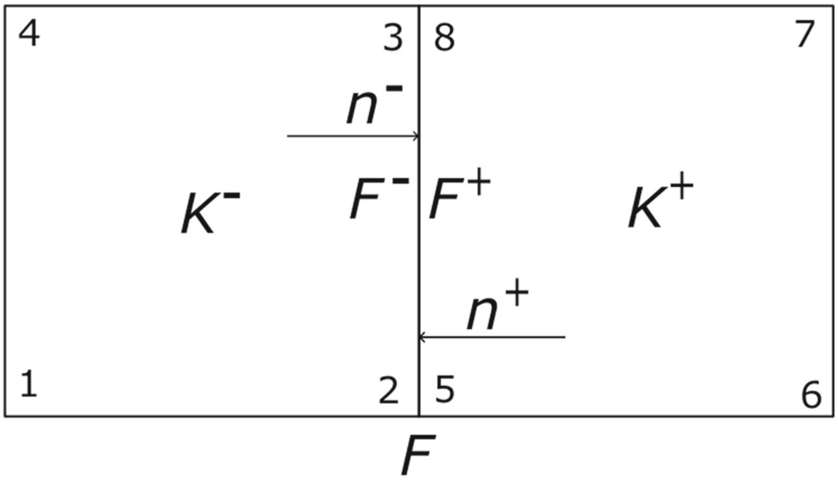


Fig. 1 Two adjacent DG elements sharing the interior interface F. Element K^- is numbered by node indices 1,2,3,4 and element K^+ is numbered by node indices 5,6,7,8. Face F^- is numbered by node indices 2,3 and face F^+ is numbered by node indices 5,8

And the upwind value on an interelementary interface is taken to be:

$$\boldsymbol{n}^{\text{up}} \cdot \boldsymbol{a} u_h^{\text{up}} = \begin{cases} \boldsymbol{n}^+ \cdot \boldsymbol{a} u_h^+, & \text{if } \boldsymbol{n}^+ \cdot \boldsymbol{a} > 0 \\ \boldsymbol{n}^- \cdot \boldsymbol{a} u_h^-, & \text{if } \boldsymbol{n}^- \cdot \boldsymbol{a} > 0 \end{cases}. \tag{11}$$

The definition of the unit normal vectors $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ is illustrated in Fig. 1. The main idea behind the upwind method is that the cross-over advective flux is approximated by the flux from the upstream.

3.1.4 Diffusion term

The diffusion terms are treated with the IPG method. This leads to the formulation:

$$\begin{aligned}
&\sum_{K \in \mathcal{T}_h} \sum_{F \in \partial K} \langle v_h, \boldsymbol{n} \cdot \boldsymbol{D} \nabla u_h \rangle_F \\
&= \sum_{K \in \mathcal{T}_h} \Bigg\{ \sum_{F \in \partial K_N} \langle v_h, g_N \rangle_F + \\
&\sum_{F \in \partial K_D} \langle v_h, \boldsymbol{n} \cdot \boldsymbol{D} \nabla u_h \rangle_F - \\
&\sum_{F \in \partial K_D} \langle \boldsymbol{n} \cdot \boldsymbol{D} \nabla v_h, \epsilon \left(u_h - g_D\right) \rangle_F - \\
&\sum_{F \in \partial K_D} \langle v_h, \sigma \mid \boldsymbol{D} \mid (u_h - g_D) \rangle_F \Bigg\} + \\
&\sum_{F \in \Gamma_{int}} \langle [\![v_h]\!], \{\!\{ \boldsymbol{n} \cdot \boldsymbol{D} \nabla u_h \}\!\} \rangle_F - \\
&\sum_{F \in \Gamma_{int}} \langle \{\!\{ \boldsymbol{n} \cdot \boldsymbol{D} \nabla v_h \}\!\}, \epsilon [\![u_h]\!] \rangle_F - \\
&\sum_{F \in \Gamma_{int}} \langle [\![v_h]\!], \sigma \mid \boldsymbol{D} \mid [\![u_h]\!] \rangle_F .
\end{aligned} \tag{12}$$

The terms containing the part $(u_h - g_D)$ indicate the penalizing treatment of the Dirichlet boundary. Similarly, the jump term $[\![u_h]\!]$ means that the continuity condition of interelementary interfaces is weakly imposed by penalizing. The part $\{\!\{ \boldsymbol{n} \cdot \boldsymbol{D} \nabla u_h \}\!\}$ means that the diffusive cross-over flux is approximated by the average flux of two adjacent elements and ensures the formulation is consistent. The terms multiplied by the parameters ϵ and σ are the (non-)symmetrizing terms and penalty terms respectively. Since we only consider isotropic diffusive cases in this work, $\mid \boldsymbol{D} \mid$ refers to the L_1 norm of scalar diffusion coefficients. Choosing different values of ϵ and σ we get the four variants of the IPG methods:

1. Symmetric IPG (SIPG): $\epsilon = -1$ and $\sigma > 0$
2. Non-symmetric IPG (NIPG): $\epsilon = 1$ and $\sigma > 0$
3. Incomplete IPG (IIPG): $\epsilon = 0$ and $\sigma > 0$
4. Oden, Babuška and Baumann scheme (OBB): $\epsilon = 1$ and $\sigma = 0$.

The penalty parameter σ has a dependency on the local element size h_K and can be defined as $\sigma = \alpha / h_K$. The constant α is a problem-specific parameter [24, 25, 54]. Throughout this work, we use the SIPG method, and the constant α is set to be $max\{1, p_s(p_s + 1)\}$.

3.2 Semi-discretization in space with SUPG

3.2.1 Finite element space

We first outline the FE subspace (5) for the conventional CG method, denoted by $\mathcal{D}_{p_s}^{CG}$. The space $\mathcal{D}_{p_s}^{CG}$ is piecewise continuous because the continuity condition is enforced on interelementary interfaces. The solution ansatz retains the same form as Eq. 7. However, the shape functions $\varphi_j(x)$ are piecewise continuous and defined in the whole domain.

In the case of the SUPG discretization [13, 18], the modified test function reads

$$v_h^{SUPG} = v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \tag{13}$$

where the primal test function v_h is defined in $\mathcal{D}_{p_s}^{CG}$ above. The stabilization parameter τ proposed in [13] is locally defined on the element K as

$$\tau = \frac{h_K}{2 \mid \boldsymbol{a} \mid} \zeta \left(Pe_K\right), \tag{14}$$

where h_K is the element size and $\zeta\left(Pe_K\right)$ is a function of the grid Peclet number:

$$\zeta\left(Pe_K\right) = coth\left(Pe_K\right) - \frac{1}{Pe_K}, \tag{15}$$

$$Pe_K = \frac{\mid \boldsymbol{a} \mid h_K}{2 \mid \boldsymbol{D} \mid}. \tag{16}$$

3.2.2 Variational formulation

Multiplying the governing (3) by the SUPG test function (13) and integrating over the domain results in the variational formulation:

$$\begin{aligned}
&\left(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \frac{\partial u_h}{\partial t}\right)_{\mathcal{T}_h} + \\
&\left(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \nabla \cdot (\boldsymbol{a} u_h)\right)_{\mathcal{T}_h} - \\
&\left(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \nabla \cdot (\boldsymbol{D} \nabla u_h)\right)_{\mathcal{T}_h} \\
&= \left(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, f\right)_{\mathcal{T}_h} .
\end{aligned} \tag{17}$$

By integrating by parts we obtain the final variational formulation:

$$\begin{aligned}
&\left(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \frac{\partial u_h}{\partial t}\right)_{\mathcal{T}_h} + \\
&(v_h + \tau \boldsymbol{a} \cdot \nabla v_h, \nabla \cdot \boldsymbol{a} u_h)_{\mathcal{T}_h} + \\
&(\nabla v_h, \boldsymbol{D} \nabla u_h)_{\mathcal{T}_h} - (\tau \boldsymbol{a} \cdot \nabla v_h, \nabla \cdot \boldsymbol{D} \nabla u_h)_{\mathcal{T}_h} \\
&= (v_h + \tau \boldsymbol{a} \cdot \nabla v_h, f)_{\mathcal{T}_h} + \sum_{K \in \mathcal{T}_h} \sum_{F \in \partial K_N} \langle v_h, g_N \rangle_F . \qquad (18)
\end{aligned}$$

The Neumann boundary condition is implemented as a right-hand-side term in Eq. 18. The Dirichlet boundary is treated as an essential boundary condition. In the following, we use SUPG(p_s) to denote the SUPG scheme with polynomials of degree p_s.

3.3 Time discretization

3.3.1 ODE system with time derivative

By implementing the solution ansatz and the test function either with DG (Section 3.1.1) or SUPG (Section 3.2.1) into the corresponding variational formulation, the original governing equation is transformed into n_s^{th} order system of ODEs. Each row has the following statement:

$$\boldsymbol{M}_{ij} \sum_{j=1}^{n_s} \dot{c_j}\,(t) + \boldsymbol{A}_{ij} \sum_{j=1}^{n_s} c_j\,(t) = \boldsymbol{B}_i, \qquad (19)$$

where $i = 1, 2, \cdots, n_s$. The matrix $\boldsymbol{M}$ denotes the mass matrix, $\boldsymbol{A}$ is the system matrix which results from the space discretization of advection and diffusion terms, and $\boldsymbol{B}$ is the right-hand-side term.

The tensor-product TDG method for treating the time derivative in Eq. 19 will be presented as follows.

3.3.2 Partitions and finite element space

Let $\mathcal{I}_h = \{I_m\}$ be the decomposition of the time interval I, where $I_m \in \mathcal{I}_h$ is a single 1D temporal element and spans the subinterval $\left[t_{m-1}, t_m\right]$. Let $P_{p_t}\,(I_m) \in \mathbb{R}$ be the space of polynomials of degree $p_t \in \mathbb{N}_0$ defined on a temporal element I_m. The finite element space of the temporal part is then defined as:

$$\begin{aligned}
\mathcal{V}_{p_t}\,(\mathcal{I}_h) : = &\Big\{ v \in L^2(I) \mid \\
&\quad v\,|_{I_m} \in P_{p_t}\,(I_m) \text{ for } I_m \in \mathcal{I}_h \Big\} \\
&= \{\psi_1, \cdots, \psi_{n_t}\}, \dim(\mathcal{V}_{p_t}\,(\mathcal{I}_h)) = n_t, \qquad (20)
\end{aligned}$$

where n_t is the DoF in time. The corresponding temporal shape functions are denoted by $\psi_1, \cdots, \psi_{n_t}$, which are obtained from

$$\hat{P}_{p_t}(\hat{I_m}) = \operatorname{span}\big\{ t^{\alpha_j} \mid \alpha_j \in \mathbb{N}_0, \alpha_j \leq p_t \big\}. \qquad (21)$$

Similarly, there is no continuity constraint between two adjacent elements in the TDG method. We let TDG(p_t) denote the TDG method with polynomials of degree p_t in time.

The solution ansatz (7) can then be extended to the space-time version:

$$\begin{aligned}
u_h\,(x, t) &= \sum_{j=1}^{n_s} \varphi_j\,(x)\, c_j\,(t) \\
&= \sum_{j=1}^{n_s} \varphi_j\,(x) \left(\sum_{l=1}^{n_t} \psi_l\,(t)\, \xi_{jl} \right), \qquad (22)
\end{aligned}$$

with

$$c_j\,(t) = \sum_{l=1}^{n_t} \psi_l\,(t)\, \xi_{jl}. \qquad (23)$$

Here, $\psi_l\,(t)$ is the temporal shape function at the temporal node l and ξ_{jl} is the scalar coefficient at the spatial node j and the temporal node l.

3.3.3 Variational formulation with jump terms

Multiplying (19) by the temporal test function $\psi_k\,(t)$, incorporating the solution ansatz Eq. 23 and integrating over the temporal domain yields the variational formulation in space-time:

$$\begin{aligned}
&\int_{\mathcal{I}_h} \psi_k \left(\boldsymbol{M}_{ij} \dot{\psi}_l\,(t) + \boldsymbol{A}_{ij} \psi_l\,(t) \right) \sum_{j=1}^{n_s} \sum_{l=1}^{n_t} \xi_{jl} dt \\
&= \int_{\mathcal{I}_h} \psi_k \boldsymbol{B}_i dt, \qquad (24)
\end{aligned}$$

where $k = 1, 2, \cdots, n_t$, $i = 1, 2, \cdots, n_s$. By sorting all the space-time nodes, the scalar coefficients can be written in vector form:

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}^1 \\ \cdots \\ \boldsymbol{\xi}^l \\ \cdots \\ \boldsymbol{\xi}^{n_t} \end{bmatrix}, \qquad (25)$$

where

$$\boldsymbol{\xi}^l = \begin{bmatrix} \xi_{1l} \\ \cdots \\ \xi_{jl} \\ \cdots \\ \xi_{n_s l} \end{bmatrix} \qquad (26)$$

denotes the coefficient vector at the temporal node l (say $\boldsymbol{\xi}^0$ being the given initial condition). Correspondingly, Eq. 24 is taken to be a block linear system:

$$\boldsymbol{L}\boldsymbol{\xi} = \boldsymbol{b}, \qquad (27)$$

where $\boldsymbol{L}$ is a block matrix having $n_t \times n_t$ blocks. Each block reads:

$$\boldsymbol{L}_{kl} = \begin{bmatrix} \int_{\mathcal{I}_h} \psi_k \boldsymbol{M}_{11} \dot{\psi}_l dt & & \\ & \ddots & \\ & & \int_{\mathcal{I}_h} \psi_k \boldsymbol{M}_{n_s n_s} \dot{\psi}_l dt \end{bmatrix} + \begin{bmatrix} \int_{\mathcal{I}_h} \psi_k \boldsymbol{A}_{11} \psi_l dt & & \\ & \ddots & \\ & & \int_{\mathcal{I}_h} \psi_k \boldsymbol{A}_{n_s n_s} \psi_l dt \end{bmatrix} \in \mathbb{R}^{n_s \times n_s}, \qquad (28)$$

and $\boldsymbol{b}$ is a block vector having n_t blocks. Each block reads:

$$\boldsymbol{b}_k = \begin{bmatrix} \int_{\mathcal{I}_h} \psi_k \boldsymbol{B}_1 dt \\ \vdots \\ \int_{\mathcal{I}_h} \psi_k \boldsymbol{B}_{n_s} dt \end{bmatrix} \in \mathbb{R}^{n_s}. \qquad (29)$$

Obviously, system (27) is block-wise diagonal due to the discontinuous function space in time. Each block corresponds to a subinterval in time. Hence it is possible to decouple (27) and solve it block by block.

As stated in [55], for the TDG method a jump term weakly imposing the continuity condition is crucial at every interelementary interface t_{m-1}. The variational formulation of such a jump term is chosen as:

$$\psi\left(t_{m-1}^{+}\right) \int_{\mathcal{T}_h} \varphi_i \left(u_h(t_{m-1}^{+}) - u_h(t_{m-1}^{-})\right) dx, \qquad (30)$$

where the superscripts + and − denote the two sides of the interelementary interface shared by two adjacent temporal elements I_{m-1} and I_m (see Fig. 2). The jump term (30) takes a consistent form in the spatial FE subspace defined before. Thus it is multiplied by the spatial test function φ_i and integrated over the spatial domain $\mathcal{T}_h$. Let the temporal node r representing t_{m-1}^{-} and s representing t_{m-1}^{+}, the jump term (30) becomes

$$\psi_s\left(t_{m-1}^{+}\right) \int_{\mathcal{T}_h} \varphi_i \varphi_j dx \sum_{j=1}^{n_s} \left(\xi_{js} - \xi_{jr}\right) \qquad (31)$$

and in matrix form:

$$\begin{bmatrix} \psi_s\left(t_{m-1}^{+}\right) \boldsymbol{M}_{11} & & \\ & \ddots & \\ & & \psi_s\left(t_{m-1}^{+}\right) \boldsymbol{M}_{n_s n_s} \end{bmatrix} \left(\boldsymbol{\xi}^s - \boldsymbol{\xi}^r\right). \qquad (32)$$

Here, $\psi_s\left(t_{m-1}^{+}\right)$ actually equals 1 above. However, we need the index s to know which block of the linear system (27) the jump term should be added to. The reason for choosing $\psi_s\left(t_{m-1}^{+}\right)$ in the jump term Eq. 30 is that we can still decouple the block linear system, by sequentially solving each subsystem starting from I_1 with the given initial condition $\boldsymbol{\xi}^0$. An example of sequentially solving the block linear system is presented in Appendix A.

Complementing (27) by jump terms at all interelementary interfaces, we get the final block linear system which represents the consistent variational formulation with jump terms in the whole space-time domain.

3.3.4 Other time integration schemes for comparison

We note that all TDG schemes are unconditionally stable. The special case TDG(0) is equivalent to the well-known implicit BE method. The CN time discretization scheme is also used in this work as a higher-order scheme for comparison purposes. As it is a standard scheme, it will not be described further.

4 Solution algorithm

We use structured elements to generate continuum meshes for fracture networks. The small length scale of fractures is resolved by local quadtree-refinements and thus the resulting mesh can well represent the topology of the fracture network. Compared to the work in [10], the final mesh does not need to be converted into a triangular mesh and can be straightforwardly applied to the FEM solver [56].

Algorithm 1 illustrates the workflow of generating a mesh pattern with fractures. There are in total three refining stages. The first generates a uniform mesh without any hanging nodes. The second stage locally refines the elements which are intersected by fractures. This step is done `N` times until the finest element size can adequately resolve the small length scale of fractures. Next, those finest elements intersected by fractures are marked as fracture elements. The last

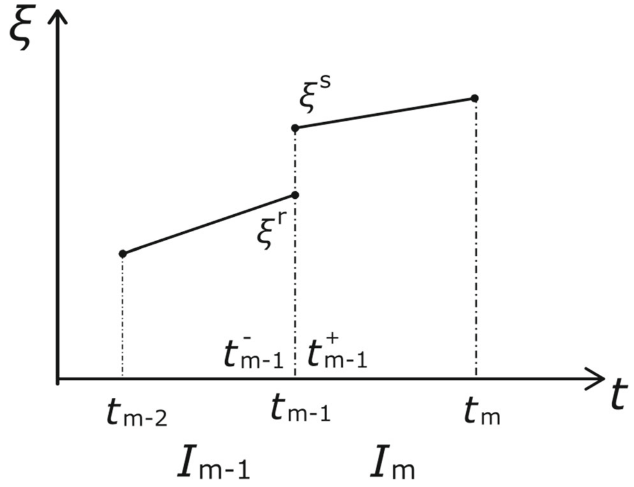


Fig. 2 Illustration of the jump term at t_{m-1} for TDG(1). I_{m-1} and I_m present two adjacent time intervals on either side of the temporal node t_{m-1}. t^-_{m-1} and t^+_{m-1} are temporal nodes defined as the left and right limits of t_{m-1}. $\boldsymbol{\xi}^r$ and $\boldsymbol{\xi}^s$ are the solution vectors on t^-_{m-1} and t^+_{m-1}, respectively, indicating the discontinuous solution at t_{m-1}

refinement step is optional if higher-resolution results are needed, e.g. reference solutions.

Algorithm 2 presents the workflow for solving flow and transport problems. The stationary groundwater flow is first solved by the mixed finite element method [49–51], which provides the locally conservative property and a continuous velocity field. More details about assembling the block linear system with jump terms and efficient solver strategy are demonstrated in Appendix A and B respectively.

Algorithm 1 Generate mesh with fractures.

```
Generate uniform mesh given a global level
Set a required local refinement level N
Randomly generate fractures
for n = 1 : N do
    for each element do
        for each fracture do
            if fracture ∩ element then
                Set refinement flag
            end if
        end for
    end for
    Refine flagged elements
end for
for each element do
    for each fracture do
        if fracture ∩ element then
            Mark as fracture element
        else
            Mark as matrix element
        end if
    end for
end for
Further refine mesh if needed
```

Algorithm 2 Solve flow and transport problems.

```
Solve the stationary groundwater flow Eq. 1
Calculate advection coefficients by the formula 2
Spatial semi-discretization of the transport Eq. 3 either with the DG
  method (weak formulations 8, 9 and 12) or the SUPG method (the
  weak formulation 18), yielding the ODE system Eq. 19
Setup the time step length $\Delta t$
Time discretization of the ODE system Eq. 19 based on the TDG
  weak formulation 24, yielding the block system Eq. 27
Including jump terms Eq. 32 into the system Eq. 27
Decouple the block linear system with jump terms in a temporally
  sequential manner (see Appendix A)
Assign the initial condition of the transport Eq. 3
for t = $t_0$ : $\Delta t$ : $t_0 + T$ do
    Iteratively solve the block linear system in the current time inter-
      val (see Appendix B)
    Save the transport state
    Update the transport state as the initial condition for the next
      time interval
end for
```

All source codes are developed in C++ based on the FEM library deal.II [56].

5 Verification and convergence study of the different schemes

Before transport in fractured media is considered, a benchmark advection-diffusion problem in a homogeneous domain [57] is considered as validation and to test the properties of the implemented methods. The test case describes advection in a circle flow field and diffusion of a 2D Gaussian hat-shaped initial concentration distribution. We use a dimensionless formulation of this general case. The advection coefficient $\boldsymbol{a}$ equals $(-4y, 4x)$. Regarding the diffusion coefficient, we choose two different isotropic homogeneous cases:

1. case with strong diffusion: $|\ \boldsymbol{D}\ |= 10^{-2}$
2. advection dominated case: $|\ \boldsymbol{D}\ |= 10^{-8}$.

The Gaussian initial condition reads

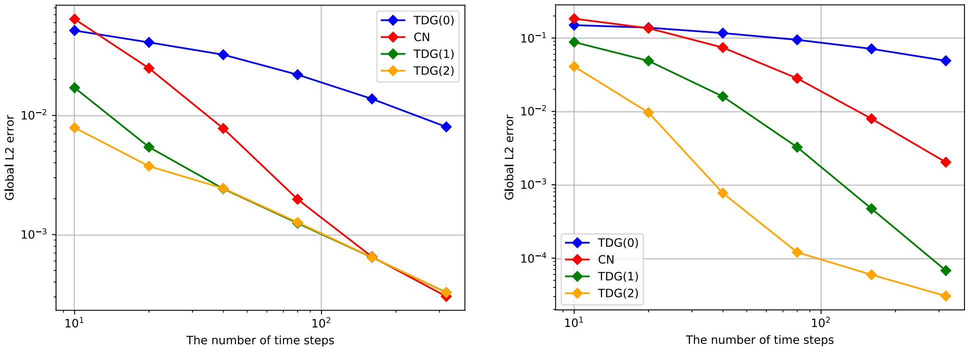
$$u\,(x, y, t_0) = \exp\left(-\frac{(x+0.2)^2 + y^2}{2\sigma^2}\right), \tag{33}$$

where the width of the distribution σ is set to 0.1. The exact solution given in [57] reads

$$u_{ex} = \frac{2\sigma^2}{2\sigma^2 + 4\ |\ \boldsymbol{D}\ |\ t}\exp\left(-\frac{(\tilde{x}+0.2)^2 + (\tilde{y})^2}{2\sigma^2 + 4\ |\ \boldsymbol{D}\ |\ t}\right), \tag{34}$$

where $\tilde{x} = x\cos(4t) + y\sin(4t)$ and $\tilde{y} = -x\sin(4t) + y\cos(4t)$. Since the exact solution holds for the infinite spatial domain, the simulation domain is bounded by

Fig. 3 Convergence study of different time schemes. Left: $|\boldsymbol{D}| = 10^{-2}$. Right: $|\boldsymbol{D}| = 10^{-8}$



$[-0.5, -0.5] \times [0.5, 0.5]$, and the Dirichlet boundary condition is computed from the exact solution.

The error is defined as the L_2 norm of the difference between the numerical solution and the exact solution at the last time step:

$$\varepsilon = \|u_h - u_{ex}\|_{L_2} = \sqrt{\int_{\Omega} (u_h - u_{ex})^2 \, dx}. \tag{35}$$

The convergence results of different TDG schemes and the CN method are shown in Fig. 3. From the figures we see that all investigated schemes converge for both cases. The error of the TDG(0) scheme is the highest in both aspects: error and convergence rate. The numerical error caused by TDG(1) is lower than that by CN. Also, the convergence rate of TDG(1) is better than that of CN for both cases, especially for the advection-dominated case. Though the error produced by TDG(2) is not higher than that by TDG(1), TDG(2) does not greatly improve the convergence rate, notably when the diffusion term dominates. The reason is that the very high resolution in space needed to achieve the optimal convergence rate of TDG(2) is not feasible. TDG(2) is considered a very high-order scheme in time; see [58]. Thus it is not surprising that convergence curves of TDG(1) and TDG(2) overlap given a fine time step.

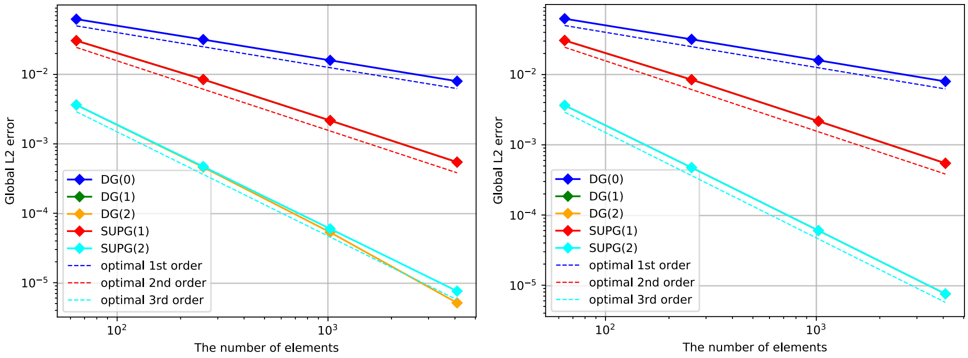
The results of the grid convergence study for different spatial schemes are shown in Fig. 4. For both the strong diffusion and the advection-dominated case, all spatial schemes optimally converge at the rate $p_s + 1$ in the L_2 norm. Note that there is hardly any difference of L_2 errors between SUPG and DG schemes in this convergence study, especially for the advection-dominated case.

The implementation of the numerical methods is within the limits of feasibility verified by this case.

6 Comparing different schemes for transport in fractured media

As outlined before, the goal of this work is to test and compare the benefit of higher-order space-time discretization schemes for transport modeling in fractured media. For this purpose, we consider three numerical examples in this work. The first example is a transport problem in fractured media, in which the fracture network is highly regular and prototypical. Due to the orthogonal structure of the fracture network, a grid with adaptive refinement to resolve the interfaces well can be constructed without problems. The second and third examples are extensions of the first one, in which a large number of fractures are randomly placed. Thus the cases are more realistic but do not allow resolving the interfaces in the same detail as in the first case. Although there is still grid refinement, the medium is in this way rather represented as a highly heterogeneous domain. This setup is chosen to test if restrictions found for transport modeling in a domain with a clear distinction between fracture and matrix are weakened if there

Fig. 4 Convergence study of different spatial schemes. Left: $|\boldsymbol{D}| = 10^{-2}$. Right: $|\boldsymbol{D}| = 10^{-8}$



is a wide range of matrix block sizes, in particular smaller ones. We consider here a diffusive and a less diffusive problem. The global average L_2 errors are estimated separately in the fracture and matrix domain between different schemes. To assess the impact of errors on cumulative quantities, we compare local inverse breakthrough curves (BTC) at different points and global ones over cross-sections (note that we use the abbreviation BTC for the inverse breakthrough curve). We plot inverse BTCs on a logarithmic scale in the whole work for better visualizations. Apart from the errors in general, a particular focus is on the generation of spurious oscillations (over- and undershoots). We consider different measures to assess this question. The computation times of different methods are also compared for the test cases using the wall times.

Details of the model setups and the error definitions are outlined in this section. For the fractured media cases, we use a formulation with units.

6.1 Benchmark fracture network

This test case is a solute transport problem based on the regular fracture network of [59], with slightly different setups. Note that the model with different measures was also used as a benchmark case for flow by [60], however, we focus here on the transport process. In this geometry, the fracture network and the shapes of matrix blocks are simple enough to identify the relevant transport time scales. We focus here on time discretization, while different space discretization schemes are considered in the next example.

The domain has a span of 1 m by 1 m. The mesh is locally refined near the fractures (see Fig. 5). The global refinement level is set to be 5 and the local level equals 4. As a result, all fractures span two finite elements in the transverse direction. They have an aperture of $3.9 \times 10^{-3}\,m$. All model parameters are listed in Table 1. We simulate sudden tracer injection from the left boundary with a constant flow rate of $10^{-5}\,ms^{-1}$ and concentration of $c_L = 100\%$. The upper and lower boundaries are no-flow boundaries. The initial condition is a concentration of zero in the whole domain.

This model is considered to be advection-dominated in the fractures, where Peclet numbers reach up to $Pe = 100$, and diffusion-dominated in the matrix blocks, where Peclet numbers are $Pe = 0.005$ with a definition of the Peclet number for the length of the domain. The simulation runs with an initial time step of 432 seconds until 1 day, after that, the time step is enlarged to 1 day until 1200 days.

As there is no analytical solution to this problem, a numerical reference solution is generated. The time step for this is eight times finer in the early time and four times finer afterward, with the TDG(0) scheme being used. SUPG(2) is chosen for space discretization as a higher-order scheme that is not so prone to spurious oscillations (found in Section 6.2). We compare here only the TDG(0) (backward Euler), the TDG(1), and the CN scheme, as the TDG(2) scheme did not lead to much smaller errors in the validation test.

Figure 5 shows the flow result (left) and the concentration distribution at t = 400 days (right). This figure is in analogy to the results shown in [59].

To evaluate the schemes, we investigate only global measures for this case. The flux-weighted inverse breakthrough curve on the outflow boundary is considered as one measure. This is according to [59], however, we focus here on earlier times. It allows for assigning different processes to different time regimes, see for example [61]. We also investigate the solution accuracy averaged over the domain. Two error measures are defined to indicate the error separately in the fracture and matrix domains:

1. average L_2 error in the fracture domain:

$$\varepsilon_f = \frac{\sqrt{\int_{\Omega_f} \left(u_h - u_{ref}\right)^2 dx}}{\int_{\Omega_f} 1 dx} \tag{36}$$

2. average L_2 error in the matrix domain:

$$\varepsilon_m = \frac{\sqrt{\int_{\Omega_m} \left(u_h - u_{ref}\right)^2 dx}}{\int_{\Omega_m} 1 dx}. \tag{37}$$

The breakthrough time in the BTC is determined by the transport time scale for the fastest connection in the fracture network (here expected to be determined by advection). A fast drop after the breakthrough is determined by the macrodispersion in the fast flow regimes, here in the backbone of the fracture network. If there is a clear separation of time scales for processes in the fractures and in the matrix, and there is only one matrix block size, the fast decrease is followed by a tailing with a power of $-1/2$, which comes from the diffusion in the matrix blocks (stagnant zones). After reaching the diffusive time scale, an exponential decrease is obtained (matrix is filled). If there are different shapes and sizes of matrix blocks, different diffusion times are assigned. The BTC shows then a superposition of the different power law contributions, leading to a different power law. If models are used to investigate such a behavior, it is important that the full span of processes time scales is well represented.

With a mean velocity in the fastest path of $v = 0.002\,m/s$, the fastest advective transport time is at $t_a = 500\,s$. This is well reproduced in the BTC (Fig. 6) by all discretization schemes. After the breakthrough, a sharp drop-down of the concentration (1 - u) is observed at the early time. This is well reproduced by the TDG(1) and the CN scheme (except for the spurious oscillations), while the lower order

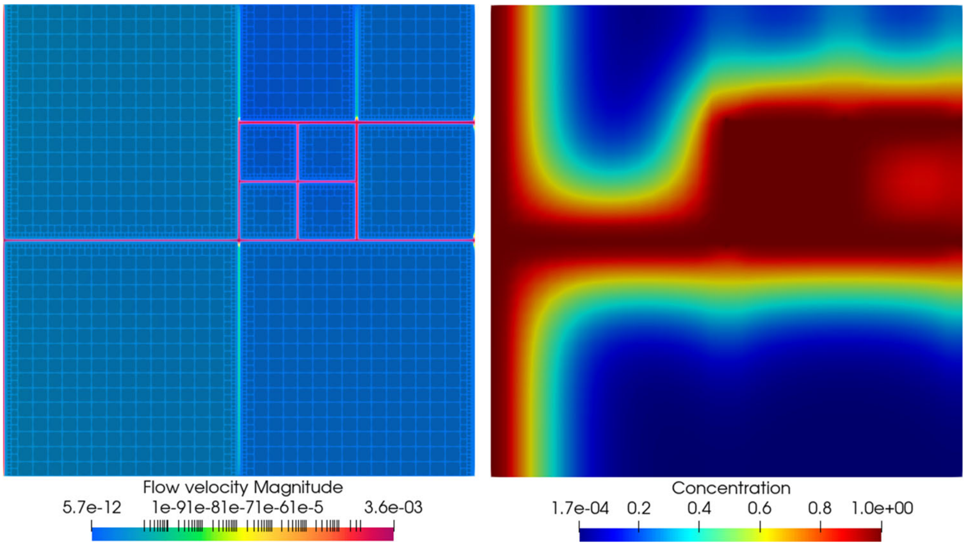


Fig. 5 Left: flow result and the mesh (test case 1). Right: concentration distribution [-] at t = 400 days (test case 1)

scheme does not capture the slope as good. This over-diffusive behavior is expected. The CN scheme indicates that the advective moving of the concentration front in the fracture in the early breakthrough time causes non-monotonic behavior. This behavior fades away when the diffusive time scale comes into effect. The time scale for the diffusive filling of the smallest blocks can be estimated to $t_{d,\,\min} = 10^6 s$. Accordingly, after the breakthrough, the BTC is characterized by a $-1/2$ slope. The first time where a deviation is expected is at $t \approx 10^6 s$. Although the crossover is not that sharp, the slope of the BTC steepens in this range, as is expected. This is reproduced well by all schemes.

The errors for different time schemes at t = 1 day and t = 600 days are listed in Table 2. It is not surprising that the error dominates in the fracture domain due to the dominance of the advection term. TDG(0) yields larger errors than TDG(1) because of numerical dissipation. CN results in extraordinarily large errors in the fracture domain, attributed to non-monotonic behavior (see also Fig. 6). In summary, the TDG(1) scheme produces monotonic solutions while maintaining a satisfactory solution accuracy.

Table 1 Model parameters (test case 1)

Parameter	Value	Unit
Permeability matrix	1×10^{-12}	m^2
Permeability fracture	8.3×10^{-5}	m^2
Viscosity	2.98×10^{-3}	$Pa \cdot s$
Porosity matrix	0.25	–
Porosity fracture	1.0	–
Diffusion coefficient	10^{-10}	$m^2 \cdot s^{-1}$
Longitudinal dispersivity	10^{-2}	m
Transverse dispersivity	10^{-3}	m

To further evaluate the monotonicity of the solution, we compute the total variation [62] in higher dimensions over time using the following definition:

$$TV = \sum_K \max \mid u_i - u_j \mid, \tag{38}$$

where i and j are different indices of spatial DoF on the element K. The maximum difference of the scalar quantity u is initially evaluated within the element K, and the total variation is the summation of these differences taken over all elements or a subset of elements. Since this definition does not take the cell size into account, it can only be employed to compare solutions on the same grid. As an increase of TV over time is here not only created by spurious oscillations but also due to the generation of a sharp front as the concentration evolves, the global TV is difficult to interpret. Instead, we evaluate TV locally in the fracture domain and within a small rectangular zone measuring $0.01m$ by $0.01m$, which surrounds the central point (0.5, 0.5). The results are depicted in Fig. 7. Lines below the reference solution are indicative of numerical dissipation, while lines above suggest non-monotonic behavior. It is observed in the left figure that both CN and TDG(1) result in numerical oscillations at

Table 2 Errors for different time discretization schemes and computation times (test case 1)

Scheme	t = 1 day		t = 600 days		Wall time
	ε_f	ε_m	ε_f	ε_m	[min]
TDG(0)	3.9e-4	1.5e-4	4.7e-4	1.2e-4	13
TDG(1)	5.5e-5	2.2e-5	1.6e-4	4.0e-5	37 (23)
CN	1.5e-3	1.7e-4	1.4e-3	4.1e-5	13
Reference	–	–	–	–	58

In parentheses, the wall time of TDG with the fixed-point iteration is shown

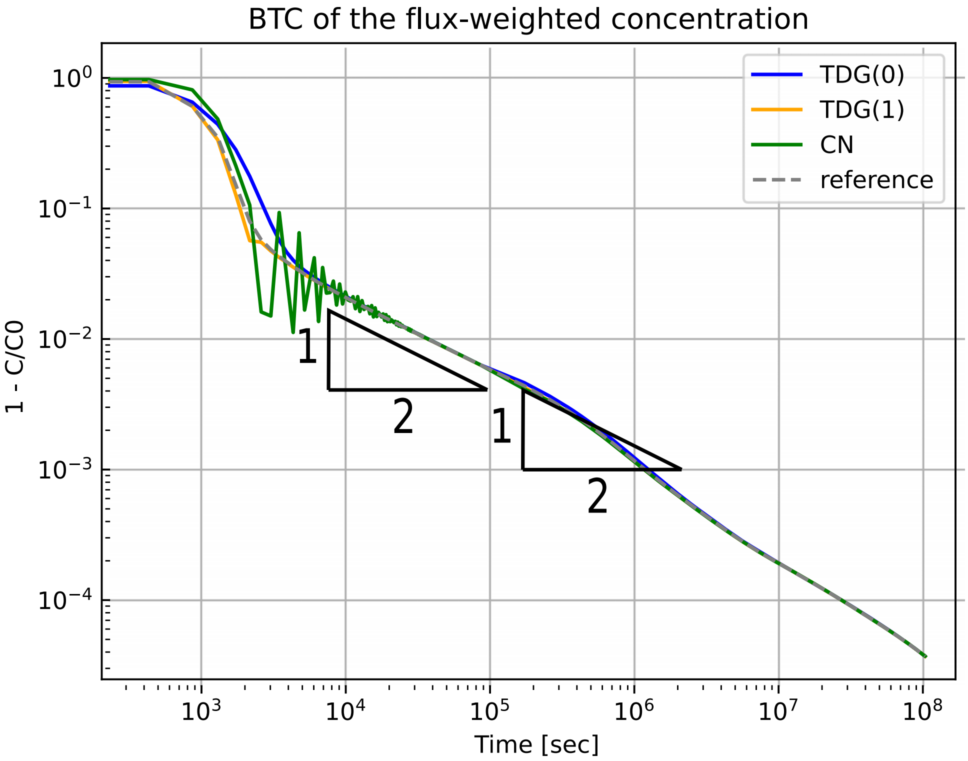


Fig. 6 Inverse breakthrough curve of the flux-weighted concentration for different time discretization schemes (test case 1)

early times. However, the non-monotonic behavior is quickly mitigated for TDG(1), while it persists with a high magnitude for a prolonged period in the case of CN. In contrast, TDG(0) exhibits a significant numerical dissipation.

It should be noted that while there are differences between the schemes in the early time behavior, all schemes show similar behavior in the later time. This is not surprising because the transport process in the later time is dominated by the diffusion exchange between fracture and matrix.

Table 2 presents the computation times for various time discretization schemes. TDG(0) requires the least computation time. Directly solving the block linear system of TDG(1) can be several times more costly than that of TDG(0). By employing a fixed-point iteration, the computation time is reduced from 37 minutes to 23 minutes. The computation time of the iteration method, of course, depends on the stopping criterion. We observe that a relative tolerance of 10^{-3} is an appropriate value to halt the iteration, ensuring accuracy (see Appendix B). Especially in the later transport times, only two or three iteration steps are needed. Compared to the reference solution, TDG(1) produces an equally robust solution while maintaining low computation times.

6.2 Complex fracture network

For a second transport problem in fractured media, we simulate a heat transport problem with a complex fracture network. In this problem, we analyze the space discretization schemes and the time discretization schemes. For the comparison of the space discretization schemes, TDG(1) was used for time discretization and DG(1) was used as

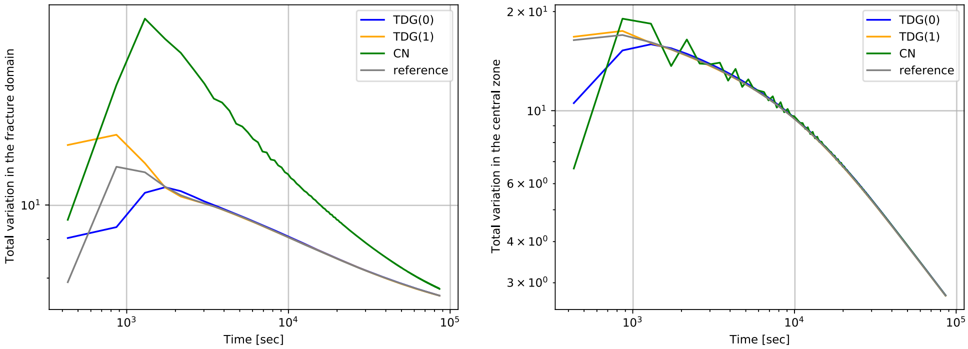


Fig. 7 Total variation over time (test case 1). Left: in the fracture domain. Right: in the central zone spanning 0.01 m by 0.01 m

space discretization scheme to evaluate the different time discretization schemes. Although the setup is still prototypical, it is more realistic than the previous one in that it contains a full spectrum of matrix blocks of different sizes that are not all isolated. The clear distinction between advective and diffusive transport is here no longer expected. With a heat transport considered with realistic parameters, diffusive transport is stronger than in the previous example. For this reason, it is expected that the numerical problems in the previous example do not persist. It is possible that the problems occur locally, but do not have a strong influence on the cumulative transport behavior (such as a BTC over a cross-section studied in the previous example).

In this problem, 50 fractures are randomly placed in a spatial domain of $16m \times 8\,m$ (Fig. 8). Model parameters are shown in Table 3. Hot water of 100 degrees Celsius is injected from the left side, with the flow being driven by two Dirichlet boundaries from left $h = 92\,m$ to right $h = 90\,m$. The upper and lower boundaries are no-flow boundaries. The initial temperature field equals zero degrees Celsius all over. The simulation runs for 50 days with a uniform time step of 8 hours. In this problem the grid no longer allows for a very sharp resolution of the interfaces.

Figure 9 presents the reference simulation results of the flow velocity and the temperature distribution. A highly heterogeneous velocity field is observed, but the transport in the matrix is diffusive. The temperature result well reflects the preferential flow paths of the heat fluxes.

To evaluate this test example, we use local and global measures. We use the averaged error as defined in the previous subsection. For the local measures, four observation points (Fig. 8) are selected for plotting BTCs:

$$P_1 = (1.46, 3.3)$$
$$P_2 = (1.46, 3.65)$$
$$P_3 = (8, 3.5)$$
$$P_4 = (8, 3)\,.$$

P_1 is located in a fracture connection and very close to the inflow boundary. P_2 is near the point P_1 but located in the matrix domain. The points P_3 and P_4 are also in a fracture connection and in the matrix domain, but with a larger distance to the inflow boundary. For this example, we only consider the inverse breakthrough curves at the locations close to the inlet, while the other ones are used in the next example. In the next example, as additional global measures, average BTCs over the cross-section A with $x = 1.46\,m$ and the cross-section B with $x = 8\,m$ are investigated.

Table 3 Model parameters (test case 2)

Parameter	Value	Unit
Specific storage	10^{-6}	m^{-1}
Hydraulic conductivity fracture	2×10^{-3}	ms^{-1}
Hydraulic conductivity matrix	2×10^{-7}	ms^{-1}
Porosity	0.375	–
Density capacity solid	1.4×10^{6}	$Jm^{-3}K^{-1}$
Density capacity water	4.17×10^{6}	$Jm^{-3}K^{-1}$
Diffusion coefficient solid	1	$Wm^{-1}K^{-1}$
Diffusion coefficient water	0.598	$Wm^{-1}K^{-1}$

The errors (see Eqs. 36 and 37) of different space and time discretization schemes are given in Tables 4 and 5 at t = 25 and 50 days. It holds for all schemes that the error dominates in the fracture domain. The reason is the internal sharp gradients at the interfaces of fractures and the matrix.

Considering the space discretization schemes, as expected, DG(0) has the highest errors among the spatial schemes due to the high numerical dissipation. We observe that DG(2) and SUPG(2) lead to higher errors than the other schemes, in particular DG(2). This is due to spurious oscillations that are stronger in the schemes with higher-order polynomial degree (see also Fig. 10). By comparing DG(1) and SUPG(1) schemes, we note that both schemes generate similar errors. Regarding the time discretization schemes, errors are a bit lower using CN than using TDG(0) in the matrix domain. TDG(1) has the lowest global errors both in the fracture and matrix domains. All in all, the errors produced by the different time discretization schemes do not differ so much.

The spurious oscillations can be evaluated more closely with the local observations. The BTCs for the observation points close to the inflow boundary are shown in Fig. 10. The other BTCs for this case are not shown, as they do not show new aspects. For the space discretization, the BTCs obtained with DG(0) show high numerical dissipation. Otherwise, the schemes do not differ much, except for the DG(2) scheme,

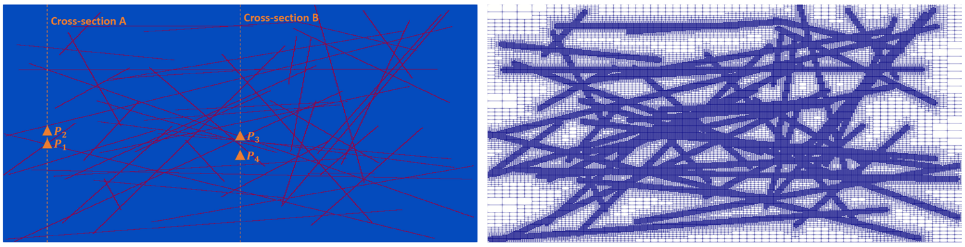


Fig. 8 Left: fracture network and observation points (test cases 2 and 3). Right: the corresponding mesh (test cases 2 and 3)

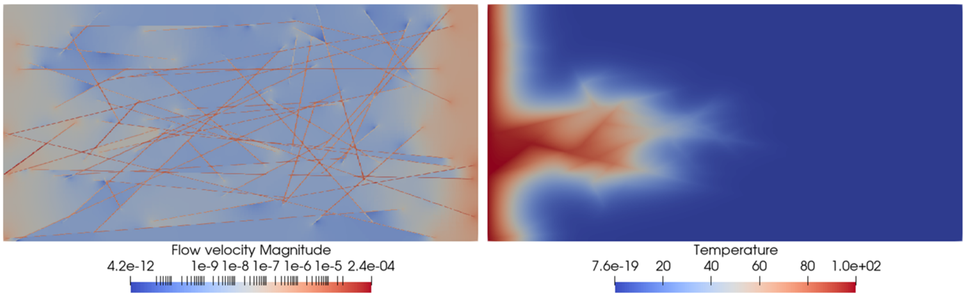


Fig. 9 Left: flow result (test cases 2 and 3). Right: temperature distribution [°C] at t = 15 days (test case 2)

which leads to a strong deviation of the BTC in both domains. This is due to persistent spurious oscillations generated with DG(2). An example of the solution at a given time is shown at the bottom of Fig. 10. For the time discretization, in the early breakthrough time of the fracture domain, the TDG(0) scheme does not match well the reference solution (Fig. 10b). The CN scheme produces even slightly non-monotonic solutions in the early time. In the late time, all temporal schemes show a similar behavior. In the matrix domain, all temporal curves match well with each other in the entire time scale (Fig. 10d).

For this example, it can be concluded that the space discretization schemes with polynomial order 2, in particular DG(2), and the CN scheme for time discretization, cause spurious oscillations. DG(0) for space discretization causes stronger numerical dissipation. Otherwise, the schemes did not lead to larger differences in terms of BTCs or generation of spurious oscillations. The case is quite diffusive, which also dampens the preferential transport in the fractures (seen in the snapshot in Fig. 9). For this reason, in the next section, the schemes are compared for a more advective case.

The computation times of different schemes are shown in Tables 4 and 5. As expected, DG(0) is the least costly spatial scheme. A spatial DG scheme is much more costly than a SUPG scheme given the same polynomial order. A second-order spatial scheme can be several times more costly than the corresponding first-order scheme. Considering the temporal schemes, TDG(0) and CN have an equivalent computation time. TDG(1) is not that expensive if the fixed-point iteration method (see Appendix B) is used: the computation time of TDG(1) is reduced from 153 minutes to 54 minutes and leads to only a factor of 2.4 between the other two schemes.

6.3 Complex fracture network with a smaller heat conductivity and larger time steps

To test and compare the schemes for more advective transport problems with respect to the numerical errors, we change the heat conductivity in the complex fracture network, described in the last test case, by a factor of 100. Also, we increase the time discretization steps to make time discretization more challenging. Such settings would be needed if a process on large length scales would be considered and the conditions would be relevant near wells. The time discretization steps are enlarged to 1 day for the reference solution (TDG(0) used) and 5 days for others. The simulation runs for 600 days. Other model setups are kept the same as in the last test case.

As the problem of the spurious oscillations in the space discretization schemes can for this case be best visualized with a snapshot of the solution, we show such snapshots for the space discretization schemes. The time discretization schemes are evaluated with the same measures as in the last example. In this case, the BTCs at all locations (see Section 6.2) are investigated.

We present the temperature results for SUPG(1), DG(1), and DG(0) in Fig. 11 to evaluate the space discretization schemes. DG(2) and SUPG(2) are here not considered, as they showed spurious oscillations in the more diffusive case already. The significant temperature contrast between fractures and the surrounding matrix is here much more pronounced (cf. Fig. 9). Here, also the space discretiza-

Table 4 Global errors for different space discretization schemes and computation times (test case 2)

Scheme	t = 25 days		t = 50 days		Wall time
	ε_f	ε_m	ε_f	ε_m	[min]
DG(0)	2.3e0	6.0e–1	2.9e0	7.8e–1	7
DG(1)	1.1e–1	2.2e–2	1.3e–1	2.9e–2	153
DG(2)	4.5e0	4.5e–1	4.7e0	5.0e–1	1553
SUPG(1)	1.1e–1	2.2e–2	1.4e–1	2.9e–2	14
SUPG(2)	3.6e–1	6.7e–2	3.3e–1	6.9e–2	92

Table 5 Global errors for different time discretization schemes and computation times (test case 2)

Scheme	t = 25 days		t = 50 days		Wall time
	ε_f	ε_m	ε_f	ε_m	[min]
TDG(0)	1.5e–1	2.8e–2	1.9e–1	3.6e–2	22
TDG(1)	1.1e–1	2.2e–2	1.3e–1	2.9e–2	153 (54)
CN	2.7e–1	2.2e–2	2.3e–1	2.9e–2	23

In parentheses, the wall time of TDG with the fixed-point iteration is shown

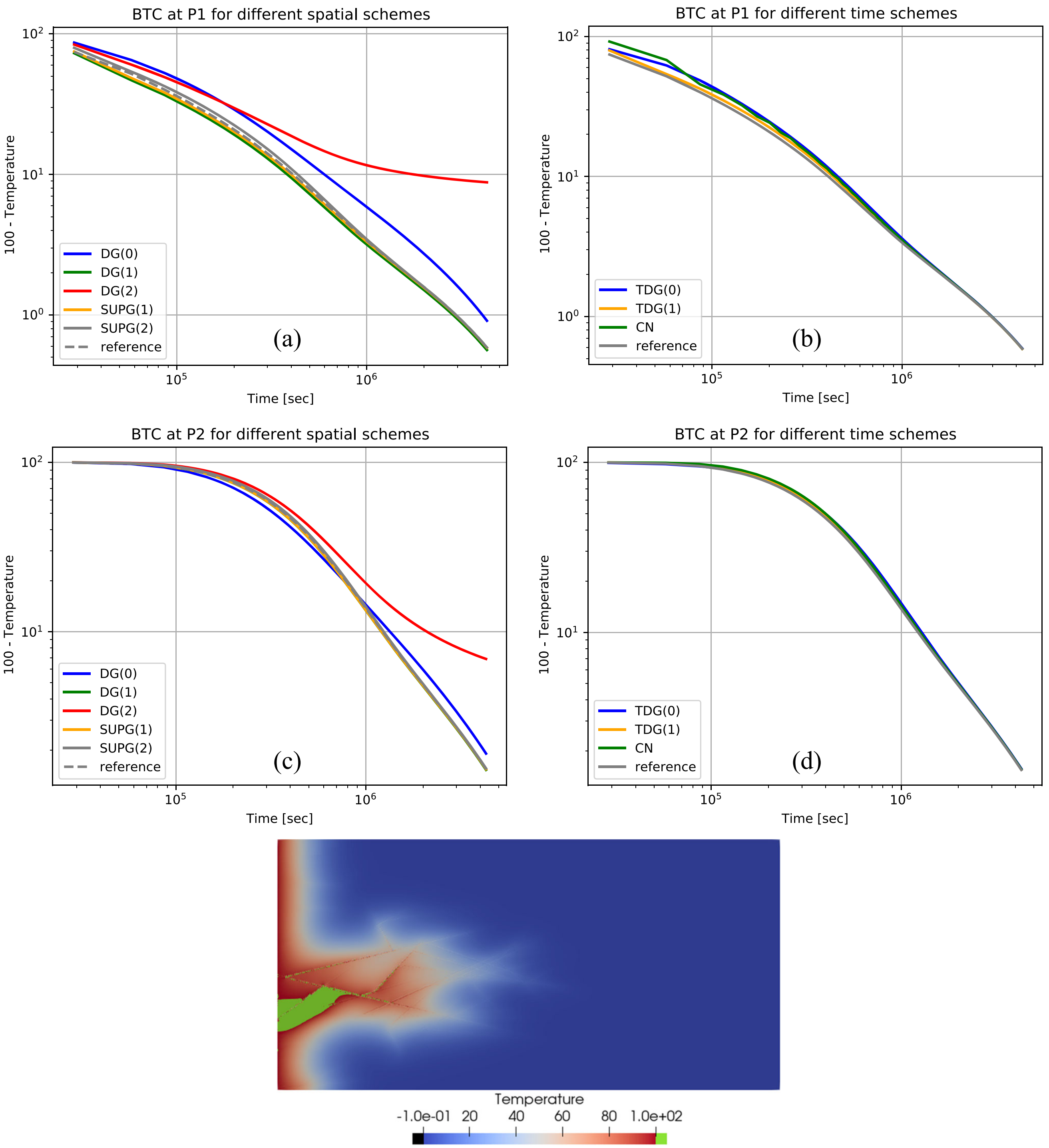


Fig. 10 Top: inverse breakthrough curves in the logarithmic scale close to the inflow boundary, for different schemes used in the second test case (a to d). Bottom: non-monotonic solution at t = 15 days given by DG(2), under- and overshoots are colored in black and green respectively (test case 2)

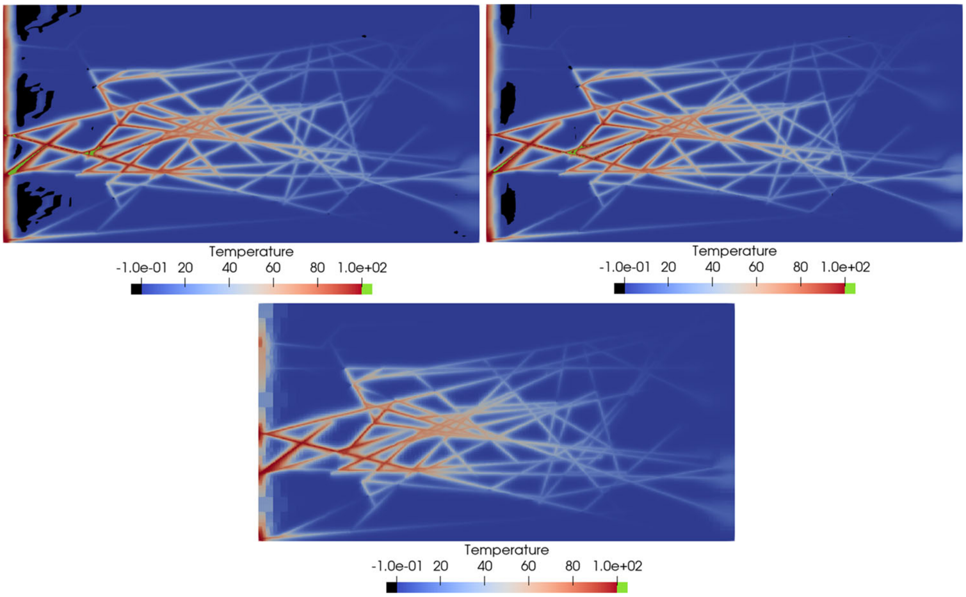


Fig. 11 Temperature distribution [°C] at t = 15 days (test case 3). Upper left: SUPG(1). Upper right: DG(1). Lower: DG(0)

tion schemes with polynomial degree one create spurious oscillations. Both SUPG(1) and DG(1) display undershoots (colored in black) in matrix blocks and overshoots (colored in green) in fractures. Additionally, it is observed that SUPG(1) produces spurious oscillations characterized by a wiggling structure. In contrast, DG(1) effectively prevents the spread of numerical oscillations. As only DG(0) produces oscillation-free results, we use it as the reference spatial scheme for further investigation of time schemes.

The errors for the different time discretization schemes are shown in Table 6. The high errors for the CN scheme are due to the strong numerical oscillations. To evaluate the numerical oscillations, the BTCs for all points and cross sections are presented in Fig. 12. As anticipated, the CN method exhibits strong numerical oscillations for the observation points within the fracture domain (Fig. 12a and c). Slight discrepancies between the low-order TDG(0) scheme and the reference solution are observed at observation points 1 to 3 (Fig. 12a, b, and c). In contrast, the TDG(1) scheme remains stable and consistently aligns well with the reference solution, irrespective of the observation point location. However, oscillations of the CN scheme are restricted to fractures, so that even in this example, the local deviations do not strongly impact the cumulative transport behavior over the cross-sections (Fig. 12e and f).

The computation times are also presented in Table 6. By using the fixed-point iteration method, the computation time of TDG(1) is reduced from 8 minutes to 4 minutes. Similar to the previous example, TDG(1) requires more computation time compared to TDG(0) and CN, but only by a factor of 1.3. In comparison with the reference solution, TDG(1) is deemed highly efficient.

Table 6 Global errors for different time discretization schemes and computation times (test case 3)

Scheme	t = 15 days		t = 300 days		Wall time
	ε_f	ε_m	ε_f	ε_m	[min]
TDG(0)	8.8e–1	1.5e–1	6.1e–2	2.5e–2	3
TDG(1)	2.6e–1	4.5e–2	1.5e–2	6.0e–3	8 (4)
CN	3.5e0	1.6e–1	1.4e0	1.4e–2	3
Reference	–	–	–	–	13

In parentheses, the wall time of TDG with the fixed-point iteration is shown

7 Discussion

The test cases in the previous sections revealed different numerical problems. Numerical dissipation is reflected in the accuracy of the solutions and is relevant for all transport problems with strong advections. For transport in fractured media, it is not only challenging to get a good accuracy of the solution, but also to prevent spurious oscillations, which can be a serious difficulty in the application of transport models.

In the validation problem that was considered to study the convergence, all discretization schemes generated monotonous solutions without spurious oscillations, also in the advection-dominated case. In this problem, the velocity field is smooth and does not have strong gradients. In the test cases with fractured media, due to the strong parameter contrast, velocity fields are very heterogeneous and have strong gradients at

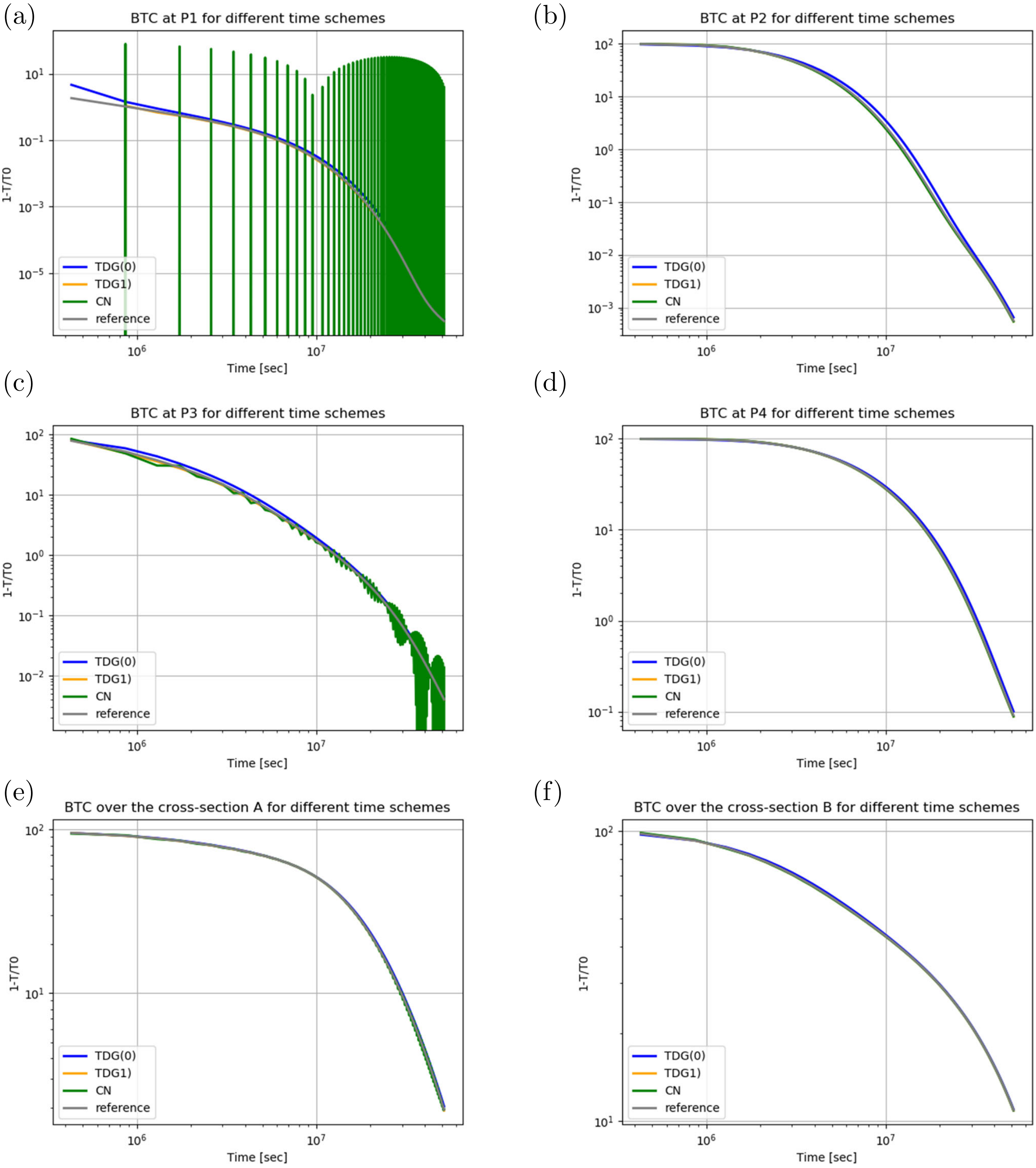


Fig. 12 Inverse breakthrough curves in logarithmic scale for different time schemes (test case 3)

interfaces. In these test cases, the problem of spurious oscillations became apparent.

The different space discretization schemes were only tested for the more complex fracture network cases. The solutions of the lowest-order DG(0) scheme were always monotonic, however, the strong numerical dissipation compared to all higher-order schemes was here clearly visible in the BTCs. The schemes with polynomial degree two caused spurious oscillations already for the more diffusive case (with realistic parameters). For the DG(2) scheme they were so strong that not only the global error of the solution was high (cf. Table 4), but also local information obtained in a BTC was strongly affected, no matter where in the domain the BTC was taken. For the SUPG(2) scheme, the oscillations were weaker, but strong enough to cause higher averaged errors than the schemes with lower polynomial degree. Although DG(1) and SUPG(1) lead to better results than the lowest-order DG(0) scheme in this case, for a more advective problem, the last example, also these schemes caused strong spurious oscillations. In this case, they were stronger with SUPG(1) than with the DG(1) scheme. Taking this into consideration, DG and SUPG schemes cannot well be rated against each other.

When comparing different time discretization schemes, the TDG scheme was clearly of advantage compared to a comparable standard scheme (here CN). This could best be illustrated with the benchmark fracture network, where advection and diffusion processes could be reflected in the BTCs. For the early-time behavior, the numerical dissipation leads to large errors, seen in strong differences between the TDG(0) (backward Euler) and other schemes. The CN scheme leads to spurious oscillations in the early time regime, when advection in the fracture dominates and creates sharp gradients of the solution. Conversely, in the late time regime, where the diffusive processes are relevant, the differences between the solutions generated with all schemes got very small. The spurious oscillations generated with the CN scheme were always present in the fracture network test examples and became very large in the case of strong advections (the last example in Section 6.3).

For the artificial benchmark network, it would be possible to anticipate the strong advection leading to spurious oscillations with transport time scales. For the more complex fracture networks, this would not be possible, as time scales are not clearly identifiable and not clearly distinct.

If the spurious oscillations deteriorate the results depends also on the quantities of interest. For example, the spurious oscillations in the benchmark example or in the fracture network created with the CN scheme are local, inside of the fracture or close to the inlet boundary, and do not affect, for example, BTCs taken in the matrix domain or over cross-sections (cf. BTCs at cross sections in Fig. 12). They did, however, show up in the averaged errors, which were high in the fracture domain when using the CN scheme. This does, however, not hold for the space discretization schemes of polynomial degree 2. Spurious oscillations became relevant for any measure in the fracture domain for these schemes (cf. errors in Table 4).

8 Conclusions

In this study, we performed a comparative study between discontinuous and continuous space and time Galerkin discretization methods for solving transport problems in fractured media. Large parameter contrasts result in large velocity differences, making this problem prone to the generation of spurious oscillations. In Section 3, we first outlined both the DG and SUPG discretizations for space, as well as the TDG discretization for time. In Section 4, the solution algorithms were introduced, specially, the decoupling strategy and the fixed-point iteration for TDG, see also Appendices A and B. The schemes for space and time discretizations were verified in Section 5, achieving optimal convergence rates for all space discretization schemes. In contrast, optimal convergence rates in high-order time schemes were hardly achieved because sufficient high resolutions in space were unfeasible.

The schemes were compared with a more regular and for a more complex fracture network in Section 6. Regarding the space discretization, all higher-order schemes can cause spurious oscillations. The lowest-order scheme DG(0) did not generate spurious oscillations, however, showed large numerical dissipation compared to the other schemes. This leads to the conclusion that if spurious oscillations are not acceptable, the lowest-order scheme would have to be used with a fine spatial resolution. For higher-order schemes, flux or slope limiters [63, 64] would be needed. The results of this study illustrate the need for such methods if higher-order schemes are used for transport models in fractured media. The DG schemes cannot be rated against the continuous SUPG schemes. It might be easier to handle flux or slope limiters with DG schemes. However, the DG schemes were computationally more costly than the SUPG schemes.

In terms of the time discretization, we conclude that the TDG(1) scheme had clear advantages for modeling advective transport in fractured media compared to the other methods. Higher-order TDG schemes can effectively reduce numerical dissipation and generate monotonic solutions while maintaining reasonable computational costs when employing the decoupling strategy and the fixed-point iteration method. Moreover, the TDG method allows having flexible space semi-discretization schemes with any flux correction techniques [45, 64].

Appendix A Sequential solution of TDG

Saying we decompose the temporal domain T into $\mathcal{T}_h$, which contains 3 subdomains: I_1, I_2 and I_3. On each subdomain we apply TDG(1), thus we have in total 6 temporal nodes. The block linear system Eq. 27 without jump terms then reads:

$$\begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{L}_{12} & & & & \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} & & & & \\ & & \boldsymbol{L}_{33} & \boldsymbol{L}_{34} & & \\ & & \boldsymbol{L}_{43} & \boldsymbol{L}_{44} & & \\ & & & & \boldsymbol{L}_{55} & \boldsymbol{L}_{56} \\ & & & & \boldsymbol{L}_{65} & \boldsymbol{L}_{66} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^1 \\ \boldsymbol{\xi}^2 \\ \boldsymbol{\xi}^3 \\ \boldsymbol{\xi}^4 \\ \boldsymbol{\xi}^5 \\ \boldsymbol{\xi}^6 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \end{bmatrix}. \tag{A1}$$

After adding jump terms, Eq. A1 becomes:

$$\begin{bmatrix} \boldsymbol{L}_{11}+\boldsymbol{M} & \boldsymbol{L}_{12} & & & & \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} & & & & \\ & -\boldsymbol{M} & \boldsymbol{L}_{33}+\boldsymbol{M} & \boldsymbol{L}_{34} & & \\ & & \boldsymbol{L}_{43} & \boldsymbol{L}_{44} & & \\ & & & -\boldsymbol{M} & \boldsymbol{L}_{55}+\boldsymbol{M} & \boldsymbol{L}_{56} \\ & & & & \boldsymbol{L}_{65} & \boldsymbol{L}_{66} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^1 \\ \boldsymbol{\xi}^2 \\ \boldsymbol{\xi}^3 \\ \boldsymbol{\xi}^4 \\ \boldsymbol{\xi}^5 \\ \boldsymbol{\xi}^6 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1+\boldsymbol{M}\boldsymbol{\xi}^0 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \\ \boldsymbol{b}_5 \\ \boldsymbol{b}_6 \end{bmatrix}. \tag{A2}$$

Apparently, $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ can directly be obtained by solving

$$\begin{bmatrix} \boldsymbol{L}_{11}+\boldsymbol{M} & \boldsymbol{L}_{12} \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^1 \\ \boldsymbol{\xi}^2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1+\boldsymbol{M}\boldsymbol{\xi}^0 \\ \boldsymbol{b}_2 \end{bmatrix}. \tag{A3}$$

After having $\boldsymbol{\xi}^2$, $\boldsymbol{\xi}^3$ and $\boldsymbol{\xi}^4$ can be obtained by solving

$$\begin{bmatrix} \boldsymbol{L}_{33}+\boldsymbol{M} & \boldsymbol{L}_{34} \\ \boldsymbol{L}_{43} & \boldsymbol{L}_{44} \end{bmatrix} \begin{bmatrix} \boldsymbol{\xi}^3 \\ \boldsymbol{\xi}^4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_3+\boldsymbol{M}\boldsymbol{\xi}^2 \\ \boldsymbol{b}_4 \end{bmatrix}. \tag{A4}$$

Similarly, we can solve $\boldsymbol{\xi}^5$ and $\boldsymbol{\xi}^6$ with $\boldsymbol{\xi}^4$.

Therefore, for an arbitrarily large number of time subintervals, we can always solve the block linear system sequentially in time. This property is quite favorable in practice for avoiding solving a huge block linear system, namely decreasing computation times.

Appendix B Solve block linear system

Using the sequential solution strategy (Appendix A), we finally need to solve a block linear system (e.g. Eqs. A3 or A4). The matrix on the left-hand side is a block sparse matrix (see Fig. 13). It is hence hard to find a good preconditioner matrix if one wants to use an iterative solver.

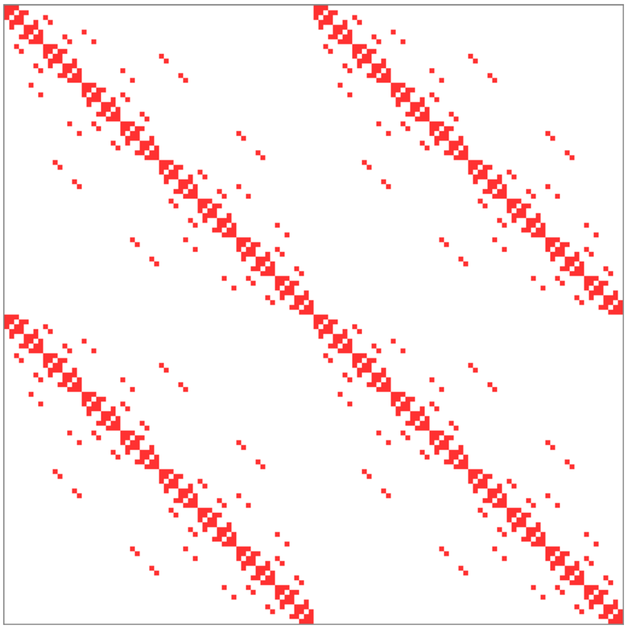


Fig. 13 Exemplary sparsity pattern for a block sparse matrix by TDG(1)

A fixed-point iteration method (see e.g. [65]) can be used to solve the block linear system in a decoupled way. The main two advantages are

1. the dimension of the original block linear system is decreased;
2. for every sublinear system the matrix on the left-hand side is a standard FE sparse matrix which is diagonally dense and non-diagonally sparse (see Fig. 14).

Thus the computation times can be greatly decreased.

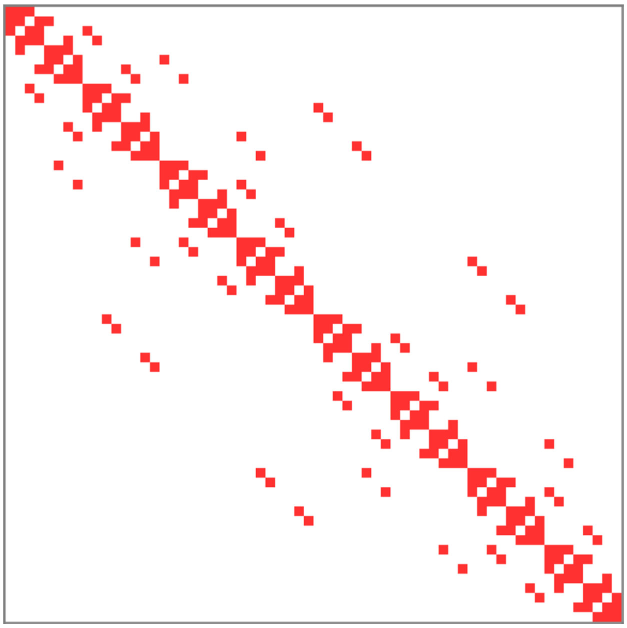


Fig. 14 Exemplary sparsity pattern for a standard sparse matrix in FEM

We use the block linear system (A3) as the instance. First reformulate (A3) with the two sublinear systems:

$$\boldsymbol{L}_{22}\boldsymbol{\xi}^2 = \boldsymbol{b}_2 - \boldsymbol{L}_{21}\boldsymbol{\xi}^1 \tag{A5}$$

$$\boldsymbol{L}_{11}\boldsymbol{\xi}^1 = \boldsymbol{b}_1 + \boldsymbol{M}\boldsymbol{\xi}^0 - \boldsymbol{L}_{12}\boldsymbol{\xi}^2. \tag{A6}$$

Algorithm 3 describes the process of how to solve the two sublinear systems iteratively. We use the direct solver (UMFPACK) to solve the systems (A5) and (A6).

Algorithm 3 Iteratively solve Eqs. A5 and A6.

```
Require: $\xi^0$
Set the relative tolerance $\varepsilon_{tol} = 10^{-3}$
Initialize the error $\varepsilon$ = positive infinity
Initial guess $\boldsymbol{\xi}^1 = \boldsymbol{\xi}^0$
while $\varepsilon \geq \varepsilon_{tol}$ do
    Create a vector $\boldsymbol{\xi}^1_{old}$ saving the last result
    $\boldsymbol{\xi}^1_{old} = \boldsymbol{\xi}^1$
    Solve Eq. A5 and get $\boldsymbol{\xi}^2$
    Solve Eq. A6 and get $\boldsymbol{\xi}^1$
    Calculate the error $\varepsilon = \frac{\left\|\xi^1 - \xi^1_{old}\right\|_{L_2}}{\left\|\xi^1_{old}\right\|_{L_2}}$
end while
Save $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$
```

Similarly, we choose $\boldsymbol{\xi}^2$ as the initial guess of $\boldsymbol{\xi}^3$, when Eq. A4 is considered. Our experience is that the chosen initial guess obtained from the last time step is very close to the unknown vector considered in the current time step at the jump point. Because the amplitude of the jump term should not be very large. Hence, we can converge to the desired result in a few steps with a proper relative tolerance.

We remark that this method is not quite feasible for the case where the degree of polynomials in time is greater than 2. Nevertheless, since TDG(0) and TDG(1) are most widely used in practice, this method still has considerable value in application.

Acknowledgements The authors gratefully acknowledge the financial support from the doctoral program "International Research Training Group (IRTG)" 2657 funded by the German Research Foundation (DFG).

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability The data generated or analyzed during this work are available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to this work.

References

1. Dietrich, P., Helmig, R., Sauter, M., Hötzl, H., Köngeter, J., Teutsch, G.: Flow and Transport in Fractured Porous Media. Springer, Berlin (2005)
2. Berkowitz, B.: Characterizing flow and transport in fractured geological media: a review. Adv. Water Resour. **25**(8–12), 861–884 (2002)
3. Becker, M.W., Shapiro, A.M.: Tracer transport in fractured crystalline rock: evidence of nondiffusive breakthrough tailing. Water Resour. Res. **36**(7), 1677–1686 (2000)
4. Kosakowski, G., Berkowitz, B., Scher, H.: Analysis of field observations of tracer transport in a fractured till. J. Contam. Hydrol. **47**(1), 29–51 (2001)
5. Geiger, S., Emmanuel, S.: Non-fourier thermal transport in fractured geological media. Water Res. Res. **46**(7) (2010)
6. Polubarinova-Koch, P.I.: Theory of ground water movement. In: Theory of ground water movement. Princeton University Press, Princeton (2015)
7. Berre, I., Doster, F., Keilegavlen, E.: Flow in fractured porous media: a review of conceptual models and discretization approaches. Transp. Porous Media **130**(1), 215–236 (2019)
8. Jackson, C.P., Hoch, A.R., Todman, S.: Self-consistency of a heterogeneous continuum porous medium representation of a fractured medium. Water Resour. Res. **36**(1), 189–202 (2000)

9. Eikemo, B., Lie, K.-A., Eigestad, G.T., Dahle, H.K.: Discontinuous Galerkin methods for advective transport in single-continuum models of fractured media. Adv. Water Resour. **32**, 493–506 (2009)
10. Sweeney, M.R., Gable, C.W., Karra, S., Stauffer, P.H., Pawar, R.J., Hyman, J.D.: Upscaled discrete fracture matrix model (UDFM): an octree-refined continuum representation of fractured porous media. Comput. Geosci. **24**(1), 293–310 (2020)
11. Colella, P., Dorr, M., Hittinger, J., Martin, D., McCorquodale, P.: High-order finite-volume adaptive methods on locally rectangular grids. In: Journal of Physics: Conference Series, vol. 180. p. 012010. IOP Publishing (2009)
12. Babuška, I.: The p and hp versions of the finite element method: the state of the art. Finite Elem. 199–239 (1988)
13. Brooks, A.N., Hughes, T.J.: Streamline upwind/petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. Comput. Methods Appl. Mech. Eng. **32**(1–3), 199–259 (1982)
14. Hughes, T.J., Franca, L.P., Hulbert, G.M.: A new finite element formulation for computational fluid dynamics: Viii. the Galerkin/least-squares method for advective-diffusive equations. Comput. Methods Appl. Mech. Eng. **73**(2), 173–189 (1989)
15. Oñate, E., Miquel, J., Zárate, F.: Stabilized solution of the multidimensional advection-diffusion-absorption equation using linear finite elements. Comput. Fluids **36**(1), 92–112 (2007)
16. Kuzmin, D., Löhner, R., Turek, S.: Flux-corrected Transport: Principles, Algorithms, and Applications. Springer, Berlin (2012)
17. Falk, R.S.: Analysis of finite element methods for linear hyperbolic problems. In: Discontinuous Galerkin Methods, pp. 103–112. Springer, Berlin (2000)
18. Burman, E., Smith, G.: Analysis of the space semi-discretized SUPG method for transient convection-diffusion equations. Math. Models Methods Appl. Sci. **21**(10), 2049–2068 (2011)
19. Bochev, P.B., Gunzburger, M.D., Shadid, J.N.: Stability of the SUPG finite element method for transient advection-diffusion problems. Comput. Methods Appl. Mech. Eng. **193**(23–26), 2301–2323 (2004)
20. Giere, S., Iliescu, T., John, V., Wells, D.: SUPG reduced order models for convection-dominated convection-diffusion-reaction equations. Comput. Methods Appl. Mech. Eng. **289**, 454–474 (2015)
21. Vellando, P., Puertas, J., Colominas, I.: SUPG stabilized finite element resolution of the navier-stokes equations: applications to water treatment engineering. Comput. Methods Appl. Mech. Eng. **191**(51–52), 5899–5922 (2002)
22. Kirk, B.S., Carey, G.F.: Development and validation of a SUPG finite element scheme for the compressible navier-stokes equations using a modified inviscid flux discretization. Int. J. Numer. Meth. Fluids **57**(3), 265–293 (2008)
23. Pérez Carreras, A., Carol, I., Prat Catalán, P.: Heat transport with advection in fractured rock. In: COMPLAS 2019: XV international conference on computational plasticity: fundamentals and applications. pp. 565–576 (2019)
24. Rivière, B.: Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. SIAM, Philadelphia (2008)
25. Di Pietro, D.A., Ern, A.: Mathematical Aspects of Discontinuous Galerkin Methods. Mathématiques et Applications. Springer, Berlin (2012). https://books.google.de/books?id=ak-qQvWGA5oC
26. Szabo, B., Mehta, A.: p-convergent finite element approximations in fracture mechanics. Int. J. Numer. Meth. Eng. **12**(3), 551–560 (1978)
27. Cockburn, B., Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J. Numer. Anal. **35**(6), 2440–2463 (1998)
28. Nguyen, N.C., Peraire, J.: Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. J. Comput. Phys. **231**(18), 5955–5988 (2012)
29. Younes, A., Koohbor, B., Fahs, M., Hoteit, H.: An efficient discontinuous Galerkin-mixed finite element model for variable density flow in fractured porous media. J. Comput. Phys. **477**, 111937 (2023)
30. Guo, H., Feng, W., Xu, Z., Yang, Y.: Conservative numerical methods for the reinterpreted discrete fracture model on non-conforming meshes and their applications in contaminant transportation in fractured porous media. Adv. Water Resour. **153**, 103951 (2021)
31. Eriksson, K., Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Computational Differential Equations. Cambridge University Press, Cambridge (1996)
32. Oden, J.T.: A general theory of finite elements. ii. applications. Int. J. Numeric. Methods Eng. **1**(3), 247–259 (1969)
33. Fried, I.: Finite-element analysis of time-dependent phenomena. AIAA J. **7**(6), 1170–1173 (1969)
34. Jamet, P.: Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain. SIAM J. Numer. Anal. **15**(5), 912–928 (1978)
35. Hughes, T.J., Hulbert, G.M.: Space-time finite element methods for elastodynamics: formulations and error estimates. Comput. Methods Appl. Mech. Eng. **66**(3), 339–363 (1988)
36. Langer, U., Steinbach, O.: Space-Time Methods: Applications to Partial Differential Equations, vol. 25. Walter de Gruyter GmbH & Co KG, Berlin (2019)
37. Wick, T.: Space-time Methods: Formulations, Discretization, Solution, Goal-oriented Error Control and Adaptivity. Springer, Compact Textbooks in Mathematics, Berlin (2023). http://www.thomaswick.org/links/Wi23_st_book_preprint_Aug_8_2023.pdf
38. Köcher, U., Bruchhäuser, M.P., Bause, M.: Efficient and scalable data structures and algorithms for goal-oriented adaptivity of space-time fem codes. SoftwareX. **10**, 100239 (2019). https://doi.org/10.1016/j.softx.2019.100239
39. Roth, J., Thiele, J.P., Köcher, U., Wick, T.: Tensor-Product Space-Time Goal-Oriented Error Control and Adaptivity with Partition-of-Unity Dual-Weighted Residuals for Nonstationary Flow Problems. Computational Methods in Applied Mathematics. (2023). https://doi.org/10.1515/cmam-2022-0200
40. Sapotnick, A., Nackenhorst, U.: A combined FIC-TDG finite element approach for the numerical solution of coupled advection-diffusion-reaction equations with application to a bioregulatory model for bone fracture healing. Int. J. Numer. Meth. Eng. **92**(3), 301–317 (2012)
41. Feng, D., Neuweiler, I., Nackenhorst, U.: A spatially stabilized TDG based finite element framework for modeling biofilm growth with a multi-dimensional multi-species continuum biofilm model. Comput. Mech. **59**(6), 1049–1070 (2017)
42. Gander, M.J., Neumuller, M.: Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM J. Sci. Comput. **38**(4), 2173–2208 (2016)
43. Chen, Z., Steeb, H., Diebels, S.: A time-discontinuous Galerkin method for the dynamical analysis of porous media. Int. J. Numer. Anal. Meth. Geomech. **30**(11), 1113–1134 (2006)
44. Schmich, M., Vexler, B.: Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. SIAM J. Sci. Comput. **30**(1), 369–393 (2008)
45. Feng, D., Neuweiler, I., Nackenhorst, U., Wick, T.: A time-space flux-corrected transport finite element formulation for solving multi-dimensional advection-diffusion-reaction equations. J. Comput. Phys. **396**, 31–53 (2019). https://doi.org/10.1016/j.jcp.2019.06.053
46. Sudirham, J., Vegt, J.J., Damme, R.M.: Space-time discontinuous Galerkin method for advection-diffusion problems on time-

dependent domains. Appl. Numer. Math. **56**(12), 1491–1518 (2006)

47. Sudirham, J., Damme, R.M., Vegt, J.J.: Space-time discontinuous Galerkin method for wet-chemical etching of microstructures. In: Proceedings of European Congress in Applied Sciences and Engineering (ECCOMAS). Jyvaskyla, Finland (2004)
48. Chen, Z., Steeb, H., Diebels, S.: A space-time discontinuous Galerkin method applied to single-phase flow in porous media. Comput. Geosci. **12**(4), 525–539 (2008)
49. Bergamaschi, L., Mantica, S., Manzini, G.: A mixed finite element-finite volume formulation of the black-oil model. SIAM J. Sci. Comput. **20**(3), 970–997 (1998)
50. Durlofsky, L.J.: Accuracy of mixed and control volume finite element approximations to darcy velocity and related quantities. Water Resour. Res. **30**(4), 965–973 (1994)
51. Chavent, G., Cohen, G., Jaffre, J.: Discontinuous upwinding and mixed finite elements for two-phase flows in reservoir simulation. Comput. Methods Appl. Mech. Eng. **47**(1–2), 93–118 (1984)
52. Mikelić, A., Wheeler, M.F., Wick, T.: A phase-field method for propagating fluid-filled fractures coupled to a surrounding porous medium. SIAM Multiscale Model. Simul. **13**(1), 367–398 (2015)
53. Amaziane, B., Hontans, T., Koebbe, J.: Equivalent permeability and simulation of two-phase flow in heterogeneous porous media. Comput. Geosci. **5**, 279–300 (2001)
54. Ainsworth, M.: A posteriori error estimation for discontinuous Galerkin finite element approximation. SIAM J. Numer. Anal. **45**(4), 1777–1798 (2007)
55. Codina, R.: Comparison of some finite element methods for solving the diffusion-convection-reaction equation. Comput. Methods Appl. Mech. Eng. **156**(1–4), 185–210 (1998)
56. Arndt, D., Bangerth, W., Feder, M., Fehling, M., Gassmöller, R., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.-P., Sticko, S., Turcksin, B., Wells, D.: The `deal.II` library, version 9.4. J. Numeric. Mathematics. **30**(3), 231–246 (2022) https://doi.org/10.1515/jnma-2022-0054
57. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. J. Comput. Phys. **228**(9), 3232–3254 (2009)
58. Delfour, M., Hager, W., Trochu, F.: Discontinuous Galerkin methods for ordinary differential equations. Math. Comp. **36**, 455–473 (1981)
59. Geiger, S., Dentz, M., Neuweiler, I.: A novel multirate dual-porosity model for improved simulation of fractured and multi-porosity reservoirs. SPE J. **18**(04), 670–684 (2013)
60. Flemisch, B., Berre, I., Boon, W., Fumagalli, A., Schwenck, N., Scotti, A., Stefansson, I., Tatomir, A.: Benchmarks for single-phase flow in fractured porous media. Adv. Water Resour. **111**, 239–258 (2018)
61. Carrera, J., Sanchez-Vila, X., Benet, I., Medina, A., Galarza, G., Guimera, J.: On matrix diffusion: formulations, solution methods and qualitative effects. Hydrogeol. J. **6**, 178–190 (1998). https://doi.org/10.1007/s100400050143
62. Harten, A.: High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**(3), 357–393 (1983)
63. Kuzmin, D.: On the design of general-purpose flux limiters for finite element schemes. i. scalar convection. J. Comput. Phys. **219**(2), 513–531 (2006)
64. Kuzmin, D.: A new perspective on flux and slope limiting in discontinuous Galerkin methods for hyperbolic conservation laws. Computer Methods in Applied Mechanics and Engineering. **373**, 113569 (2021)
65. Richter, T., Wick, T.: Einführung in die Numerische Mathematik - Begriffe. Konzepte und Zahlreiche Anwendungsbeispiele. Springer, Berlin (2017)