`DarcyLite` Modules for a Property-preserving Transport Solver *

Jiangguo Liu, Boyang Yu, and Yingli Li

[1] Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA, liu@math.colostate.edu
[2] School of Mathematics, Jilin University, Changchun, Jilin 130012, China, yuby21@mails.jlu.edu.cn
[3] Department of Mathematics, Colorado State University, Fort Collins, CO 80523-1874, USA, yingli.li@colostate.edu

Abstract. For transport modelled by the time-dependent convection-diffusion equation, positivity of numerical concentrations and mass conservation are two important properties numerical solvers should respect. This paper investigates such a solver based on the implicit Euler time-marching and finite volume discretization on quadrilateral meshes. The solver uses mapped Q_1 bilinear polynomials for approximation of the concentration. A new upwinding technique is adopted to handle convection dominance. Flux correction is devised to ensure nonnegative numerical concentrations. `Matlab` code modules based on efficient implementation of this solver are incorporated into our package `DarcyLite`. Numerical experiments are presented to illustrate the performance of the new solver.

Keywords: Convection-diffusion · Finite volumes · Flux correction · Mass conservation · Positivity-preserving · Quadrilateral meshes

1 Introduction

In this paper, we focus on implementation of a property-preserving finite volume solver for the time-dependent convection-diffusion equation with boundary and initial conditions prototyped as

$$\begin{cases} \partial_t c + \nabla \cdot (\mathbf{v}c - \mathbf{D}\nabla c) = s(\mathbf{x},t), & (\mathbf{x},t) \in \Omega \times (0,T] \\ c(\mathbf{x},t) = c_D(\mathbf{x},t), & (\mathbf{x},t) \in \Gamma_D \times (0,T], \\ (\mathbf{v}c - \mathbf{D}\nabla c) \cdot \mathbf{n} = f_N(\mathbf{x},t), & (\mathbf{x},t) \in \Gamma_N \times (0,T], \\ c(\mathbf{x},0) = c_0(\mathbf{x}), & \mathbf{x} \in \Omega, \end{cases} \tag{1}$$

where Ω is a polygonal domain, $T > 0$ the final time for numerical simulations, $c(\mathbf{x},t)$ the unknown concentration of the substance being transported,

* Y. Li and J. Liu were partially supported by US National Science Foundation under grant DMS-2208590.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

$\mathbf{v} \in H(\mathrm{div}, \Omega)$ a known velocity, $\mathbf{D}$ a symmetric uniformly positive-definite matrix for describing diffusion/dispersion, $s(\mathbf{x}, t)$ a known source. Here Γ_D, Γ_N are respectively the Dirichlet and Neumann boundaries which do not overlap but $\Gamma_D \cup \Gamma_N = \partial\Omega$, c_D, f_N are respectively the Dirichlet and Neumann data (concentration and flux), and $c_0(\mathbf{x})$ is the initial condition.

Transport problems modelled by the time-dependent convection-diffusion equation are important for simulations of many real-world problems, e.g., groundwater contamination [10], drug release [16], controlled drug delivery to vitreous humors [14], and flow and transport in fractured media [1], just to name a few.

For the transport and similar problems, numerical methods need to respect certain important physical properties, e.g., positivity (nonnegativity) of solution. There have been some successful efforts in this regard. Bound-preserving discontinuous Galerkin method for compressible miscible displacement in porous media were investigated in [3,2]. An efficient bound-preserving and energy stable algorithm for compressible gas flow in porous media was recently developed in [6]. Finite volume methods with positivity-preserving have also been developed in [7,8] [9]. Some techniques have also been extended to time-fractional convection-diffusion problems [21].

In this paper, we first investigate a finite volume solver on a quadrilateral mesh for the time-dependent convection-diffusion equation with focus on its efficient implementation. Then this transport solver is coupled with a weak Galerkin finite element solver for Darcy flow on a quadrilateral mesh. The numerical velocity obtained from the latter is fed into the former. The coupling provides an integrated solver for transport in porous media. Note that spatial discretizations on quadrilateral meshes are adopted, as done in [4,5,13,18], since quadrilateral meshes can accommodate complicated geometry flexibly and may involve less computation than that on triangular meshes. `Matlab` modules for these solvers will be incorporated into our code package `DarcyLite`. These new efforts extend our earlier work in [11,19,20].

The rest of this paper is organized as follows. Section 2 presents finite volume discretization on quadrilaterals using mapped bilinear approximants, upwinding treatment, and a positivity-correction technique for both convective and diffusive fluxes. Section 3 combines these spatial discretization techniques with the implicit Euler temporal discretization to develop a transport solver that is mass-conservative and guarantees non-negativity of the numerical concentration. Section 4 presents coupling of this transport solver with a Darcy solver. Section 5 discusses procedures and strategies for efficient implementation in `Matlab`. Section 6 presents numerical results. Section 7 concludes the paper with remarks on further work.

2 FV $\mathcal{Q}_1$-functions, Upwinding, and Flux Correction

For 2-dim problems, quadrilateral meshes are equally flexible as triangular meshes in accommodation of complicated geometry. They may be in better agreement

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

with physical features and involve less unknowns. For now, we consider shape-regular quadrilateral meshes that do not contain hanging nodes.

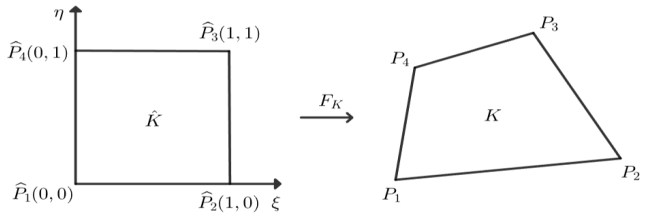

Fig. 1. A bilinear mapping from the reference square $[0,1]^2$ to a general quadrilateral.

A convex **quadrilateral** K is viewed as a bilinear mapping image of the reference element $\hat{K} = [0,1]^2$ (we may use $\hat{E}$ also), namely, the unit square. Assume the four vertices $P_i(x_i, y_i), 1 \le i \le 4$ are oriented counterclockwise. The bilinear mapping and its Jacobian matrix are expressed as

$$\begin{cases} x = a_1 + a_2\xi + a_3\eta + a_4\xi\eta, \\ y = b_1 + b_2\xi + b_3\eta + b_4\xi\eta, \end{cases} \qquad \mathbf{J}(\xi,\eta) = \begin{bmatrix} a_2 + a_4\eta & a_3 + a_4\xi \\ b_2 + b_4\eta & b_3 + b_4\xi \end{bmatrix}, \tag{2}$$

where (ξ, η) are the reference coordinates in $\hat{K}$. The coefficients are determined as

$$\begin{cases} a_1 = x_1 \\ b_1 = y_1 \end{cases} \begin{cases} a_2 = x_2 - x_1 \\ b_2 = y_2 - y_1 \end{cases} \begin{cases} a_3 = x_4 - x_1 \\ b_3 = y_4 - y_1 \end{cases} \begin{cases} a_4 = (x_1 + x_3) - (x_2 + x_4) \\ b_4 = (y_1 + y_3) - (y_2 + y_4) \end{cases} \tag{3}$$

For finite volume discretization, we need 4 normal vectors on the 4 edges. They are not outward unit vectors. Instead, their directions match the $\xi-, \eta-$ axes.

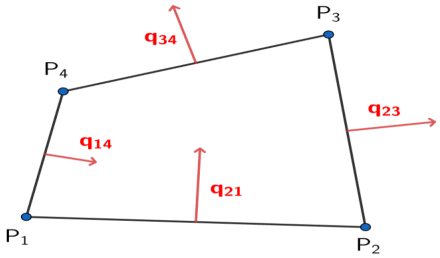

Fig. 2. Normal vectors on the edges of a quadrilateral.

It is convenient to consider convex combinations of these normal vectors.

$$\mathbf{q}_1(\xi) = (1-\xi)\,\mathbf{q}_{14} + \xi\,\mathbf{q}_{23}, \qquad \mathbf{q}_2(\eta) = (1-\eta)\,\mathbf{q}_{21} + \eta\,\mathbf{q}_{34}. \tag{4}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

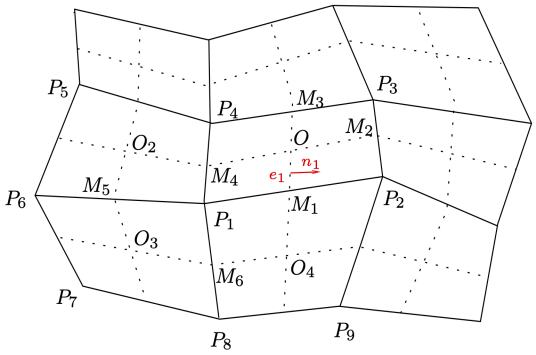

Fig. 3. Primal nodes and sub-volumes.

Let $\hat{w}(\xi,\eta)$ be a bilinear function defined on $\hat{E} = [0,1]^2$ and $w(x,y)$ be the shape function after being mapped to E. Specifically, let the four nodal values be $w_{P_1}, w_{P_2}, w_{P_3}, w_{P_4}$. It can be verified that

$$\begin{aligned} J(\xi,\eta)(\nabla w)|_E = &(w_{P_2} - w_{P_1})(1-\eta)\,\mathbf{q}_1(\xi) + (w_{P_3} - w_{P_4})\eta\,\mathbf{q}_1(\xi) \\ &+ (w_{P_4} - w_{P_1})(1-\xi)\,\mathbf{q}_2(\eta) + (w_{P_3} - w_{P_2})\xi\,\mathbf{q}_2(\eta). \end{aligned} \tag{5}$$

For finite volume discretizations based on the mapped bilinear shape functions on a quadrilateral mesh, each quadrilateral/volume in the primal mesh is divided into 4 sub-volumes by connecting the volume center with the 4 edge midpoints, as shown in Fig.3. The sub-volumes surrounding a node, say P_1, form a dual volume. Applying the Gauss Divergence Theorem, an integral of a physical quantity on the dual volume is converted to a line integral on those dual edges, for instance, e_1 that connects M_1 (edge point) and O (element center). Clearly, $\xi = \frac{1}{2}$ and $\eta \in [0, \frac{1}{2}]$ on e_1. Let C_h be the numerical concentration, then

$$\begin{aligned} &J\Big(\frac{1}{2},\eta\Big)\,(\nabla C_h)|_{e_1} \\ &= \Big(C_h(P_2) - C_h(P_1)\Big)(1-\eta)\,\mathbf{q}_1\big(\frac{1}{2}\big) + \Big(C_h(P_3) - C_h(P_4)\Big)\eta\,\mathbf{q}_1\big(\frac{1}{2}\big) \\ &+ \Big(C_h(P_4) - C_h(P_1)\Big)\frac{1}{2}\,\mathbf{q}_2(\eta) + \Big(C_h(P_3) - C_h(P_2)\Big)\frac{1}{2}\,\mathbf{q}_2(\eta). \end{aligned} \tag{6}$$

2.1 A New Upwinding Technique

As shown in Fig. 3, one primal quadrilateral element is divided into 4 sub-volumes, whereas several (usually 4 for a logically rectangular quadrilateral mesh) sub-volumes surrounding a primal node form **a dual volume**. Here we see $M_1OM_4O_2M_5O_3M_6O_4M_1$ form the dual volume for P_1. For convenience, we denote it as $E^*_{P_1}$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

On the other hand, for a point (x, y) on a primal edge shared by two primal elements E_1, E_2, we consider ***an averaged gradient***

$$\overline{\nabla} C_h(x,y) = \frac{1}{2}\Big(\nabla(C_h|_{E_1})(x,y) + \nabla(C_h|_{E_2})(x,y)\Big). \tag{7}$$

For any point on e_1, the reference coordinates are $\left(\frac{1}{2}, \eta\right)$ with $\eta \in [0, \frac{1}{2}]$. Its ***upstream point*** is defined as

$$(x^*(\eta), y^*(\eta)) = \begin{cases} F_K(0,\eta) & \text{if } \int_{e_1} \mathbf{v}\cdot\mathbf{n}_1 \geq 0, \\ F_K(1,\eta) & \text{if } \int_{e_1} \mathbf{v}\cdot\mathbf{n}_1 < 0. \end{cases}$$

This point is either on the line segment P_1M_4 or on the line segment P_2M_2. Accordingly, for a point on e_1, we define is ***the upwind approximation of the concentration*** as (viewed as a Taylor expansion)

$$c(x,y) \approx C_h^{\text{up}} = C_h((x^*(\eta), y^*(\eta))) + \mathbf{h}\cdot\overline{\nabla} C_h(x^*(\eta), y^*(\eta)), \tag{8}$$

where $\mathbf{h} = [x - x^*(\eta), y - y^*(\eta)]$ is the position vector.

A new upwinding technique. Now we modify the discrete bilinear form for convection as

$$\mathcal{B}_h(C_h, \psi_h) = \sum_{E_P^* \in \mathcal{E}_h^*} \int_{\partial E_P^*} (\mathbf{v}\cdot\mathbf{n}) C_h^{\text{up}}\, \psi_h, \quad \forall C_h \in \mathcal{U}_h, \quad \forall \psi_h \in \mathcal{W}_h, \tag{9}$$

where $\mathcal{U}_h$ is the trial function space defined in (26) and $\mathcal{W}_h$ is the test function space defined in (27).

2.2 Splitting of the Upwind Convective Flux

Consider the dual edge e_1 in Fig. 3, shared by two sub-volumes (also two dual volumes), which surrounds two primal nodes P_1, P_2, respectively.

Now consider the convective fluxes across the dual edge e_1:

$$\mathcal{F}^c_{P_1,e_1} = \int_{e_1} (\mathbf{v}\cdot\mathbf{n}_1) C_h^{\text{up}} ds, \qquad \mathcal{F}^c_{P_2,e_1} = -\int_{e_1} (\mathbf{v}\cdot\mathbf{n}_1) C_h^{\text{up}} ds. \tag{10}$$

Assume $\displaystyle\int_{e_1} (\mathbf{v}\cdot\mathbf{n}_1) ds \geq 0$. Utilizing the upwinding info, we obtain

$$\begin{aligned}
\mathcal{F}^c_{P_1,e_1} &= \int_0^{\frac{1}{2}} \hat{\mathbf{v}}\Big(\frac{1}{2},\eta\Big)\cdot\mathbf{q_1}\Big(\frac{1}{2}\Big)\Big(C_h(\hat{x}(\eta),\hat{y}(\eta)) + \hat{\mathbf{h}}(\eta)\cdot\overline{\nabla} C_h(\hat{x}(\eta),\hat{y}(\eta))\Big)\, d\eta \\
&= \int_0^{\frac{1}{2}} \hat{\mathbf{v}}\Big(\frac{1}{2},\eta\Big)\cdot\mathbf{q_1}\Big(\frac{1}{2}\Big)\Big((1-\eta)C_h^{P_1} + \eta C_h^{P_4} + \hat{\mathbf{h}}(\eta)\cdot\overline{\nabla} C_h(\hat{x}(\eta),\hat{y}(\eta))\Big)\, d\eta \\
&= \int_0^{\frac{1}{2}} \hat{\mathbf{v}}\Big(\frac{1}{2},\eta\Big)\cdot\mathbf{q_1}\Big(\frac{1}{2}\Big) d\eta\; C_h^{P_1} - 0\times C_h^{P_2} \\
&+ \int_0^{\frac{1}{2}} \hat{\mathbf{v}}\Big(\frac{1}{2},\eta\Big)\cdot\mathbf{q_1}\Big(\frac{1}{2}\Big)\Big(\eta(C_h^{P_4} - C_h^{P_1}) + \hat{\mathbf{h}}(\eta)\cdot\overline{\nabla} C_h(\hat{x}(\eta),\hat{y}(\eta))\Big)\, d\eta,
\end{aligned} \tag{11}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

where $\hat{\mathbf{v}}(\xi,\eta)=\mathbf{v}\circ F_K(\xi,\eta)$ and $\hat{\mathbf{h}}(\eta)=F_K(\frac{1}{2},\eta)-F_K(0,\eta)$.

Note P_1, P_2 are respectively the upstream and downstream nodes. We rewrite the convective flux as

$$\mathcal{F}^c_{P_1,e_1}=\kappa_{e_1}C_h^{P_1}-0\times C_h^{P_2}+R^c_{P_1,e_1}, \tag{12}$$

where

$$\kappa_{e_1}=\int_0^{\frac{1}{2}}\hat{\mathbf{v}}\big(\frac{1}{2},\eta\big)\cdot\mathbf{q_1}\big(\frac{1}{2}\big)d\eta\geq 0, \tag{13}$$

$$R^c_{P_1,e_1}=\int_0^{\frac{1}{2}}\hat{\mathbf{v}}\big(\frac{1}{2},\eta\big)\cdot\mathbf{q_1}\big(\frac{1}{2}\big)\left(\eta(C_h^{P_4}-C_h^{P_1})+\hat{\mathbf{h}}(\eta)\cdot\overline{\nabla}C_h(\hat{x}(\eta),\hat{y}(\eta))\right)d\eta. \tag{14}$$

The benefit of the rewriting is two-fold. Firstly, it demonstrates a quasi two-point flux structure. Secondly, the remainder provides hints for positivity correction. Accordingly, we rewrite the reverse flux as

$$\mathcal{F}^c_{P_2,e_1}=0\times u_{P_2}-\kappa_{e_1}u_{P_1}+R^c_{P_2,e_1},\qquad R^c_{P_2,e_1}=-R^c_{P_1,e_1}. \tag{15}$$

2.3 Diffusive Flux and its Splitting

In a similar way, we consider the diffusive fluxes across the dual edge e_1.

$$\mathcal{F}^d_{P_1,e_1}=-\int_{e_1}\mathbf{D}\nabla C_h\cdot\mathbf{n}_1 ds,\qquad \mathcal{F}^d_{P_2,e_1}=-\int_{e_1}\mathbf{D}\nabla C_h\cdot(-\mathbf{n}_1)ds. \tag{16}$$

Obviously, $\mathcal{F}^d_{P_1,e_1}+\mathcal{F}^d_{P_2,e_1}=0$. Going through similar technical details, we obtain a useful splitting (see [21] for more details)

$$\mathcal{F}^d_{P_1,e_1}=\gamma_{e_1}(C_h^{P_1}-C_h^{P_2})+R^d_{P_1,e_1}, \tag{17}$$

where

$$\gamma_{e_1}=\int_0^{\frac{1}{2}}\frac{\mathbf{D}\mathbf{q_1}(\frac{1}{2})\cdot\mathbf{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}(1-\eta)d\eta, \tag{18}$$

and

$$\begin{aligned}R^d_{P_1,e_1}=&\int_0^{\frac{1}{2}}\eta\frac{\mathbf{D}\mathbf{q_1}(\frac{1}{2})\cdot\mathbf{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}d\eta\left(C_h^{P_4}-C_h^{P_3}\right)\\&+\int_0^{\frac{1}{2}}\frac{1}{2}\frac{\mathbf{D}\mathbf{q_2}(\eta)\cdot\mathbf{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}d\eta\left(C_h^{P_1}-C_h^{P_4}\right)\\&+\int_0^{\frac{1}{2}}\frac{1}{2}\frac{\mathbf{D}\mathbf{q_2}(\eta)\cdot\mathbf{q_1}(\frac{1}{2})}{J_K(\frac{1}{2},\eta)}d\eta\left(C_h^{P_2}-C_h^{P_3}\right).\end{aligned} \tag{19}$$

Similarly,

$$\mathcal{F}^d_{P_2,e_1}=\gamma_{e_1}\left(C_h^{P_2}-C_h^{P_1}\right)+R^d_{P_2,e_1},\qquad R^d_{P_2,e_1}=-R^d_{P_1,e_1}. \tag{20}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

2.4 Flux Correction for Positivity of Numerical Concentration

Flux splitting motivates a technique for positivity correction. First, we define

$$\begin{aligned}\mathcal{I}_{P_1,e_1} &= \mathcal{F}_{P_1,e_1} + \mathcal{G}_{P_1,e_1} = \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R^d_{P_1,e_1} + R^c_{P_1,e_1},\\ \mathcal{I}_{P_2,e_1} &= \mathcal{F}_{P_2,e_1} + \mathcal{G}_{P_2,e_1} = \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} + R^d_{P_2,e_1} + R^c_{P_2,e_1}.\end{aligned} \tag{21}$$

Setting $R_{e_1} = R^d_{P_1,e_1} + R^c_{P_1,e_1} = -R^d_{P_2,e_1} - R^c_{P_2,e_1}$, we obtain

$$\begin{aligned}\mathcal{I}_{P_1,e_1} &= \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R_{e_1},\\ \mathcal{I}_{P_2,e_1} &= \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - R_{e_1}.\end{aligned} \tag{22}$$

Utilizing the positive and negative parts of R_{e_1}, we rewrite

$$\begin{aligned}\mathcal{I}_{P_1,e_1} &= \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - \gamma_{e_1} u_{P_2} + R^+_{e_1} - R^-_{e_1},\\ \mathcal{I}_{P_2,e_1} &= \gamma_{e_1} u_{P_2} - \left(\gamma_{e_1} + \kappa_{e_1}\right) u_{P_1} - R^+_{e_1} + R^-_{e_1}.\end{aligned} \tag{23}$$

We take an empirical large constant $B > 0$ to slightly nonlinearize the quantity. For example, considering solute transport across the dual edge e_1 that involves concentration unknowns at node P_1, P_2, We have

$$\begin{aligned}\widetilde{\mathcal{I}}_{P_1,e_1} &= \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR^+_{e_1}}{BC^{P_1}_h + h^2}\right) C^{P_1}_h - \left(\gamma_{e_1} + \frac{BR^-_{e_1}}{BC^{P_2}_h + h^2}\right) C^{P_2}_h,\\ \widetilde{\mathcal{I}}_{P_2,e_1} &= \left(\gamma_{e_1} + \frac{BR^-_{e_1}}{BC^{P_2}_h + h^2}\right) C^{P_2}_h - \left(\gamma_{e_1} + \kappa_{e_1} + \frac{BR^+_{e_1}}{BC^{P_1}_h + h^2}\right) C^{P_1}_h.\end{aligned} \tag{24}$$

3 A Property-preserving FV-$\mathcal{Q}_1$ Transport Solver

Now we return to the time-dependent convection-diffusion problem (24). The implicit Euler is employed for temporal discretization, whereas finite volumes with mapped bilinear shape functions on quadrilaterals are used to approximate the convective and diffusive fluxes. A new upwinding technique is utilized to hand the convection-dominance, and the positivity-correction technique is applied to both convective and diffusive fluxes. All these techniques combined lead to an efficient transport solver that is mass-conservative and ensures non-negativity of numerical concentrations.

First, a temporal partition is established for the time period $[0, T]$:

$$0 = t_0 < t_1 < \cdots < t_{n-1} < t_n < \cdots < t_N = T.$$

When a uniform partition is adopted, we have $\Delta t = T/N, t_n = n\Delta t, 0 \le n \le N$. A straightforward transport solver takes the following form

$$\left(\frac{C^{(n)}_h - C^{(n-1)}_h}{\Delta t}, \psi\right) + \left((\mathbf{v}\cdot\mathbf{n})C^{(n)}_h, \psi\right) + \left((-\mathbf{D}\nabla C^{(n)}_h)\cdot\mathbf{n}, \psi\right) = \left(s(\cdot, t_n), \psi\right). \tag{25}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

This leads to a time-marching algorithm. When finite volume methods are used for spatial discretization, the 2nd and 3rd terms on the RHS are usually converted to line integrals.

Let $\mathcal{E}_h$ be a quadrilateral mesh for $\Omega \subset \mathbb{R}^2$. Mathematically, ***the trial space*** based on the mapped bilinear shape functions is defined as

$$\begin{aligned} \mathcal{U}_h &= \{C_h \in C(\overline{\Omega}) :\ C_h|_E = \hat{C}_h \circ F_E^{-1},\ \hat{C}_h \in Q_1(\widehat{E}),\ \forall E \in \mathcal{E}_h\} \\ &= \mathrm{Span}\{\phi_P :\ P \in \mathcal{P}_h\}, \end{aligned} \tag{26}$$

where ϕ_P represents a typical nodal basis function, and $\{\mathcal{P}_h\}$ is the collection of all nodes in $\mathcal{E}_h$. Accordingly, ***the test space*** is defined on ***the dual mesh*** $\mathcal{E}_h^*$, which consists of the dual volumes surrounding the nodes in the primal mesh

$$\mathcal{W}_h = \{\psi \in L_2(\overline{\Omega}) :\ \psi|_{E_P^*} = \text{const},\ \forall E_P^* \in \mathcal{E}_h^*\}. \tag{27}$$

Transport Solver (FV with upwinding & flux correction). Set $C_h^{(0)}$ as the nodal interpolation of the initial condition. For $1 \le n \le N$, seek $C_h^{(n)} \in \mathcal{U}_h$ so that

$$\begin{aligned} &\left(C_h^{(n)}, \psi_P\right) + \Delta t\big(\widetilde{\mathcal{A}}_h(C_h^{(n)}, \psi_P) + \widetilde{\mathcal{B}}_h(C_h^{(n)}, \psi_P)\big) \\ &= \big(C_h^{(n-1)}, \psi_P\big) + \Delta t\left(s^{(n)}, \psi_P\right), \qquad \forall P \in \mathcal{P}_h, \end{aligned} \tag{28}$$

where the modified bilinear forms $\widetilde{\mathcal{A}}_h, \widetilde{\mathcal{B}}_h$ have already incorporated positivity-correction and upwinding treatment and thus satisfy

$$\widetilde{\mathcal{A}}_h(c_h^n, \psi_P) + \widetilde{\mathcal{B}}_h(c_h^n, \psi_P) = \sum_{e \subset \partial E_P^*} \widetilde{I}_{P,e}. \tag{29}$$

Note that (28) is a typical time-marching scheme for a time-dependant problem. Within each time step, it is actually a slightly nonlinear problem, due to the flux correction in (24). A Picard iterative algorithm is usually employed along with a stopping criterion. More details can be found in [21].

For real-world applications, a transport problem is usually coupled with other problems, e.g., flow in porous media. Then the velocity in the transport problem is obtained from another numerical solver for the Darcy equation. When the Darcy flow equation is solved by the weak Galerkin $(P_1, P_1; AC_1)$ finite element method on the same quadrilateral mesh (as used by the transport solver), we can utilize the numerical velocity $\mathbf{v}_h \in AC_1(\mathcal{E}_h)$ in a nice way. Such a coupling will be accounted in a later section of this paper.

4 Coupling with a Weak Galerkin (WG) Darcy Solver

In this paper, we would like to address transport in porous medium, which is usually modelled mathematically as coupling of the Darcy equation and the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

time-dependant convection-diffusion equation. For simplicity, here we consider a one-way coupling. This means the velocity depicted by the Darcy equation is utilized in the transient transport equation, but the solute transport does not affect the flow in the porous medium. The coupled problem is described as

$$\begin{cases} \nabla \cdot (-\mathbf{K}\nabla p) = \nabla \cdot \mathbf{v} = f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \partial_t c + \nabla \cdot (\mathbf{v}c - \mathbf{D}\nabla c) = s(\mathbf{x}, t), & (\mathbf{x}, t) \in \Omega \times (0, T], \end{cases} \tag{30}$$

where $p(\mathbf{x})$ is the fluid pressure, $\mathbf{K}$ is a 2×2 permeability matrix that is uniformly symmetric positive-definite, $f(\mathbf{x})$ is an external force; whereas $c, \mathbf{D}, s$ in the 2nd equation of (30) bear the same meaning as that in Equation (1).

$$\begin{cases} p|_{\Gamma_D^{\mathcal{D}}} = p_D, \\ (-\mathbf{K}\nabla p)|_{\Gamma_N^{\mathcal{D}}} = u_N, \end{cases} \qquad \begin{cases} c|_{\Gamma_D^{\mathcal{T}} \times (0,T]} = c_D, \\ (\mathbf{v}c - \mathbf{D}\nabla c) \cdot \mathbf{n}|_{\Gamma_N^{\mathcal{T}} \times (0,T]} = f_N, \\ c(\mathbf{x}, 0) = c_0(\mathbf{x}), \quad \mathbf{x} \in \Omega. \end{cases} \tag{31}$$

There are many popular numerical methods for Darcy flow problems: discontinuous Galerkin (DG) methods, enriched Galerkin (EG) methods [15], weak Galerkin (WG) methods [13,12], and the classical mixed finite element methods (MFEMs). For good Darcy solvers, two important properties need to be satisfied.

- Local mass conservation;
- Normal flux continuity.

The latter implies that the numerical velocity from a Darcy solver is in $H(\mathrm{div}, \Omega)$.

The above Darcy flow problem is essentially is an elliptic boundary value problem. It has been proved in [13] that the weak Galerkin $(P_k, P_k; AC_k)$ (with $k \geq 0$) finite element methods for quadrilateral meshes indeed satisfy the aforementioned two properties. The velocity after a local post-processing (here $\mathbf{Q}_h$ is an elementwise L^2-projection into the AC_k space)

$$\mathbf{v}_h = \mathbf{Q}_h(-\mathbf{K}\nabla_w p_h)$$

is actually in the global $\mathcal{AC}$-space, which is a finite-dim subspace of $H(\mathrm{div}, \Omega)$.

To match the 2nd order spatial accuracy by the mapped Q_1 finite volume solver for transport problems, we choose $\mathrm{WG}(P_1, P_1; AC_1)$ solver for Darcy flow. Inside each primal quadrilateral element, the numerical velocity is a linear combination of the following 10 basis functions:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} X \\ 0 \end{bmatrix}, \begin{bmatrix} Y \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ X \end{bmatrix}, \begin{bmatrix} 0 \\ Y \end{bmatrix}, \begin{bmatrix} X^2 \\ XY \end{bmatrix}, \begin{bmatrix} XY \\ Y^2 \end{bmatrix}, \mathcal{P}_E \begin{bmatrix} 1 - \hat{x}^2 \\ 2\hat{x}\hat{y} \end{bmatrix}, \mathcal{P}_E \begin{bmatrix} 2\hat{x}\hat{y} \\ 1 - \hat{y}^2 \end{bmatrix},$$

where $X = x - x_c, Y = y - y_c$ with (x_c, y_c) being the element center, $\mathcal{P}_E$ is the Piola transform, and $(\hat{x}, \hat{y})$ are the coordinates in the reference square $\hat{E} = [0, 1]^2$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

5 Procedures and Strategies for Efficient Implementation

The transport solver investigated in this paper has been implemented in `Matlab` and new code modules have been incorporated into our package `DarcyLite`. These include the following major functions

```
[DualMesh] = function FV_GenDualMesh(QuadriMesh);
TransCD_QuadriFVQ1_AsmGlbMassMat(DualMesh);
TransCD_QuadriFVQ1_AsmConvFluxUpwndPosCorr( ...
  DualMesh, Ch, NumerVelCofAC1, BigB);
TransCD_QuadriFVQ1_AsmDiffFluxPosCorr(DualMesh, Ch, BigB);
TransCD_QuadriFVQ1_AsmSource(DualMesh, t, fxns);
TransCD_QuadriFVQ1_AsmBndryConds(GlbMat, GlbRHS, BndryEdge, EdgeVal);
```

respectively for

(i) Generating the dual mesh based on a known primal quadrilateral mesh;
(ii) Assembling the global mass matrix (using the *lump of mass* technique), See [17,21,22];
(iii) Assembling the global stiffness matrix for the convective flux with upwinding treatment and positivity-correction;
(iv) Assembling the global stiffness matrix for the diffusive flux with positivity-correction;
(v) Assembling the global RHS with contribution from the source term, which is just easy numerical integration on sub-volumes;
(vi) Modifying the algebraic system as the boundary conditions are enforced.

Note the Picard iterations are nested inside the time-marching loop and needs to repeat (iii), (iv), (vi) for each iteration.

5.1 Dual Mesh Data Structure

The dual mesh used by our finite volume method consists of the sub-volumes and all related geometric and topological info, e.g., the volume/element centers, edge midpoints, the normal vectors on the dual edges. We handle the nodes in the primal quadrilateral mesh along with the element centers and edge midpoints altogether. Secondary info includes adjacency of such points and sub-volumes.

The above info will be extensively used for computation of the convective and diffusive fluxes across the dual edges and also assembly of the resulting matrices. The computation is performed on the dual edges, but the assembly is node-wise for the whole (quadrilateral) mesh.

On the other hand, computations of the mass matrices and contributions from the source term are conducted on the sub-volumes, which are treated (regarding numerical integration) in a similar way as the quadrilaterals in the primal mesh.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

5.2 Standard and Mapped Shape Functions on Quadrilaterals

Note that C_h is a linear combination of four mapped $\mathcal{Q}_1$ nodal basis functions $\phi_i(x,y), 1 \le i \le 4$. For each such basis function, $\phi(x,y) = \hat{\phi}(\hat{x},\hat{y})$, where

$$\hat{\phi}_1 = (1-\hat{x})(1-\hat{y}), \quad \hat{\phi}_2 = \hat{x}(1-\hat{y}), \quad \hat{\phi}_3 = \hat{x}\hat{y}, \quad \hat{\phi}_4 = (1-\hat{x})\hat{y}. \tag{32}$$

$$\hat{\nabla}\hat{\phi}_1 = \begin{bmatrix} -(1-\hat{y}) \\ -(1-\hat{x}) \end{bmatrix}, \ \hat{\nabla}\hat{\phi}_2 = \begin{bmatrix} 1-\hat{y} \\ -\hat{x} \end{bmatrix}, \ \hat{\nabla}\hat{\phi}_3 = \begin{bmatrix} \hat{y} \\ \hat{x} \end{bmatrix}, \ \hat{\nabla}\hat{\phi}_4 = \begin{bmatrix} -\hat{y} \\ 1-\hat{x} \end{bmatrix}, \tag{33}$$

$$\nabla\phi = \begin{bmatrix} \partial_x\phi \\ \partial_y\phi \end{bmatrix} = \begin{bmatrix} \partial_x\hat{\phi} \\ \partial_y\hat{\phi} \end{bmatrix} = \begin{bmatrix} \frac{\partial\hat{x}}{\partial x} & \frac{\partial\hat{y}}{\partial x} \\ \frac{\partial\hat{x}}{\partial y} & \frac{\partial\hat{y}}{\partial y} \end{bmatrix} \begin{bmatrix} \partial_{\hat{x}}\hat{\phi} \\ \partial_{\hat{y}}\hat{\phi} \end{bmatrix} = (\mathbf{J}^{-1})^T \hat{\nabla}\hat{\phi}. \tag{34}$$

The above **T**ranspose of the **I**nverse of the **J**acobian matrix (referred as JIT), along with the formulas are very useful for mesh-wise calculations (not through looping over all individual elements), namely, the vectorization feature/technique discussed in [12].

5.3 Positivity-correction for Diffusive and Convective Fluxes

The positive and negative parts of a quantity can be easily implemented as

$$\texttt{Rp} = (\texttt{abs(R)} + \texttt{R})/2; \qquad \texttt{Rn} = (\texttt{abs(R)} - \texttt{R})/2;$$

An empirical large positive parameter `B` is set for the nonlinearization in (24). For Picard iterations, we set a maximal number of iterations allowed (`MaxItr`) and a threshold (ε) for maximal componentwise discrepancy (vector ∞-norm) between two successive approximations.

- This applies to both convective and diffusive fluxes across all four dual edges.
- Treatment for fluxes in the opposite directions is also performed.

5.4 Picard Iterations and Solving Linear Systems

The positivity correction for fluxes involves nonlinear approximation to the nodal concentration, which is used in calculations of both convective and diffusive fluxes. At each time step, a Picard iteration loop is embedded with `MaxItr`, usually 100, specified for the maximal number of iterations, and `epsilon`, usually 10^{-4}, specified for the acceptable discrepancy between two successive approximations. Within the loop, a large-size sparse linear system is solved for the nodal concentration. For now, `Matlab` built-in backslash is used.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

5.5 Coupling with WG($P_1, P_1; AC_1$) Darcy Solver

In this paper, we have discussed also the coupling of the FV $\mathcal{Q}_1$ transport solver with the WG($P_1, P_1; AC_1$) Darcy solver. This is a viable choice. The numerical pressure and concentration are all approximated on the same quadrilateral mesh. The numerical velocity obtained from post-processing will also be used on the same quadrilateral mesh. Specifically, it is in the $AC_1(\mathcal{E}_h)$ space, which has normal continuity across the edges in the primal quadrilateral mesh $\mathcal{E}_h$. As the convective and diffusive fluxes are computed on the dual edges, velocity normal continuity is obvious.

6 Numerical Experiments

This section presents numerical results to demonstrate positivity-preserving of the new transport solver investigated in this paper. A time-dependent convection-diffusion problem is solved by the finite volume $\mathcal{Q}_1$ method that has incorporated a new upwinding technique and flux correction for positivity of numerical concentration. A numerical velocity is fed by the WG($P_1, P_1; AC_1$) solver that solves the Darcy equation with the permeability shown in Fig. 4 left panel.

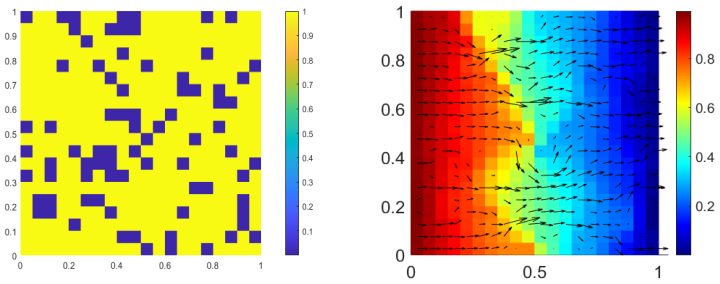

Fig. 4. Ex.1: left: Permeability profile; Right: Numerical pressure and velocity by WG.

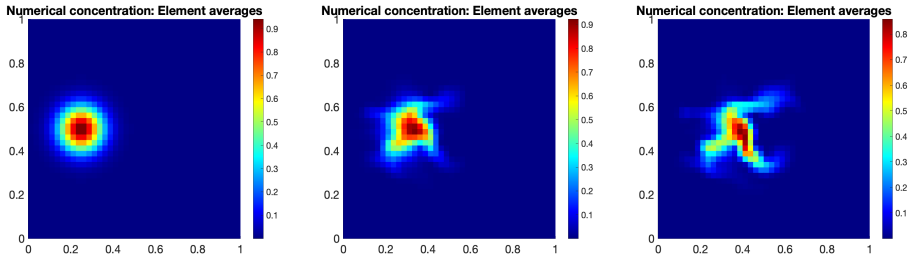

Fig. 5. Ex.1 ($T = 0.5$): Numerical concentrations at $t = 0, \frac{T}{2}, T$; No negative values.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

Example 1 (Positivity). The domain is $\Omega = [0,1]^2$. For fluid flow, the permeability takes value 1 or 10^{-6} on a 20×20 grid, demonstrating heterogeneity. A Dirichlet condition $p = 1$ is posed on the left side $x = 0$, whereas $p = 0$ is posed on $x = 1$. A no-flow condition is posed on the top and bottom sides, as shown in Fig. 4 left panel. A transport problem is posted on the same domain $\Omega = [0,1]^2$ with the final simulation time $T = 0.25$. The initial condition for concentration is set as a Gaussian hump centred as $(x_c, y_c) = (0.25, 0.50)$ and $2\sigma^2 = 0.01$:

$$c_0(x,y) = \texttt{exp}\left(-\frac{(x-x_c)^2 + (y-y_c)^2}{2\sigma^2}\right),$$

which is numerically compactly supported. The diffusion is $\mathbf{D} = 10^{-6}\mathbf{I}$ but there is no source. For boundary conditions, we set the left side as inflow, but the right, top, and bottom sides as outflow/flow.

A uniform rectangular mesh was adopted for both flow and transport. The flow problem was solved by the WG$(P_1, P_1; AC_1)$ with $h = 1/40$. The numerical pressure and velocity can be found in Fig. 4 left panel. For the transport solver, we choose $\Delta t = T/40$. For positivity-correction based on Picard iterations, we choose the parameters as $B = 10^{10}, \texttt{MaxItr} = 100, \epsilon = 10^{-4}$. Recall $B > 0$ is a big constant used in (24) for positivity correction. Here $\epsilon = 10^{-4}$ is a threshold for discrepancy of approximate solutions in the Picard iterations. Shown in Fig. 5 are the concentration profiles at $t = 0, T/2, T$, respectively. More specifically, we use the the nodal values of the numerical concentration to generate element averages for presentation. Clearly, no negative values are observed.

7 Discussion

To conclude, we discuss extension of the mathematical and implementation techniques developed in this paper.

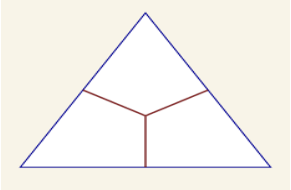

Fig. 6. TQuad: A triangle is divided into three quadrilaterals

#1. Extension to triangular meshes with $\mathcal{P}_1$ approximants Although a triangle can be divided into three convex quadrilaterals, as shown in Fig. 6, and the transport solver in this paper can be applied, we are more interested in extending the upwinding and flux correction techniques to transport solvers to

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

triangular meshes. Some preliminary results are already available. A complete account will be presented in our future work.

#2. Transport with reaction. This is nontrivial for nonlinear reaction, especially for maintaining positivity. It is being investigated by our group and will be presented in our future work.

#3. Other choices for time-marching. The implicit Euler (IE) is a good choice due to its usefulness in positivity-preserving. However, it is only 1st order accurate in temporal approximation. We are exploring the following options.
– *The 2nd order Runge-Kutta method*;
– *Leap-frog scheme*.

#4. Combination with characteristic tracking? The numerical solver studied in this paper is in the Eulerian approach. It is relatively easier for implementation, since no tracking along characteristic is involved. With the latter, Lagrangian or Eulerian-Lagrangian approaches are adopted. These two approaches incur smaller errors in temporal discretization, but need to deal with certain technical details in characteristics-tracking.

#5. Time-fractional convection-diffusion problems. Interestingly, some techniques discussed in this paper can be extended to time-fractional 2-dim convection-diffusion problems [21]. The major achievements rely on a novel 3-part decomposition of the standard L1 approximation of the Caputo derivative. The introduction of a transition term between the current and history terms plays a critical role in maintaining positivity of numerical solutions.

#6. Extension to 3-dim problems. This is undoubtedly an important project, as we aim at large-scale simulations for drug delivery and petroleum reservoir. Although there is no big changes in the numerical schemes, 3-dim implementations of dual mesh structures, upwinding, and flux corrections need further efficient strategies, which need to be consistent with our other existing techniques [5,19,20]. `C++` code modules will be furnished in our in-house packageDarcy+. This is currently investigation and will be reported in our future work.

On the list, Item #2 is challenging, Item #5 feasible, while Item #6 is promising.

Acknowledgements. We express our sincere thanks to Prof. Yonghai Li (Jilin University, China) for the stimulating discussion.

References

1. Ambartsumyan, I., Khattatov, E., Nguyen, T., Yotov, I.: Flow and transport in fractured poroelastic media. Int. J. Geo. Math. **10**, 11 (2019)
2. Feng, W., Guo, H., Kang, Y., Yang, Y.: Bound-preserving discontinuous Galerkin methods with second-order implicit pressure explicit concentration time marching for compressible miscible displacements in porous media. J. Comput. Phys. **463**, 111240 (2022)
3. Guo, H., Yang, Y.: Bound-preserving discontinuous Galerkin method for compressible miscible displacement in porous media. SIAM J. Sci. Comput. **39**, A1969–A1990 (2017)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25

4. Harper, G., Liu, J., Tavener, S., Wildey, T.: Coupling Arbogast–Correa and Bernardi–Raugel elements to resolve coupled Stokes–Darcy flow problems. Comput. Meth. Appl. Mech. Engrg. **373**, 113469 (2021)
5. Harper, G., Wang, R., Liu, J., Tavener, S., Zhang, R.: A locking-free solver for linear elasticity on quadrilateral and hexahedral meshes based on enrichment of Lagrangian elements. Comput. Math. Appl. **80**, 1578–1595 (2020)
6. Kou, J., Wang, X., Chen, H., Sun, S.: An efficient bound-preserving and energy stable algorithm for compressible gas flow in porous media. J. Comput. Phys. **473**, 111751 (2023)
7. Lan, B., Sheng, Z., Yuan, G.: A new finite volume scheme preserving positivity for radionuclide transport calculations in radioactive waste repository. Int. J. Heat Mass Trans. **121**, 736–746 (2018)
8. Lan, B., Sheng, Z., Yuan, G.: A new positive finite volume scheme for two-dimensional convection-diffusion equation. Z Angew Math Mech. **99**, e201800067 (2019)
9. Li, A., Yang, H., Gao, Y., Li, Y.: A new upwind finite volume element method for convection-diffusion- reaction problems on quadrilateral meshes. Commun. Comput. Phys. (2024)
10. Liu, J., Gu, J., Li, H., Carlson, K.H.: Machine learning and transport simulations for groundwater anomaly detection. J. Comput. Appl. Math. **380**, 112982 (2020)
11. Liu, J., Sadre-Marandi, F., Wang, Z.: DarcyLite: A Matlab toolbox for Darcy flow computation. Proc. Comput. Sci. **80**, 1301–1312 (2016)
12. Liu, J., Tavener, S., Wang, Z.: Lowest-order weak Galerkin finite element method for Darcy flow on convex polygonal meshes. SIAM J. Sci. Comput. **40**, B1229–B1252 (2018)
13. Liu, J., Tavener, S., Wang, Z.: Penalty-free any-order weak Galerkin FEMs for elliptic problems on quadrilateral meshes. J. Sci. Comput. **83**, 47 (2020)
14. Stay, M.S., Xu, J., Randolph, T.W., , Barocas, V.H.: Computer simulation of convective and diffusive transport of controlled-release drugs in the vitreous humor. Pharm. Res. **20**, 96–102 (2003)
15. Sun, S., Liu, J.: A locally conservative finite element method based on piecewise constant enrichment of the continuous galerkin method. SIAM J. Sci. Comput. **31**, 2528–2548 (2009)
16. Sun, Y., Li, Y., Liu, J.: Mathematical modeling and numerical simulations for drug release from PLGA particles. Lec. Notes Comput. Sci. **14077**, 347–360 (2023)
17. Thomée, V.: Galerkin Finite Element Methods for Parabolic Problems. Springer, 2nd edn. (2006)
18. Wang, R., Wang, Z., Liu, J.: Penalty-free any-order weak Galerkin FEMs for linear elasticity on quadrilateral meshes. J. Sci. Comput. **95**, 20 (2023)
19. Wang, Z., Harper, G., O'Leary, P., Liu, J., Tavener, S.: deal.II implementation of a weak Galerkin finite element solver for Darcy flow. Lec. Notes Comput. Sci. **11539**, 495–509 (2019)
20. Wang, Z., Liu, J.: deal.II implementation of a two-field finite element solver for poroelasticity. Lec. Notes Comput. Sci. **12143**, 88–101 (2020)
21. Yu, B., Li, Y., Liu, J.: A positivity-preserving and robust fast solver for time-fractional convection-diffusion problems. J. Sci. Comput. **98**, 59 (2024)
22. Yu, B., Li, Y., Liu, J.: Two new schemes for coupled flow and transport: Physical-property-preserving and optimal-order convergence. Comput. Math. Appl. p. Submitted (2025)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_25