J Eng Math (2007) 57:79–98
DOI 10.1007/s10665-006-9057-y

ORIGINAL PAPER

Discretization methods with embedded analytical solutions for convection–diffusion dispersion–reaction equations and applications

Jürgen Geiser

Received: 22 June 2005 / Accepted: 2 May 2006 / Published online: 11 October 2006

Abstract Numerical methods are described and results are presented for a system of convection–diffusion dispersion–reaction equations. Discretization methods that were developed earlier by the author are used. The methods allow large time steps for simulating the transport–reaction model of a waste disposal. With higher-order discretization methods, based on finite-volume methods, one may use large time steps without loss of accuracy. A multi-physical multi-dimensional equation is broken down into simpler physical and one-dimensional equations. These simpler equations are handled with locally higher-order discretization methods and the results are coupled by operator-splitting methods. An improved explicit time-discretization method, with embedded analytical solutions, for the convection–reaction equation and an implicit time-discretization diffusion–dispersion equation is described. For the numerical experiments the underlying program-tool R^3T is briefly introduced and the main concepts are presented. Benchmark problems for testing the discretization methods of higher order are described. Real-life problems for simulating radioactive-waste disposals with underlying flowing groundwater are presented and discussed.

Keywords Convection–diffusion dispersion–reaction equation · Embedded analytical solutions · Finite-volume methods · Multi-physics · Simulation of radio-active-waste disposals

1 Introduction

Our motivation for studying transport–reaction processes arose from the need to simulate waste disposals for radioactively contaminated groundwater flowing through an overlying rock. The underlying model problem is given as a convection–diffusion dispersion–reaction equation, which is analyzed and described in [1, Chapter 4], [2, Chapter 1], [3, Sect. 1.1]. The analysis of the existence and uniqueness of these convection–diffusion dispersion–reaction equations is described in the literature [4, Chapter 2]. Because of the dominant convective term, owing to a strong influence of the flow term, the behavior of the equations is governed by a hyperbolic partial differential equation; see [5, pp. 349–427]. For the numerical analysis, conservative discretization methods with stabilization of the convective term (see

J. Geiser (✉)
Department of Mathematics, Humboldt-Universität zu Berlin,
Unter den Linden 6, D-10099 Berlin, Germany
e-mail: geiser@mathematik.hu-berlin.de

[6, Chapters 1, 2], [7, Chapters 4, 12]) are considered. For the spatial-discretization methods, conservative methods, for example, finite-volume methods [7], [8, Chapter 3], flux-based methods [9], or discontinuous Galerkin methods [10] are considered, due to the conservation of mass. Because of the proposed large time scales for simulating time periods of about 10,000 years (cf. [11]), one has to utilize very large time steps. To obtain such large time steps, a combination of explicit and implicit time-discretization methods is needed [12]. Therefore the convection–reaction parts are discretized with explicit time-discretization methods and the diffusion–dispersion equation with implicit discretization methods (see [7, 13]). For coupling the two different parts of the discretized equations, one may use explicit–implicit Runge–Kutta methods, called FS-RK methods (see [14, Chapter 3]) or apply an operator-splitting method and solve each equation independently ([15, 16]). Explicit–implicit Runge–Kutta methods are, however, too complicated to implement and ineffective for large systems of equations [12] whereas operator-splitting methods are simpler and allow one to achieve at least second-order convergence [15]. For solving the implicit discretized parts, on the other hand, a multi-grid method, developed for elliptic and parabolic equations (see [17, Chapter 4], [18]), is applied (cf. [19, Chapter 2]). Adaptive and parallel software codes are important to obtain accurate and efficient results for 2d and 3d applications in complex multi-layered domains. The computations can be further accelerated by larger time steps by designing higher-order discretization methods which will be addressed in a future paper.

The paper is organized as follows. A mathematical model of contaminant transport in flowing groundwater is introduced in Sect. 2. The decoupling of the complex equation to physically simpler equations is described in Sect. 3. In Sect. 4 we introduce the discretization methods for the decoupled equations with respect to the convection–reaction equation. Analytical solutions embedded in our discretization methods are discussed in Sect. 5. We introduce the numerical solvers and concentrate on a multi-grid solver in Sect. 6. We present the software tools in Sect. 7 and results for the methods in Sect. 8. Finally we discuss future work in the area of discretization methods.

2 Mathematical model

In the following we describe an approach to solve a complicated system of convection–diffusion dispersion–reaction equations for the convection-dominant case. For the time-discretization we propose a decomposition method and split complex and multi-dimensional equations into simpler equations. The proposed operator-splitting method is based on a sequential first-order splitting (see [12, Chapter 4]) whose order can be increased by using a higher-order splitting method; see [15, 20]. For our model problem the space discretization is quite large, because of the adapted coarse grid which, however, is reduced by using higher-order methods in space. As a result, the time-discretization error, even for a first-order method, is much smaller than the spatial discretization error and does not influence the result appreciably. It follows that discretization of the space variable must be done in a very careful manner by adapting the discretization method to each part of the equation. For the convection–reaction equation we use explicit time-discretization and higher-order finite-volume methods for the space discretization. An improved higher-order discretization method for the convection–reaction equation is presented and the splitting error in time is skipped because of the embedding of the exact solution; cf. [21, Chapters 3, 5]. The method is based on analytical solutions for the convection–reaction equation for each one-dimensional flow direction. For the diffusion–dispersion equation we use an implicit time discretization and the finite-volume method for the space discretization. So for both discretization methods one can achieve higher-order accuracy. Furthermore, for the implicit methods we use preconditioned multi-grid solvers. These methods are implemented and applied in the software package R^3T; see [22, 23].

Here we deal with the following equation, see [2], given by

$$\partial_t\, R_\alpha\, c_\alpha + \nabla\cdot(\mathbf{v}c_\alpha - D\nabla c_\alpha) + \lambda_{\alpha\beta}\, R_\alpha\, c_\alpha = \sum_{\gamma\in\gamma(\alpha)} \lambda_{\gamma\alpha}\, R_\gamma\, c_\gamma \tag{1}$$

$$\text{with } \alpha = 1,\ldots,m, \tag{2}$$

where the unknowns $c_\alpha = c_\alpha(x,t)$ are defined in $\Omega\times(0,T)\subset \mathbb{R}^d\times\mathbb{R}$ with d being the space dimension. The parameters $R_\alpha \in \mathbb{R}^+$ are constant and are called retardation factors. The decay parameters are $\lambda_{\alpha\beta}$ and represent the decay rates from α to β, where $\gamma(\alpha)$ are the predecessors of element α; D is the Scheidegger diffusion–dispersion tensor and $\mathbf{v}$ is the velocity.

The main contribution of this paper is the application of a new method based on an embedded analytical solution for systems of one-dimensional convection–reaction equations with different retardation factors and constant velocities and an explicit finite-volume method for multi-dimensional convection–reaction equations.

The newly proposed method has no splitting error and is exact in each one-dimensional direction. For solving multi-dimensional transport equations we consider one-dimensional transport equations and couple the results by a correct summation of their mass transport. So large time steps and the application of coarser grids are possible with acceptable small discretization errors.

The higher-order finite-volume method based on TVD methods and constructed under the discrete minimum and maximum principle is used to reach second-order accuracy for all components.

To derive the new discretization method we concentrate on d-dimensional convection–reaction equations with equilibrium sorption. The proposed equation is given as

$$R_i\partial_t c_i + \nabla\cdot(\mathbf{v}c_i) = -\lambda_i R_i c_i + \lambda_{i-1}R_{i-1}c_{i-1}, \quad i = 1,\ldots,m, \tag{3}$$

$$c_i = (c_{1,i},\ldots,c_{d,i})^T \in \mathbb{R}^d, \tag{4}$$

where the trivial inflow and outflow boundary conditions are given by $c = 0$ and for the initial conditions $c_i(x,0) = c_{i,0}(x)$ rectangular, trapezoidal and polynomial impulses are used. Based on the one-dimensional convection–reaction equation with equilibrium sorption and initial impulses, we derive the novel discretization methods.

In the following section we describe the operator-splitting method for decoupling complex multi-physical equations into simpler physical equations.

3 Operator-splitting methods

Operator-splitting methods are developed to solve complex models in geophysical and environmental physics. They are described and applied to complex problems in [15, 16, 20] and [24, Chapter 1]. The idea underlying operator-splitting methods consists in decoupling complex equations into simpler ones that can be solved in a more efficient and accurate manner. In our case we can reduce the space errors of each simpler equation by using locally adapted spatial-discretization methods. It is sufficient to apply a first- or second-order operator splitting and couple our simpler equation with dominant space errors. So we achieve higher order in space and the time error does not influence our result by using coarse grids for the spatial discretization. The different operator-splitting methods are described in the following subsection.

3.1 Splitting methods of first order

Some first-order splitting methods are described below. We consider the following ordinary linear differential equations:

$$\partial_t c(t) = A\, c(t) \;+\; B\, c(t), \tag{5}$$

where the initial conditions are given as $c^n = c(t^n)$. The operators A and B are assumed to be bounded linear operators in the Banach space X with $A, B: X \to X$. In applications, the operators A and B correspond to physical operators, e.g. the convection and the diffusion operator.

The operator-splitting method is introduced as a method for solving two equation parts sequentially, with respect to initial conditions. The method is given as following

$$\begin{aligned}
&\frac{\partial c^*(t)}{\partial t} = Ac^*(t), \quad \text{with } c^*(t^n) = c^n,\\
&\frac{\partial c^{**}(t)}{\partial t} = Bc^{**}(t), \quad \text{with } c^{**}(t^n) = c^*(t^{n+1}).
\end{aligned} \tag{6}$$

where the time-step is defined as $\tau^n = t^{n+1} - t^n$. The solution of equation (5) is $c^{n+1} = c^{**}(t^{n+1})$.

The splitting error of the method is derived by means of a Taylor expansion; cf. [21]. We obtain the global error as

$$\begin{aligned}
\rho_n &= \frac{1}{\tau}(\exp(\tau^n(A+B)) - \exp(\tau^n B)\exp(\tau^n A))\ c(t^n)\\
&= \frac{1}{2}\tau^n[A,B]\ c(t^n) + O((\tau^n)^2),
\end{aligned} \tag{7}$$

where $[A,B] := AB - BA$ is the commutator of A and B. We get an error of the order $O(\tau^n)$ if the operators A and B do not commute, otherwise the method is exact.

3.2 Higher-order splitting methods

We can improve our method by the so-called Strang splitting method, which is of second order (cf. [15]).

The method is as follows:

$$\begin{aligned}
&\frac{\partial c^*(t)}{\partial t} = Ac^*(t), \quad \text{with } t^n \le t \le t^{n+1/2} \text{ and } c^*(t^n) = c^n,\\
&\frac{\partial c^{**}(t)}{\partial t} = Bc^{**}(t), \quad \text{with } t^n \le t \le t^{n+1} \text{ and } c^{**}(t^n) = c^*(t^{n+1/2}),\\
&\frac{\partial c^{***}(t)}{\partial t} = Ac^{***}(t), \quad \text{with } t^{n+1/2} \le t \le t^{n+1} \text{ and } c^{***}(t^{n+1/2}) = c^{**}(t^{n+1}),
\end{aligned} \tag{8}$$

and $c^{n+1} = c^{***}(t^{n+1})$ represents the result of the method.

The splitting error of this method is given as (cf. [13])

$$\begin{aligned}
\rho_n &= \frac{1}{\tau}\left(\exp(\tau^n(A+B)) - \exp\left(\frac{\tau^n}{2}A\right)\exp(\tau^n B)\exp\left(\frac{\tau^n}{2}A\right)\right)c(t^n)\\
&= \frac{1}{24}(\tau^n)^2([B,[B,A]] - 2[A,[A,B]])\ c(t^n) + O((\tau^n)^4).
\end{aligned} \tag{9}$$

We get second-order accuracy for the non-commuting case and an exact result if the operators commute.

We can improve the order by using more intermediate steps; cf. [20]. First-order splitting is applied to the convection–reaction term and the diffusion–dispersion term. The splitting is done with respect to the dominant space error in the convection term. So we split the system into a convection–reaction and a diffusion equation. The time error for such a combination can be neglected by using small time steps to reduce the error.

In the next section we present the discretization methods for the equations.

4 Discretization

For the space discretization we use finite-volume methods and for the time discretization we apply explicit or implicit Euler methods. In the next sections we introduce the notation for the space discretization and describe the discretization methods for each equation part.

4.1 Notation

The time interval $(0, T)$ is discretized in the time intervals (t^n, t^{n+1}) for $n = 0, 1, \ldots$. The computational cells are given as $\Omega_j \subset \Omega$ with $j = 1, \ldots, I$, with I being the number of the nodes.

For the application of the finite-volume technique we have to construct a dual mesh for the triangulation $\mathcal{T}$, see [25] and [26], for the domain Ω. First the finite elements for the domain Ω are given by $T^e, e = 1, \ldots, E$. The polygonal computational cells Ω_j are related to the vertices x_j of the triangulation.

The notation for the relation between the neighbor cells and the concerned volume of each cell is given as follows. Let $V_j = |\Omega_j|$ and the set Λ_j denote the neighbor-point x_k to the point x_j. The boundary of the cell j and k is denoted as Γ_{jk}. We define the flux over the boundary Γ_{jk} as

$$v_{jk} = \int_{\Gamma_{jk}} \mathbf{n} \cdot \mathbf{v} \, \mathrm{d}s. \tag{10}$$

The inflow flux is given as $v_{jk} < 0$, the outflow flux is $v_{jk} > 0$. The antisymmetry of the fluxes is denoted as $v_{jk} = -v_{kj}$. The total outflow flux is given as

$$v_j = \sum_{k \in \mathrm{out}(j)} v_{jk}. \tag{11}$$

The idea of the finite volumes is to construct an algebraic system of equations to express the unknowns $c_j^n \approx c(x_j, t^n)$. The initial values are given by c_j^0. The expression for the interpolation schemes can be given naturally in two ways; the first possibility is given with the primary mesh of the finite elements

$$c^n = \sum_{j=1}^{I} c_j^n \phi_j(x) \tag{12}$$

with ϕ_j being the standard global finite-element basis functions [25]. The second possibility is given by the dual mesh of the finite volumes with,

$$\hat{c}^n = \sum_{j=1}^{I} c_j^n \varphi_j(x), \tag{13}$$

where φ_j are piecewise constant discontinuous functions defined by $\varphi_j(x) = 1$ for $x \in \Omega_j$ and $\varphi_j(x) = 0$ otherwise.

4.2 First-order discretization method of the convection equation

We deal with the following convection equation

$$\partial_t R\, c - \mathbf{v} \cdot \nabla c = 0 \tag{14}$$

with the simple boundary condition $c = 0$ for the inflow and outflow boundary and the initial values $c(x_j, 0) = c_j^0(x)$. We use a piecewise constant discretization method with the upwind discretization done as

in [25] and get

$$V_j R c_j^{n+1} = V_j R c_j^n - \tau^n \sum_{k\in \mathrm{out}(j)} v_{jk} c_j^n + \tau^n \sum_{k\in \mathrm{in}(j)} c_k^n v_{kj},$$

$$V_j R c_j^{n+1} = c_j^n (R V_j - \tau^n v_j) + \tau^n \sum_{k\in \mathrm{in}(j)} c_k^n v_{kj}. \tag{15}$$

The explicit time discretization has to satisfy the discrete minimum–maximum property [25], and we get the following restriction for the time steps

$$\tau_j = \frac{R\, V_j}{v_j}, \quad \tau^n \le \min_{j=1,\ldots,I} \tau_j. \tag{16}$$

To obtain improved spatial discretization methods and apply larger time steps, we introduce a reconstruction with a linear polynomial as a higher test-function in the next subsection.

4.3 Higher-order discretization method for the convection equation

The reconstruction is based on the Godunov method and we apply a limiter function that fulfills the local min–max property. The method is explained in [25], but we discuss the algorithm in the following section.

The linear polynomials are reconstructed by the element-wise gradient and are given as

$$u^n(x_j) = c_j^n, \tag{17}$$

$$\nabla u^n|_{V_j} = \frac{1}{V_j} \sum_{e=1}^{E} \int_{T^e \cap \Omega_j} \nabla c^n \mathrm{d}x \quad \text{with } j = 1, \ldots, I. \tag{18}$$

The piecewise linear functions are denoted as follows:

$$u_{jk}^n = c_j^n + \psi_j \nabla u^n|_{V_j} (x_{jk} - x_j) \quad \text{with } j = 1, \ldots, I, \tag{19}$$

where $\psi_j \in (0, 1)$ is the limiter function and, based on this, Eq. 25 fulfills the discrete minimum–maximum property as described in [25].

We also use the limitation of the flux to get non-overshooting, when transporting the mass and obtain the maximal time step. We get the correct restriction due to the flux limiter and obtain the following concentration as

$$\tilde{u}_{jk}^n = u_{jk}^n + \frac{\tau_j}{\tau^n} \left(c_j^n - u_{jk}^n \right). \tag{20}$$

Using all the previous schemes, we may write the discretization for the second order in the following form

$$R V_j c_j^{n+1} = R V_j c_j^n - \tau^n \sum_{k\in \mathrm{out}(j)} \tilde{u}_{jk}^n v_{jk} + \tau^n \sum_{l\in \mathrm{in}(j)} \tilde{u}_{lj}^n v_{lj}. \tag{21}$$

Based on this discretization method we can embed the reaction equation as a local effect, described in the next subsection.

4.4 Discretization method for the convection–reaction equation based on embedded one-dimensional analytical solutions

We apply Godunov's method for the discretization method, cf. [7], and extend the formulation with the analytical solution of the convection–reaction equations. We reduce the multi-dimensional equation to

one-dimensional equations and solve each equation exactly. The one-dimensional solution is multiplied by the underlying volume and we get the mass formulation. The one-dimensional mass is embedded into the multi-dimensional mass formulation and we obtain the discretization of the multi-dimensional equation.

The algorithm is given in the following manner

$$\partial_t c_l + \nabla\cdot\mathbf{v}_l c_l = -\lambda_l c_l + \lambda_{l-1} c_{l-1} \quad \text{with } l = 1,\ldots,m.$$

The velocity vector $\mathbf{v}$ is divided by R_l. The initial conditions are given by $c_1^0 = c_1(x,0)$, else $c_l^0 = 0$ for $l = 2,\ldots,m$ and the boundary conditions are trivially $c_l = 0$ for $l = 1,\ldots,m$.

We first calculate the maximal time step for cell j and concentration i by use of the total outflow fluxes

$$\tau_{i,j} = \frac{V_j\, R_i}{v_j}, \quad v_j = \sum_{k\in\mathrm{out}(j)} v_{jk}.$$

We get the restricted time step with the local time steps of cells and their components

$$\tau^n \le \min_{\substack{i=1,\ldots,m\\ j=1,\ldots,I}} \tau_{i,j}.$$

The velocity of the discrete equation is given by

$$v_{i,j} = \frac{1}{\tau_{i,j}}.$$

We calculate the analytical solution of the mass, cf. Sect. 5, using Eqs. 42 and 44

$$m^n_{i,jk,\mathrm{out}} = m_{i,\mathrm{out}}(a,b,\tau^n, v_{1,j},\ldots,v_{i,j},R_1,\ldots,R_i,\lambda_1,\ldots,\lambda_i),$$
$$m^n_{i,j,\mathrm{rest}} = m^n_{i,j}\, f(\tau^n, v_{1,j},\ldots,v_{i,j},R_1,\ldots,R_i,\lambda_1,\ldots,\lambda_i),$$

where $a = V_j R_i (c^n_{i,jk} - c^n_{i,jk'})$, $b = V_j R_i c^n_{i,jk'}$ and $m^n_{i,j} = V_j R_i c^n_{i,j}$. Further $c^n_{i,jk'}$ is the concentration at the inflow- and $c^n_{i,jk}$ is the concentration at the outflow-boundary of the cell j.

The discretization with the embedded analytical mass is calculated by means of

$$m^{n+1}_{i,j} - m^n_{i,\mathrm{rest}} = -\sum_{k\in\mathrm{out}(j)} \frac{v_{jk}}{v_j}\, m_{i,jk,\mathrm{out}} + \sum_{l\in\mathrm{in}(j)} \frac{v_{lj}}{v_l}\, m_{i,lj,\mathrm{out}},$$

where v_{jk}/v_j is the re-transformation for the total mass $m_{i,jk,\mathrm{out}}$ in the partial mass $m_{i,jk}$. In the next time step the mass is given as $m^{n+1}_{i,j} = V_j\, c^{n+1}_{i,j}$ and in the old time step it is the rest mass for the concentration i. The proof is given in [21]. In the next section we derive an analytical solution for the benchmark problem; cf. [27], [28, Chapter 3].

4.5 Discretization of the reaction equation

The reaction equation is an ordinary differential equation given as follows:

$$\partial_t R_i c_i = -\lambda_i R_i c_i + \lambda_{i-1} R_{i-1} c_{i-1}, \tag{22}$$

where $i = 1,\ldots,m$ and we set $\lambda_0 = 0$. The decay factors are $\lambda_i \ge 0{\cdot}0$ and the retardation factors are $R_i > 0{\cdot}0$. The initial conditions are $c_1(x,t^0) = c_{01}$ and $c_i(x,t^0) = 0$ with $i = 2,\ldots,m$.

We can derive the solutions for these equations, cf. [29, 30], as

$$c_i = c_{01}\frac{R_1}{R_i}\Lambda_i \sum_{j=1}^{i} \Lambda_{j,i} \exp(-\lambda_j t), \tag{23}$$

where $i = 1, \ldots, m$. The solutions are defined for the case $\lambda_j \neq \lambda_k$ with $j \neq k$ and $j, k \in 1, \ldots, M$. The factors Λ_i and $\Lambda_{j,i}$ are given as

$$\Lambda_i = \prod_{j=1}^{i-1} \lambda_j, \quad \Lambda_{j,i} = \prod_{\substack{j=1 \\ j \neq k}}^{i} \frac{1}{\lambda_k - \lambda_j}. \tag{24}$$

For pairwise equal reaction factors the solution has been derived in [21].

In the next subsection the discretization of the diffusion–dispersion equation will be introduced.

4.6 Discretization of the diffusion–dispersion equation

We discretize the diffusion–dispersion equation with implicit time discretization and finite-volume method for the following equation

$$\partial_t Rc - \nabla \cdot (D \nabla c) = 0, \tag{25}$$

where $c = c(x, t)$ with $x \in \Omega$ and $t \geq 0$. The diffusion–dispersion tensor $D = D(x, \mathbf{v})$ is obtained by the Scheidegger approach, cf. [31], with $\mathbf{v}$ denoting the velocity and $R > 0{\cdot}0$ the retardation factor.

The boundary values on $\partial\Omega$ are represented by $\mathbf{n} \cdot D \ \nabla c(x, t) = 0$, whereas the initial conditions are given by $c(x, 0) = c_0(x)$.

We integrate Eq. 25 over space and time and obtain

$$\int_{\Omega_j} \int_{t^n}^{t^{n+1}} \partial_t R(c) \, \mathrm{d}t \, \mathrm{d}x = \int_{\Omega_j} \int_{t^n}^{t^{n+1}} \nabla \cdot (D \nabla c) \mathrm{d}t \, \mathrm{d}x. \tag{26}$$

The time integration is done by the backward Euler method and the diffusion–dispersion term is lumped, cf. [21]:

$$\int_{\Omega_j} (R(c^{n+1}) - R(c^n)) \mathrm{d}x = \tau^n \int_{\Omega_j} \nabla \cdot (D \nabla c^{n+1}) \, \mathrm{d}x. \tag{27}$$

Equation (27) is discretized over the space after using the Green formula.

$$\int_{\Omega_j} (R(c^{n+1}) - R(c^n)) \mathrm{d}x = \tau^n \int_{\Gamma_j} D\mathbf{n} \cdot \nabla c^{n+1} \mathrm{d}\gamma, \tag{28}$$

where Γ_j is the boundary of the finite-volume cell Ω_j. We use the approximation in space; see [21].

The spatial integration of (28) is done by the mid-point rule over the finite boundaries and given as

$$V_j R\left(c_j^{n+1}\right) - V_j R\left(c_j^n\right) = \tau^n \sum_{e \in \Lambda_j} \sum_{k \in \Lambda_j^e} |\Gamma_{jk}^e| \mathbf{n}_{jk}^e \cdot D_{jk}^e \nabla c_{jk}^{e,n+1}, \tag{29}$$

where $|\Gamma_{jk}^e|$ is the length of the boundary element Γ_{jk}^e. The gradients are calculated with the piecewise finite-element function ϕ_l, cf. (12), and we obtain

$$\nabla c_{jk}^{e,n+1} = \sum_{l \in \Lambda^e} c_l^{n+1} \nabla \phi_l \left(\mathbf{x}_{jk}^e\right). \tag{30}$$

Using the difference notation for the neighboring points j and l, cf. [26], we obtain the following discretized equation

$$V_j R\left(c_j^{n+1}\right) - V_j R\left(c_j^n\right) = \tau^n \sum_{e \in \Lambda_j} \sum_{l \in \Lambda^e \setminus \{j\}} \left(\sum_{k \in \Lambda_j^e} \left| \Gamma_{jk}^e \right| \mathbf{n}_{jk}^e \cdot D_{jk}^e \nabla \phi_l(\mathbf{x}_{jk}^e) \right) \left(c_j^{n+1} - c_l^{n+1}\right), \tag{31}$$

where $j = 1, \ldots, m$.

In the next section the analytical solutions of the convection–reaction equation will be described.

5 Analytical solutions

In this section we deal with a system of one-dimensional convection–reaction equations represented by

$$\partial_t c_i + v_i \partial_x c_i = -\lambda_i c_i + \lambda_{i-1} c_{i-1} \tag{32}$$

for $i = 1, \ldots, m$ with m the number of components. The unknown solutions $c_i = c_i(x,t)$ are denoted as the contaminant concentrations. The velocities v_i are piecewise constant and in general different. We deal with constant reaction rates λ_i and the underlying solution space is given by $(0,\infty) \times (0,T)$.

A simple (irreversible) decay chain is assumed, with $\lambda_0 = 0$, and for each contaminant we have only a single source term given as $\lambda_{i-1} c_{i-1}$. For simplicity, we assume that $v_i > 0$ for $i = 1, \ldots, m$. The analytical solutions are derived for piecewise linear initial conditions; for the use of arbitrary piecewise polynomial functions we refer to [21].

We propose Dirichlet boundary conditions for the inflow boundary $x = 0$ and the initial conditions are defined as

$$\begin{aligned} c_1(x,0) &= \begin{cases} ax + b\,, & x \in (0,1) \\ 0 & \text{otherwise} \end{cases}, \\ c_i(x,0) &= 0\,, \quad i = 2, \ldots, m, \end{aligned} \tag{33}$$

where $a, b \in \mathbb{R}^+$ are constants.

We apply the Laplace transformation and transform the partial differential equation to an ordinary differential equation; see [32]. The ordinary differential equations can be solved analytically, cf. [33], [34, Chapter 1], and the solution re-transformed in the original space of the partial differential equations. The results are applied for the discretization methods of the one-dimensional convection–reaction equation; see [21].

The analytical solutions are given as

$$c_1(x,t) = \exp(-\lambda_1 t) \begin{cases} 0 & 0 \le x < v_1 t, \\ a(x - v_1 t) + b & v_1 t \le x < v_1 t + 1 \\ 0 & v_1 t + 1 \le x, \end{cases}, \tag{34}$$

$$c_i(x,t) = \Lambda_i \left(\sum_{j=1}^{i} \exp(-\lambda_j t) \Lambda_{j,i} \sum_{\substack{k=1 \\ k \neq j}}^{i} \Lambda_{jk,i} A_{jk} \right), \tag{35}$$

$$A_{jk} = \begin{cases} 0 & 0 \le x < v_j t, \\ a(x - v_j t) & \\ + \left(b - \frac{a}{\lambda_{jk}} \right) (1 - \exp(-\lambda_{jk}(x - v_j t))) & v_j t \le x < v_j t + 1 \\ \left(b - \frac{a}{\lambda_{jk}} + a \right) \exp(-\lambda_{jk}(x - v_j t - 1)) & \\ - \left(b - \frac{a}{\lambda_{jk}} \right) \exp(-\lambda_{jk}(x - v_j t)) & v_j t + 1 \le x \end{cases}, \tag{36}$$

subject to the following assumptions

$$v_i \neq v_j, \lambda_i \neq \lambda_j, \lambda_{ij} \neq \lambda_{ik} \text{ and } v_i \neq v_j \ \wedge \ \lambda_i \neq \lambda_j, \quad \forall\, i,j,k = 1, \ldots, M, \text{ if } i \neq j \wedge i \neq k \wedge j \neq k.$$

The parameters λ_{jk} and Λ_i are given as

$$\lambda_{kj} = \lambda_{jk} := \frac{\lambda_j - \lambda_k}{v_j - v_k}, \quad \Lambda_i := \prod_{j=1}^{i-1} \lambda_j, \tag{37}$$

and the factors $\Lambda_{j,i}$ and $\Lambda_{jk,i}$ are computed as

$$\Lambda_{j,i} = \left(\prod_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_k - \lambda_j} \right), \quad \Lambda_{jk,i} = \left(\prod_{\substack{l=1 \\ l\neq j \\ l\neq k}}^{i} \frac{1}{\lambda_k - \lambda_j} \right). \tag{38}$$

The solutions (34) and (35) are used in the discretization methods for the embedded analytical mass. In the next subsection the mass reconstruction based on the analytical solution is presented.

5.1 Mass reconstruction

To simplify the calculations we consider the norm-interval $(0, 1)$. By multiplying the analytical solution, referred to as a concentration, with the underlying length of the finite interval, we obtain the analytical mass. The mass is computed in the following manner.

First we construct the total mass given as

$$m_{i,\mathrm{sum}}(t) = m_{i,\mathrm{rest}}(t) + m_{i,\mathrm{out}}(t). \tag{39}$$

The total mass is calculated by the solution of the ordinary equation and is equal to

$$m_{i,\mathrm{sum}}(t) = \prod_{j=1}^{i-1} \lambda_j \left(\sum_{j=1}^{i} \left(\prod_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_k - \lambda_j} \right) \exp(-\lambda_j t) \right) \left(\frac{a}{2} + b \right).$$

Then we compute the residual mass as the mass that is retained in the finite cell i; cf. Fig. 1. The residual mass is described in [21] and given as:

$$\begin{aligned} m_{i,\mathrm{rest}}(t) = \prod_{j=1}^{i-1} \lambda_j \sum_{j=1}^{i} \left(\prod_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_k - \lambda_j} \right) \exp(-\lambda_j t) \left(a \frac{(1 - v_j t)^2}{2} + b \left(1 - v_j t - \sum_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_{jk}} \right) \right. \\ \left. - a(1 - v_j t) \left(\sum_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_{jk}} \right) + a \left(\sum_{\substack{k=1 \\ k\neq j}}^{i} \frac{1}{\lambda_{jk}} \left(\sum_{\substack{l\geq k \\ l\neq j}}^{i} \frac{1}{\lambda_{jl}} \right) \right) \right), \end{aligned} \tag{40}$$

where the parameters λ_{jk} are given in Eq. 37. The outflowing mass is derived as the following difference

$$m_{i,\mathrm{out}}(\tau^n) = m_{i,\mathrm{sum}}(\tau^n) - m_{i,\mathrm{rest}}(\tau^n), \tag{41}$$

$$m_{i,\mathrm{out}}(\tau^n) = m_{i,\mathrm{out}}(a, b, \tau^n, v_1, \ldots, v_i, R_1, \ldots, R_i, \lambda_1, \ldots, \lambda_i), \tag{42}$$

$$m_{i,\mathrm{sum}}(\tau^n) = f_i^n \left(a\frac{1}{2} + b \right), \tag{43}$$

$$f_i^n(\tau^n) = f(\tau^n, v_1, \ldots, v_i, R_1, \ldots, R_i, \lambda_1, \ldots, \lambda_i). \tag{44}$$

where τ^n is the time-step, $v_1, \ldots, v_i$ are the velocity-, $R_1, \ldots, R_i$ are the retardation parameters, $\lambda_1, \ldots, \lambda_i$ the reaction parameters and a, b are the parameters for the linear impulse; cf. [1]. In the next section we describe the solution method that is used for the computations.

6 Solvers

For solving the implicit discretized diffusion–dispersion equation we apply iterative methods. We have the fully discretized equation for our diffusion–dispersion equation:

$(I - \tau\tilde{A})u^{n+1}(\tilde{x}) = u^n(\tilde{x})$,

where $\tilde{A}$ is the stiffness matrix obtained from the spatial discretization, u^{n+1} is the solution of the new time step, u^n is the known solution of the old time step and the coordinates of the grid points are given as $\tilde{x} \in \Omega_h$. For simplicity we deal with the linear system $Ax = b$, where $A = I - \tau\tilde{A}$, $b = u^n(\tilde{x})$ and $x = u^{n+1}(\tilde{x})$. Because of the local discretization methods and the resulting sparse matrix A, an efficient multi-grid solver is applied; cf. [17].

We will briefly introduce the methods for an intensive study and refer to the literature [18, 35]. We deal with the linear-equation system

$$Ax = b, \quad A \in \mathbb{R}^{I\times I}, \quad b \in \mathbb{R}^{I}, \tag{45}$$

where A is regular, x is the unknown and b is the right-hand side.

The iteration method is given as

$$x^{m+1} = Mx^m + Nb, \quad m \in \mathbb{N}, \tag{46}$$

where b is the right-hand side. A consistent iteration method is obtained for

$$M = \mathbb{1} - NA, \tag{47}$$

where $\mathbb{1}$ is the identity matrix. The matrix M is called the iteration matrix.

The usual notation for the iteration method is to express (46) in a normal form given as

$$x^{m+1} = x^m - N(Ax^m - b). \tag{48}$$

The iteration method (46) is applied for the multi-grid method [17]. We introduce the multi-grid method

$$M_0^{MG} := 0, \tag{49}$$

$$M_1^{MG} := M_1^{ZG}, \tag{50}$$

$$M_l^{MG} := M_l^{ZG} + S_l^{\nu_2}\, p\, \left(M_{l-1}^{MG}\right)^{\gamma}\, A_{l-1}^{-1}\, rA_l\, S_l^{\nu_1}, \tag{51}$$

where S_l is the smoother, p is the prolongation, r is the restriction, ν_1 are the pre-smoothing steps and ν_2 the post-smoothing steps. The coarse-grid correction M_l^{MGG}, also known as iteration matrix, is defined as

$$M_l^{MGG} := \mathbb{1} - p\, \left(I - (M_{l-1}^{MG})^{\gamma}\right)\, A_{l-1}^{-1}\, r\, A_l. \tag{52}$$

We use these iteration methods to solve our equations on a grid hierarchy, cf. [21].

7 Software tools

The methods described in the previous sections have been implemented in our software tool R^3T. The software package R^3T is developed for solving transport-reaction equations for multiple species in flowing groundwater in porous media. We forced the solution of the convection-dominant equations and improved the discretizations to use coarser grids and larger time steps. The package includes error estimators, solvers and discretization methods. For the parameter of the equation and the velocity field we use input files to set the different values. These input files are read in run-time and for a new computation we could change the values for an updated configuration. The solutions of the equations are written in output files and could be used for different post-processors, e.g. visualization-programs; cf. [36, Chapters 1–3]. So for these assumptions we could test different waste-case scenarios for different initial conditions; see Sect. 8.

The tool *R3T* is based on the software tool *ug*, cf. [37], which is based on unstructured grids. The methods for these unstructured grids are implemented in different libraries. Based on the grid hierarchy, the solvers of the discretized equations with respect to adaptive methods are programmed. The programming ideas consist in combining flexible tools with common libraries of solvers, discretization methods, error estimators and a flexible application level for applications of physical and chemical models.

The numerical experiments are presented in the next section with benchmark and waste-case scenarios.

8 Numerical experiments

The methods introduced in the previous sections are verified in Sect. 8.1 where we compare the numerical results with analytical solutions.

In Sect. 8.2 we apply the improved modified methods to complex waste-case scenarios. An introduction to the waste cases in 2d and 3d are given and the results and computation time are discussed.

8.1 Benchmark problems

We compare the results of the standard method with the modified method. The standard method is based on the operator-splitting method with the convection and reaction equation. This method has a splitting error in time of order $O(\tau^n)$. The modified method described in Sect. 4 is based on the discretization method with embedded analytical solutions for the convection–reaction equation.

We start with the one-dimensional problem and compare the results with the analytical solution.

8.1.1 Transported triangle

For the first experiment we use a one-dimensional benchmark problem with delta initial conditions. The analytical solution is given by Eqs. 34 and 35 and we compare the analytical solution with the numerical ones.

We calculate the solutions on a two-dimensional domain for which the velocity field is constant in the x-direction with constant values of $\mathbf{v}=(1{\cdot}0, 0{\cdot}0)^T$. We use only the convection–reaction equation with 4 components, given in the form

$$R_i \partial_t c_i + \mathbf{v}\cdot\nabla c_i = -R_i\lambda_i c_i + R_{i-1}\lambda_{i-1}c_{i-1} \quad \text{with } i = 1,\ldots,4, \tag{53}$$

where the inflow/outflow boundary condition is given by $\mathbf{n}\cdot\mathbf{v}\ c_i = 0{\cdot}0$ which means no inflow nor outflow. The initial conditions are given as

$$c_1(x,0) = \begin{cases} -x+1, & 0 \le x \le 1, \\ 0 \quad\ , & \text{otherwise}, \end{cases} \tag{54}$$

$$c_i(x,0) = 0{\cdot}0, \quad i = 2,\ldots,4. \tag{55}$$

For the one-dimensional problem we might compare the numerical solutions with the analytical ones derived in the previous sections. We use the L_1-norm to compare the solutions, which is given by

$$E^l_{L_1} := \sum_{i=1,\ldots,m} V_i |c^n_l(x_j,y_j,t^n) - C_l(x_i,y_i,t^n)| \quad \text{with } l = 1,\ldots,4, \tag{56}$$

where $c^n_l(x_i,y_i,t^n)$ is the numerical solution, while $C_l(x_i,y_i,t^n)$ is the analytical solution; see (34) and (35). We apply the L_1-norm as error mass for the convection–reaction equation; cf. [6].

The model domain is given as a rectangle of 8×1 units. The initial coarse grid is given as 8 quadratic unit elements, the uniform refinements are defined upto level 7 (131072 Elements). We choose the parameters to get results at the end of the same maximum value, so that we would not see the influence of numerical effects. For the first test we use the following parameters: We use the decay rates of $\lambda_1 = 0{\cdot}4,\ \lambda_2 = 0{\cdot}3,\ \lambda_3 = 0{\cdot}2,\ \lambda_4 = 0$ and the retardation factors $R_1 = 1,\ R_2 = 2,\ R_3 = 4,\ R_4 = 8$.

Our model time is given as $t = 0,\ldots,6$. We compare the results at the final time $t = 6$. The L_1-norm is computed and the numerical convergence rate given as

$$\rho = \left(\log\left(E^l_{L_1}\right) - \log\left(E^{l-1}_{L_1}\right)\right) / \log(0{\cdot}5) \tag{57}$$

for the computed levels $l = 4,\ldots,7$.

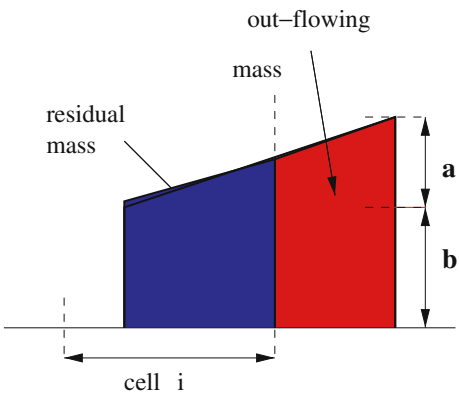

Fig. 1 Residual and out-flowing mass

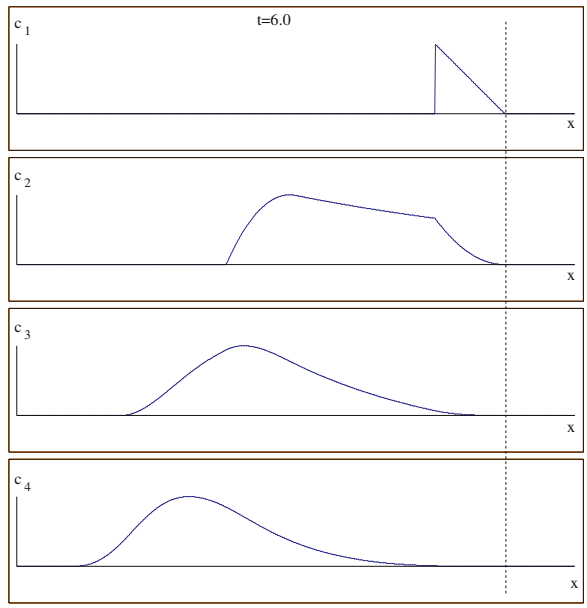

Fig. 2 Concentration for the four components with ascending retardation factors at time $t = 6$

The first results are presented with the standard method and the L_1-errors are given in Table 1. The values for the numerical convergence-orders are denoted in Table 2. The results of the calculations are of first order for all components, because of the first-order operator-splitting method.

The following results are obtained by the modified method. We run the application with the same parameters as for the standard method. The L_1-errors for the modified method for the different time- and grid-widths are presented in Table 3.

The numerical convergence orders for the modified method are calculated and presented in Table 4. For the first component we get first-order accuracy because of the space discretization for the discontinuous impulse which is of first order. The results of the next components are of higher order. We can skip the error for the time discretization and for the space discretization we obtain a higher order because of the continuous impulses.

Table 1 The L_1-errors computed with the standard method

l	$E^1_{L_1}$	$E^2_{L_1}$	$E^3_{L_1}$	$E^4_{L_1}$
4	$2{\cdot}666 \times 10^{-3}$	$9{\cdot}853 \times 10^{-4}$	$9{\cdot}77 \times 10^{-4}$	$4{\cdot}132 \times 10^{-4}$
5	$1{\cdot}297 \times 10^{-3}$	$4{\cdot}740 \times 10^{-4}$	$4{\cdot}805 \times 10^{-4}$	$2{\cdot}013 \times 10^{-4}$
6	$6{\cdot}148 \times 10^{-4}$	$2{\cdot}328 \times 10^{-4}$	$2{\cdot}377 \times 10^{-4}$	$9{\cdot}925 \times 10^{-5}$
7	$2{\cdot}969 \times 10^{-4}$	$1{\cdot}154 \times 10^{-4}$	$1{\cdot}181 \times 10^{-4}$	$4{\cdot}925 \times 10^{-5}$

Table 2 The convergence orders for the L_1-errors with the standard method

l	$\rho^1_{L_1}$	$\rho^2_{L_1}$	$\rho^3_{L_1}$	$\rho^4_{L_1}$
4				
5	1·0394	1·0556	1·023	1·0374
6	1·077	1·0257	1·015	1·0202
7	1·0501	1·0124	1·009	1·0109

Table 3 The L_1-errors computed with the modified method

l	$E^1_{L_1}$	$E^2_{L_1}$	$E^3_{L_1}$	$E^4_{L_1}$
4	$2{\cdot}666 \times 10^{-3}$	$3{\cdot}451 \times 10^{-4}$	$6{\cdot}719 \times 10^{-5}$	$2{\cdot}376 \times 10^{-5}$
5	$1{\cdot}297 \times 10^{-3}$	$1{\cdot}072 \times 10^{-4}$	$1{\cdot}669 \times 10^{-5}$	$5{\cdot}573 \times 10^{-6}$
6	$6{\cdot}148 \times 10^{-4}$	$3{\cdot}374 \times 10^{-5}$	$4{\cdot}251 \times 10^{-6}$	$1{\cdot}374 \times 10^{-6}$
7	$2{\cdot}969 \times 10^{-4}$	$1{\cdot}117 \times 10^{-5}$	$1{\cdot}091 \times 10^{-6}$	$3{\cdot}442 \times 10^{-7}$

Table 4 The convergence-orders for the L_1-errors with the modified method

l	$\rho^1_{L_1}$	$\rho^2_{L_1}$	$\rho^3_{L_1}$	$\rho^4_{L_1}$
4				
5	1·0394	1·686	2·009	2·092
6	1·077	1·667	1·973	2·0201
7	1·0501	1·594	1·962	1·997

The results of the computations are presented for the final time $t = 6$ in Fig. 2. The first component is less retarded and therefore runs till the end of the interval. The next components are decreased, stronger retarded and therefore spread out. Because of the coupling with the previous component, the next components run till the end of the first component. The last component is spread out from the first part of the interval till the end.

In the next section we will present a benchmark problem for a two-dimensional case. We also derive the analytical solution.

8.1.2 Rotating pyramid

This benchmark problem is introduced in the literature as a rotating Gaussian impulse; cf. [9]. To apply this problem also to a system of convection–reaction equations, we modify the problem for our derived analytical one-dimensional solutions. The modification is in the transformation of the Cartesian coordinates to polar coordinates. Because of this transformation the problem can be reduced to a one-dimensional problem. We can project the initial conditions, which are triangles, to polar coordinates. So on each circle the same velocity is obtained and the solution can be transformed back to that of the original one-dimensional problem. The basic reconstruction is done in [21] whereas next we present the main ideas.

The transformation from Cartesian (x, y) to polar (r, α) coordinates is given by

$$r = \sqrt{x^2 + y^2}, \quad \alpha = \arctan\left(\frac{y}{x}\right), \quad \epsilon(r) = r\alpha_0 \tag{58}$$

with α_0 as the initial angle and $\epsilon(r)$ the length of the circular arc with radius r.

First we transform the triangular impulse on the cylinder surface and get a continuous impulse. Then we transfer the continuity in the r-direction with the dependency of the initial concentration $c_0(r)$. The transformation is given by

$$r_{\text{med}} = \frac{r_a + r_b}{2}, \tag{59}$$

$$c_0(r) = c_{\text{init}} \begin{cases} \frac{2}{r_b - r_a}(r - r_a) & r_a \le r \le r_{\text{med}}, \\ \frac{-2}{r_b - r_a}(r - r_b) & r_{\text{med}} \le r \le r_b \\ 0.0 & \text{otherwise}, \end{cases} \quad c_{init} \in \mathbb{R}^+, \text{ (initial-concentration)}. \tag{60}$$

Table 5 The L_1-error and the convergence rate for modified method with embedded analytical solution

l	$E^1_{L_1}$	$\rho^1_{L_1}$	$E^2_{L_1}$	$\rho^2_{L_1}$	$E^3_{L_1}$	$\rho^3_{L_1}$	$E^4_{L_1}$	$\rho^4_{L_1}$
4	$7{\cdot}12 \times 10^{-3}$		$5{\cdot}80 \times 10^{-4}$		$3{\cdot}09 \times 10^{-5}$		$8{\cdot}28 \times 10^{-7}$	
5	$2{\cdot}74 \times 10^{-3}$	1·377	$2{\cdot}14 \times 10^{-4}$	1·44	$1{\cdot}12 \times 10^{-5}$	1·46	$2{\cdot}86 \times 10^{-7}$	1·53
6	$1{\cdot}10 \times 10^{-3}$	1·32	$8{\cdot}82 \times 10^{-5}$	1·27	$4{\cdot}90 \times 10^{-6}$	1·19	$1{\cdot}20 \times 10^{-7}$	1·25
7	$4{\cdot}40 \times 10^{-4}$	1·322	$3{\cdot}50 \times 10^{-5}$	1·33	$1{\cdot}90 \times 10^{-6}$	1·37	$4{\cdot}80 \times 10^{-8}$	1·32

This initial impulse rotates over the angle α. The length of the arc is calculated as

$$x_{\text{arc}}(r, \alpha) = r\,\alpha, \tag{61}$$

where r is the radius of the point (x, y) and α is the angle. The velocity is given in the divergence-free form

$$\mathbf{v} = \begin{pmatrix} -4{\cdot}0\; y \\ 4{\cdot}0\; x \end{pmatrix} \tag{62}$$

and in polar-coordinates it is given as

$$v = 4{\cdot}0\; r. \tag{63}$$

The initialization for the rotating pyramid is given by

$$u_{1,\text{init}} = u_{1,Tri}(x_{\text{arc}}(r, \alpha_0), t_0, \epsilon(r), c_0(r), v_1, \lambda_1), \tag{64}$$

$$u_{i,\text{init}} = 0{\cdot}0 \quad \text{with } i = 2, \ldots, m \tag{65}$$

with $t_0 = 0{\cdot}0$, $v_1 = v/R_1$; m is the number of components and $u_{1,\text{Tri}}$ the analytical solution of a convection–reaction equation with a triangular impulse.

The analytical solution for an arbitrary time is given as

$$u_{i,\text{Tri}} = u_{i,\text{Tri}}(x_{\text{arc}}(r, \alpha), t, \epsilon(r), c_0(r), v_1, \ldots, v_i, \lambda_1, \ldots, \lambda_i), \tag{66}$$

where $i = 1, \ldots, m$ and $v_i = v/R_i$.

We compute the example for four components. The retardation factors are $R_1 = 1{\cdot}0$, $R_2 = 2{\cdot}0$, $R_3 = 4{\cdot}0$, $R_4 = 8{\cdot}0$ and the reaction factors are $\lambda_1 = 1{\cdot}5$, $\lambda_2 = 1{\cdot}4$, $\lambda_3 = 1{\cdot}3$, $\lambda_4 = 0{\cdot}0$. The height of the pyramid is $c_{\text{init}} = 1$, the base area of the pyramid has a radius $0{\cdot}125 \leq r \leq 0{\cdot}375$ and the initial angle $\alpha_0 = 0{\cdot}22$. The next components are initialized with 0·0. The boundary conditions are trivial inflow and outflow conditions. We have a domain with $[-0{\cdot}5, 0{\cdot}5] \times [-0{\cdot}5, 0{\cdot}5]$ and the coarse grid consists of only one element. We maximally refine till grid-level 7. The time steps are fixed at that level and satisfy the Courant number 0·5. We calculate up to the time $t = \pi/4$.

For the modified method Table 5 reports the results for the L_1-error and convergence rates. For all components we reach higher-order results because of the higher-order modified method. The results are displayed in Figs. 3 and 4. In Fig. 3 we present the initialization whereas Fig. 4 displays the results at $t = \pi/4$. High concentrations are plotted in dark grey, no concentrations are plotted in a lighter shade of grey.

The concentrations of the higher components are more strongly retarded. The first component is transported furthest and rotated up to half of the circle. The previous components are spread out till the next components. The two-dimensional solutions also satisfy our theoretical results.

In the next section we present the complex waste-case scenarios of a waste disposal in a salt dome.

8.2 Waste-case scenarios

We calculate scenarios of waste cases that help to draw new conclusions for waste disposals in salt domes.

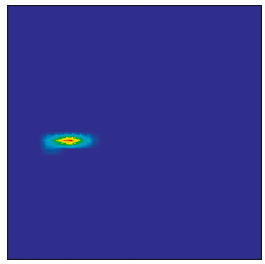

Fig. 3 The initial concentration of the first component at the time point $t = 0$, the other components are initialized with 0

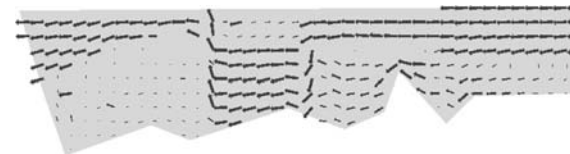

Fig. 5 Flow-field for a two-dimensional calculation

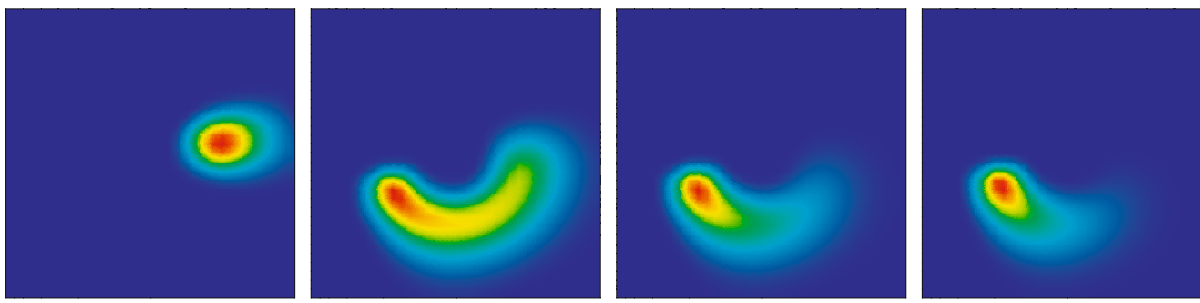

Fig. 4 The concentrations of the four components at the time-point $t=\frac{\pi}{4}$

We consider a model based on a rock overlying a salt dome. We suppose a waste case, so that a permanent source of radioactive contaminant groundwater flows from the bottom of the overlying rock, where the waste disposal is situated. We suppose that the contaminants move with the groundwater, which flows through the overlying rock. Based on our model we calculate the transport and the reaction of these contaminants coupled with decay chains. The simulation time is 10,000 [a], where "a" stands for "anno" or "year", and we calculate the concentration that flows to the top of the overlying rock. With the above data one should be able to conclude whether the waste disposal is sufficiently safe.

The next two cases are presented using data from GRS in Braunschweig (Germany); cf. [11, 22].

8.3 First waste case: two-dimensional model

We have a model domain of 6,000[m] $\times$ 150[m] with four different layers with different permeabilities, [22]. Groundwater flows through the domain from the right boundary to the left. The groundwater flows faster through the permeable layer than through the impermeable layers. Therefore, the groundwater flows from the right boundary to the middle half of the domain and through the permeable layer down to the bottom of the domain and is washed up in the left domain to the top. The groundwater flows in the left top part to the outflow at the left boundary. The flow field with the velocity is calculated with the program package **d^{3}f** and presented in Fig. 5.

In the middle-bottom of the domain the contaminants flow as a permanent source. With the stationary velocity field the contaminants are computed with the software package R^3T. The flow field transports the radioactive contaminants up to the top of the domain. The decay chain involves 26 components in the following manner:

$$\mathrm{Pu}-244 \rightarrow \mathrm{Pu}-240 \rightarrow \mathrm{U}-236 \rightarrow \mathrm{Th}-232 \rightarrow \mathrm{Ra}-228$$

$$\mathrm{Cm}-244 \rightarrow \mathrm{Pu}-240$$

$$\mathrm{U}-232$$

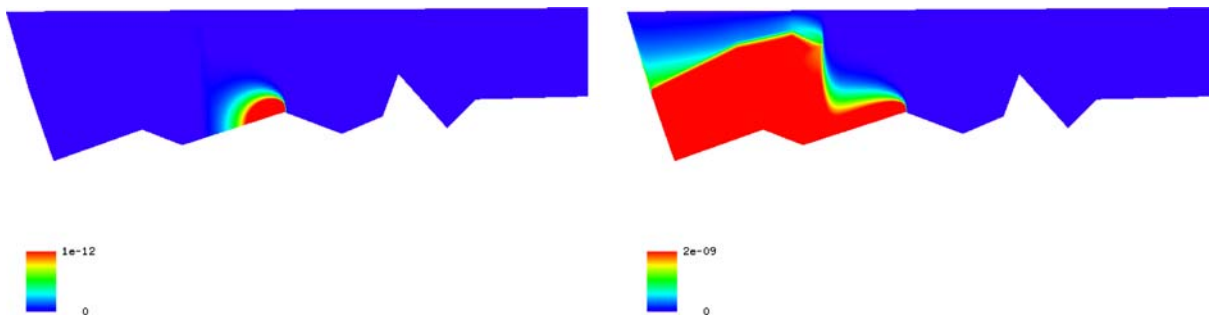

Fig. 6 Concentration of U-236 at the time point $t = 100$ [a] and $t = 10,000$ [a]

$$\mathrm{Pu}-241 \rightarrow \mathrm{Am}-241 \rightarrow \mathrm{Np}-237 \rightarrow \mathrm{U}-233 \rightarrow \mathrm{Th}-229$$
$$\mathrm{Cm}-246 \rightarrow \mathrm{Pu}-242 \rightarrow \mathrm{U}-238 \rightarrow \mathrm{U}-234 \rightarrow \mathrm{Th}-230 \rightarrow$$
$$\mathrm{Ra}-226 \rightarrow \mathrm{Pb}-210$$
$$\mathrm{Am}-242 \rightarrow \mathrm{Pu}-238 \rightarrow \mathrm{U}-234$$
$$\mathrm{Am}-243 \rightarrow \mathrm{Pu}-239 \rightarrow \mathrm{U}-235 \rightarrow \mathrm{Pa}-231 \rightarrow \mathrm{Ac}-227.$$

We introduce an important concentration in this decay chain. At the top of Fig. 6 the contaminant Uran-isotope U-236 after 100 [a] is presented. This isotope is less retarded and has a very long half-life period. As a result, the contaminant flows furthest and decays less. This effect is presented in the lower part of Fig. 6. The diffusion process has spread out the contaminant in the entire left part of the domain. Also, the impermeable layer is contaminated. After a time period of 10,000 [a] the contaminant has moved up to the top of the domain.

The calculations were carried out on uniform grids. The convergence of the used grids is confirmed by adaptive-grid calculations. The calculation confirmed the results for finer and smaller time steps; cf. Table 6. We start our calculations with explicit methods till the nature of the equations becomes diffusive. Then we apply implicit methods and are able to employ larger time steps. With this procedure we can satisfy the forced calculation time of one day at the most.

In the next section we describe a three-dimensional test case.

8.4 Second waste case: three-dimensional model

In this example we consider a three-dimensional model, because of the importance of three-dimensional effects in groundwater flow. We simulate about 10,000 [a] and focus on important contaminants that flow furthest with a high rate of concentration. We assume an anisotropy domain of 6,000 [m] $\times 2,000$ [m] $\times 1,500$ [m] with different permeable layers. We have calculated 26 components as presented in the two-dimensional case. The parameters for the diffusion and dispersion tensor are given as: $D = 1 \times 10^{-9}$ [m^2/s], $\alpha_L = 4{\cdot}0$ [m], $\alpha_T = 0{\cdot}4$ [m], $|v|_{\max} = 6 \times 10^{-6}$ [m/s], $\rho = 2 \times 10^3$, $D_L = \alpha_L |v|$ and $D_T = \alpha_T |v|$, where the longitudinal dispersion length is 10 times larger than the transversal dispersion. The source is situated at the point (4,250·0, 2,000·0, 1,040·0) and the concentrations are flowing at a constant rate. The under-

Table 6 Computation of the two-dimensional case

Processors	Refinement	Number of elements	Number of time-steps	Time for one time-step (s)	Total time (h)
30	Uniform	75,000	3,800	5	5·5
64	Adaptive	350,000	3,800	14	14·5

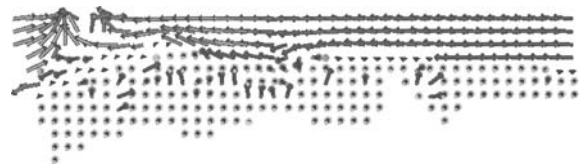

Fig. 7 Flow field for a three-dimensional calculation

lying velocity fields are calculated with d^3f and we added two sinks at the surface with the coordinates (2,000, 2,100, 2,073) and (2,500, 2,000, 2,073).

We simulate the transport and the reaction of the contaminants with our software-package R^3T. As a result, we can simulate the required pumping-rate of the sinks to bring out all the contaminant groundwater. We present the velocity field in Fig. 7. The groundwater flows form the right boundary to the middle of the domain. Due to the impermeable layers the groundwater flows downwards and filters up in the middle part of the domain. Further, the groundwater moves upwards to the sinks, seen in the left part of the domain. Because of the influence of the salt dome, the salt moves up with the groundwater and we get curls in the lower middle part of the domain. These parts are interesting for 3d calculations and, owing to these curls, the groundwater filters up. We now pay attention to the important component U-236. This component is less retarded and flows up to the earth's surface into the sinks. At the top of Fig. 8 we present the concentration in the initial concentration at time point $t = 100$ [a]. We present vertical cut-planes and in the next picture a cut-plane through the source term. In the bottom of Fig. 8 the concentration is presented at the end-time point $t = 10{,}000$ [a]. The concentration flows from the bottom up over the impermeable layer into the sinks at the top of the domain.

At the beginning of the calculation we use explicit discretization methods with respect of the convection-dominant case. After the initializing process the contaminants spread out with the diffusion process.

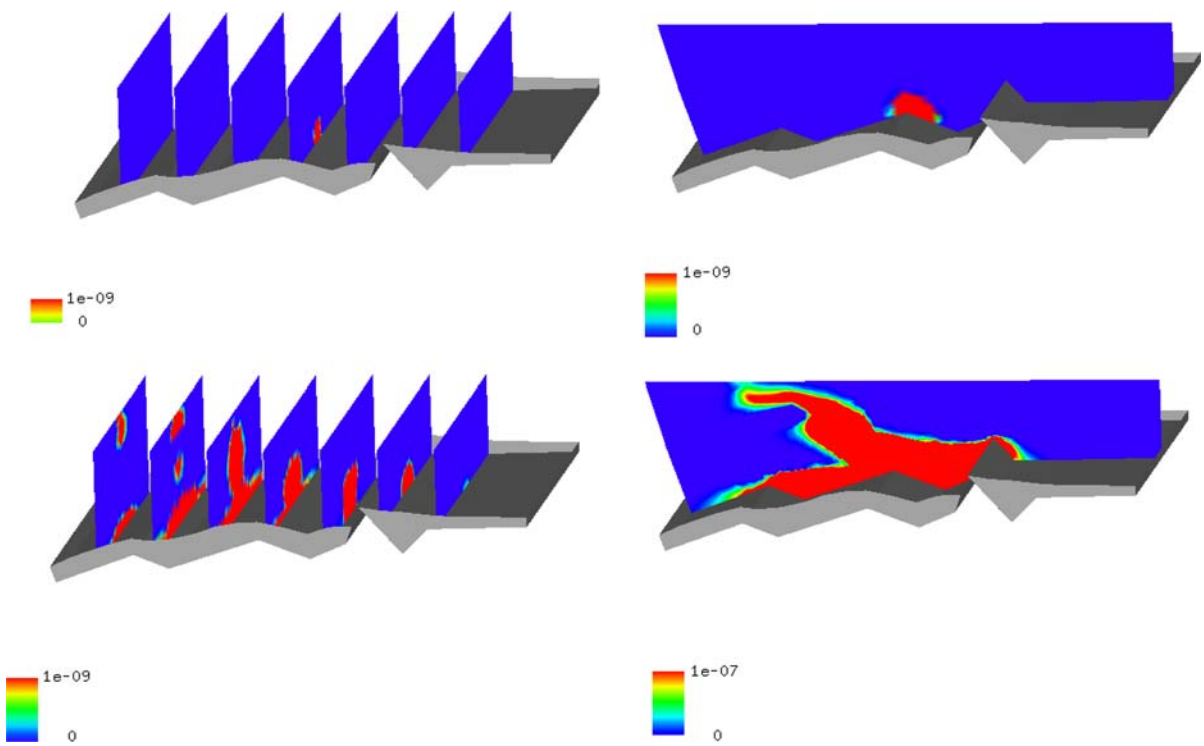

Fig. 8 Concentration of U-236 at the time-point $t = 100$ [a] and $t = 10{,}000$ [a]

Table 7 Three-dimensional calculations of a realistic potential damage event

Processors	Refinement	Number of elements	Number of time-steps	Time for one time step (s)	Total time (h)
16	Uniform	531,264	3·600	13·0	13·0
72	Adaptive	580,000	3·600	18·5	18·5

We use the implicit methods with larger time-steps and are able to calculate the forced time-period also in a higher-order discretization.

Table 7 lists the results of our computations. We begin with convergence results on uniform refined meshes. This confirms the results obtained with adaptive refined meshes and we get the same results as with smaller time-steps. The forced calculation time of one day is fulfilled.

9 Conclusions and discussions

In this paper we have proposed an efficient method for solving a complex system of convection–diffusion dispersion–reaction equations for a real-life model in waste disposal. We designed embedded discretization methods based on finite-volume methods and one-dimensional analytical solutions. For such an embedded method one can use larger time-steps because of the explicit time-discretization and analytical solutions. The diffusion–dispersion equations are discretized using a standard finite-volume method, based on central differences, which is stable for large time-steps. The combination of both equation parts with higher-order operator-splitting methods yields higher-order schemes in time and space. To accelerate the computation of the implicit discretized equations, we applied a fast multi-grid solver. The combination of discretization and solver methods allowed us to reach stable and consistent methods for complex convection–diffusion dispersion–reaction equations, with respect to the convection-dominant case. The verification of the proposed methods was done with special test examples, which were designed for the purpose of comparing the analytical solutions and the numerical solutions. We applied our methods to 2d and 3d models using the adaptive and parallel program toolbox developed in [38, 39]. The main advantages are fast computing times and accurate results based on explicit solution of the convection-dominant equation part. As a result, one can compare various waste scenarios which are simulated in less than 1 day and involve about 500,000 elements. Our results are used for predicting possible scenarios of such waste disposals placed in salt domes.

In the future we will focus us on the design of mixed discretization and solver methods which embed local analytical solutions of simpler equation parts and couple the simpler parts to iterative solver methods, for instance operator-splitting methods. The accuracy of a numerical solution can be improved by analytical parts which are independent of the underlying discretization scale. In a subsequent paper, we will consider such behavior and propose correct decoupling methods of multi-physics problems and new more accurate iterative methods.

Acknowledgements This work was funded by the German Federal Ministry of Economics and Technology (BMWi) under the contract number 02 E 9148 2.

References

1. Bear J (1972) Dynamics of fluids in porous media. Dover Publications, New York
2. Bear J, Bachmat Y (1991) Introduction to modeling of transport phenomena in porous media. Kluwer Academic Publishers Boston, Dordrecht
3. Hundsdorfer W, Verwer JG (2003) Numerical solution of time-dependent advection–diffusion–reaction equations, Springer Series in Computational Mathematics, vol 33. Springer Verlag, Berlin, Heidelberg
4. Pao CV (1992) Non linear parabolic and elliptic equation. Plenum Press, New York
5. Evans LC (1998) Partial differential equations. American Mathematical Society, Providence, Rhode Island

6. Kröner D (1997) Numerical schemes for conservation laws. John Wiley &Sons, Chichester
7. LeVeque RJ (2002) Finite volume methods for hyperbolic problems. Cambridge Texts in Applied Mathematics, Cambridge
8. Bey J (1998) Finite-volumen- und mehrgitter-verfahren für elliptische Randwertprobleme, Advances in numerical mathematics. B.G., Teubner Stuttgart, Leipzig
9. Frolkovič P (2002) Flux-based method of characteristics for contaminant transport in flowing groundwater. Comput Visuali Sci 5:73–83
10. Cockburn B (2003) Discontinuous Galerkin methods for convection-dominated problems. In: Barth TJ, Deconinck H (eds) Higher-order methods for computational physics. Lecture Notes in Computational Science and Engineering, vol 9. Springer Verlag, Berlin, Heidelberg, pp 69–225
11. Fein E, Kühle T, Noseck U (2001) Design of a 3d program-code for simulating the contaminat-transport. Fachliches Feinkonzept, GRS-Braunschweig, pp 1–30
12. Hundsdorfer W, Verwer JG (2003) Numerical solution of time-dependent advection–diffusion–reaction equations. Springer series in computational mathematics, vol 33. Berlin, Springer Verlag
13. Hundsdorfer WH (1996) Numerical solution od advection–diffusion–reaction equations. Technical Report NM-N9603, CWI
14. Butcher JC (2003) Ordinary differential equations. John Wiley &Sons, Chichester
15. Strang G (1968) On the construction and comparision of difference schemes. SIAM J Numer Anal 5:506–517
16. Verwer JG, Sportisse B (1998) A note on operator splitting in a stiff linear case. MAS-R9830, ISSN 1386-3703
17. Hackbusch W (1985) Multi-grid methods and applications. Springer-Verlag, Berlin, Heidelberg
18. Wittum G (1995) Multi-grid methods: a introduction. Bericht 1995-5 Institut für Computeranwendungen der Universität Stuttgart
19. Langtangen HP (2003) Computational partial differential equations. Text in computational science and engineering, vol 2. Springer-Verlag, Berlin, Heidelberg
20. Yoshida H (1990) Construction of higher order symplectic integrators. Phys Lett A 150:262–268
21. Geiser J (2004) Discretisation Methods for Systems of Convective-Diffusive-Dispersive-Reactive Equations and Applications. Doctor-Thesis, University of Heidelberg, Heidelberg
22. Fein E (2000) Test-example for a waste-disposal and parameters for a decay-chain. Private communications Braunschweig
23. Frolkovič P, Geiser J (2000) Numerical simulation of radionuclides transport in double porosity media with sorption. In: Handlovicova A, Komornikova M, Mikula K, Sevcovic D (eds) Algorithmy 2000, 15th conference of scientific computing-podbanske, Slovakia Vydavatel'sto STU, Bratislava
24. Zlatev Z (1995) Computer treatment of large air pollution models. Kluwer Academic Publishers, Dordrecht
25. Frolkovič P, Geiser J (2003) Discretization methods with discrete minimum and maximum property for convection dominated transport in porous media. In: Dimov I, Lirkov I, Margenov S, Zlatev Z (eds), Numerical methods and applications, 5th international conference, NMA 2002, Borovets, Bulgaria. Lecture notes in computer science, vol 2542. Berlin, Springer-Verlag, Heidelberg, pp 445–453
26. Frolkovič P, De Schepper H (2001) Numerical modelling of convection dominated transport coupled with density driven flow in porous media. Adv Water Resour 24:63–72
27. Higashi K, Pigford Th H (1980) Analytical models for migration of radionuclides in geologic sorbing media. J Nucl Sci Technol 17:700–709
28. Jury WA, Roth K (1990) Transfer functions and solute movement through soil. Bikhäuser Verlag, Basel, Boston, Berlin
29. Bateman H (1910) The solution of a system of differential equations occurring in the theory of radioactive transformations. Proc Cambridge Phil Soc 15:423–427
30. Sun Y, Petersen JN, Clement TP (1999) Analytical solutions for multiple species reactive transport in multiple dimensions. J Contaminant Hydrol 35:429–440
31. Scheidegger AE (1961) General theory of dispersion in porous media. J Geophys Res 66:32–73
32. Genuchten MTh (1985) Convective-dispersive transport of solutes involved in sequential first-order decay reactions. Comput Geosci 11:129–147
33. Eykolt GR (1999) Analytical solution for networks of irreversible first-order reactions. J Water Res 33:814–826
34. Davis B (1978) Integral transforms and their applications. Applied Mathematical Sciences vol 25. Springer Verlag, New York, Heidelberg
35. Yserentant H (1993) Old and new convergence proofs for multi-grid methods. Acta Numer 3:285–326
36. GRAPE (2001) GRAphics Programming Environment for mathematical problems, Version 5.4. Institut für Angewandte Mathematik, Universität Bonn und Institut für Angewandte Mathematik, Universität Freiburg, Freiburg, Bonn
37. Bastian P, Birken K, Eckstein K, Johannsen K, Lang S, Neuss N, Rentz-Reichert H (1997) UG - a flexible software toolbox for solving partial differential equations. Comput Visual Sci 1:27–40
38. Geiser J (2001) Numerical simulation of a model for transport and reaction of radionuclides. In: Margenov S, Wasniewski J, Yalamov PY (eds) Scientific computing, third international conference, LSSC 2001, Sozopol, Bulgaria. Lecture notes in computer science vol 2179. Springer-Verlag, Berlin, Heidelberg, pp 487–496
39. Geiser J (2004) R^3T: Radioactive-retardation–reaction-transport-program for the simulation of radioactive waste disposals. Proceedings: computing, communications and control technologies: CCCT 2004. The University of Texas at Austin, pp 1–8