Operator-splitting methods for transport equations with nonlinear reactions

Richard E. Ewing[a], Jürgen Geiser[b,*], Jiangguo Liu[a]

[a] Institute for Scientific Computation, Texas A&M University, College Station, TX 77843–3404, USA
[b] WIAS, Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstrasse 39, D-10117 Berlin, Germany

Abstract

In our paper we describe an operator-splitting method for a transport and nonlinear reaction equation with mixed discretization methods. We concentrate on adequate discretization methods by using the characteristic method for the convection–reaction equation with nonlinear reaction, and by using standard finite volume methods for the diffusion equation. We propose some new strategies based on efficient implementations for the operator-splitting method. Numerical results are included to demonstrate the performance of our methods.

Keywords: Adaptive methods; Disrupted and adaptive algorithms; Characteristic methods; Object and dynamic library concept

1. Introduction

Our motivation for studying nonlinear transport-reaction-equations comes from the background to predict scenarios for bio-models based on bio-remediation with nonlinear reaction parts. The understanding of the mechanisms for transport and chemical or biological reactions are important to simulate complex processes in porous media. The models are often complicated due to the coupling of species with nonlinear reactions and their geological structures which are based on heterogeneous media. Underlying this background for nonlinear and complex processes, first frame works in air pollutions are established, cf. [1,2].

The paper is organized as follows. A mathematical model of the transport-reaction-equations for chemical reaction or bio-remediation is introduced in section 2. The operator splitting is used as a solver-method to decouple the multi-physical and multi-dimensional equations to more simple-physical and one-dimensional equations as described in section 3. The implementation strategies to include the operator-splitting methods are based on software integration techniques which are presented in section 4. The numerical experiments and their description of our used methods are described in section 5. Finally the conclusions and an overview for our next works are discussed in section 6.

2. Convection-diffusion equations with nonlinear reactions

Our motivation for the study presented below is coming through the interests on foreseeing processes from radioactive contaminants [3] or bio-remediation [4] in porous media.

The mathematical equations are given as

$$u_t + \nabla \cdot (\mathbf{v}u - D\nabla u) = R(u) + f(x,t), x \in \Omega,\ t \in (0,T) \tag{1}$$

$$u|_{\partial\Omega} = 0,\ u(\mathbf{x},0) = u_0(\mathbf{x}) \tag{2}$$

The solution, denoted as density or concentration of the species $u = u(\mathbf{x}, t)$, is considered in $\Omega \times (0,T) \subset \mathbb{R}^d \times \mathbb{R}$, the space-dimension is denoted by d. The right-hand side $f(\mathbf{x}, t)$ is given in $\Omega \times (0, T)$, while D is the Scheidegger diffusion-dispersion tensor and $\mathbf{v}$ is the divergence-free velocity. We concentrate on the following reaction-terms $R(u)$

$$R(u) = -au,\ \text{radioactive decay},\ a \geq 0 \tag{3}$$

$$R(u) = u^p,\ \text{chemical-reaction},\ p > 0 \tag{4}$$

* Corresponding author. Tel.: +49 30 20 372 541; Fax: +49 30 20 44975; E-mail: geiser@wias-berlin.de

Computational Fluid and Solid Mechanics 2005
K.J. Bathe (Editor)

$$R(u) = au - bu^2, \text{ logistic model, } a,\ b \geq 0 \tag{5}$$

$$R(u) = \frac{au}{u+b}, \text{ bio-remediation, } a, b \geq 0 \tag{6}$$

For more complicated cases of biodegradation we refer to the literature [4,5].

In the following we describe the Operator-Splitting method as a basic splitting method to solve our equations with standard discretization methods.

3. Operator splitting methods

The operator-splitting methods are developed to solve complex models in geophysics and environmental physics; they are described and applied in [1] and [6]. The basic ideas are the decoupling to simpler equations and using higher-order discretization methods to increase the accuracy of each solution. For this aim one could write the transport Eq. (1) as

$$u_t = A(u) + B(u) \tag{7}$$

with

$$A(u) = -\mathbf{v} \cdot \nabla u + R(u) \tag{8}$$

$$B(u) = \nabla \cdot (\mathbf{D}\nabla u) + f(\mathbf{x}, t) \tag{9}$$

For the time-steps we define $\Delta t := T/N$, where N is a positive integer and T is the positive end time-point. Further, the time-points are given as $t_n = n\Delta t$ with $n = 0, 1, \ldots, N$, and are uniform partitions of the time period $[0, T]$.

The operator-splitting method for the operators A, B can be applied as a first-order method and is given as

$$u_t^{(1)} = A(u^{(1)}) \text{ with } t^n \leq t \leq t^{n+1} \text{ and } u^{(1)}(\mathbf{x}, t^n) = u(\mathbf{x}, t^n) \tag{10}$$

$$u_t^{(2)} = B(u^{(2)}) \text{ with } t^n \leq t \leq t^{n+1} \text{ and } u^{(2)}(\mathbf{x}, t^n) = u^{(1)}(\mathbf{x}, t^{n+1}) \tag{11}$$

and the splitting-error can be written as

$$\rho_n = \frac{1}{2}\Delta t((\frac{\partial A}{\partial u}B)(u) - (\frac{\partial B}{\partial u}A)(u)) + O(\Delta t^2) \tag{12}$$

We obtain a splitting-error $O(\Delta t)$ for not commuting operators A, B otherwise we get an exact method. For our nonlinear operators A and B which are given in equation (8) and (9) we apply the splitting-error and derive the following result

$$\begin{aligned}&((\frac{\partial A}{\partial u}B)(u) - (\frac{\partial B}{\partial u}A)(u))\\ &\quad = ((-\mathbf{v}\cdot\nabla + \frac{\partial R}{\partial u})(\nabla\cdot\mathbf{D}\nabla))(u)\\ &\qquad -((\nabla\cdot\mathbf{D}\nabla)(-\mathbf{v}\cdot\nabla + R))(u) \neq 0\end{aligned} \tag{13}$$

based on the nonlinear functions $R(u)$. We get a splitting-error of $O(\Delta t)$ for the nontrivial non-commuting case, cf. [1].

For the solution method of the coupled Eq. (1), we split into two equations for the small time period $[t_n, t_{n+1}]$ given as

$$\frac{du}{dt} = u_t + \mathbf{v}\cdot\nabla u = R(u) \tag{14}$$

$$u_t = \nabla\cdot(\mathbf{D}\nabla u) + f(\mathbf{x}, t) \tag{15}$$

$$u|_{\partial\Omega} = 0,\ u(\mathbf{x}, 0) = u_0(\mathbf{x}) \tag{16}$$

The convection-reaction equation is solved along the characteristics and becomes a nonlinear ODE

$$\begin{cases} \frac{du^{(1)}}{dt} = R(u^{(1)}), \quad t \in (t_n^*, t_{n+1}) \\ u^{(1)}(\mathbf{x}^*, t_n^*) = u^{(2)}(\mathbf{x}^*, t_n^*) \end{cases} \tag{17}$$

where $(\mathbf{x}, t)$ is the exact backtracking of $(\mathbf{x}, t)$ and $u^{(2)}$ is the solution for the parabolic problem. For $n = 0$ or $\mathbf{x}^* \in \partial\Omega$, we replace $u^{(2)}$ $(\mathbf{x}^*, t^*_n)$ by $u_0(\mathbf{x}^*)$ with the boundary condition given in Eq. (16). Our Eq. (17) is numerically solved by explicit Euler or Runge–Kutta methods.

The second part is an initial boundary value problem for a typical parabolic equation

$$\begin{cases} u_t^{(2)} = \nabla\cdot(\mathbf{D}\nabla u^{(2)}) + f(\mathbf{x}, t), \quad \mathbf{x} \in \Omega,\ t \in (t_n, t_{n+1}) \\ u^{(2)}|_{\partial\Omega} = 0 \\ u^{(2)}(\mathbf{x}, t_n) = u^{(1)}(\mathbf{x}, t_{n+1}) \end{cases} \tag{18}$$

The spatial discretization of Eq. (18) is done by standard finite difference, element, or volume methods. Further, we use explicit time-discretization methods of Eq. (18), e.g. forward Euler-method and restrict our time-step Δt on the stability condition $|\mathbf{D}|\Delta t/h^2 \leq 1/2$, where $|\mathbf{D}| << 1$. Because of using implicit time-discretization, e.g. backward Euler-method, the time-restriction is not necessary.

In the next section we describe the implementation of our methods.

4. Implementation of operator splitting methods

In this section, we discuss our implementation strategies for the method based on various software integration techniques such as dynamic link library (DLL) and component object model (COM).

4.1. DLL and COM

The dynamic link library (DLL) is a collection of functions and data that are available for using one or more applications running on the same computer

system. The executable code modules in DLL are loaded through a demand and linked at run time, and are unloaded when they are no longer needed. When a DLL is loaded, it is mapped into the address space of the calling process. The component object model (COM) is the Microsoft standard for interoperability of binary objects. The COM objects are implemented either as Dynamic Link Libraries (DLLs) or as executables (EXEs). For details about DLL and COM, readers are referred to [7]. Now both DLL and COM are supported by many high-level programming languages such as C++, Fortran, and even Matlab.

4.2. Tracking characteristics

Characteristic tracking is an important part of this splitting method. We define an approximate characteristic as a chain of line segments, which are adaptively formulated according to the magnitude of the velocity field. To be more precise, the number of points on each (approximate) characteristic varies. Both forward and backward tracking of characteristics could be performed in practice with respect to the accuracy. Therefore, a doubly linked list is an efficient data structure for approximate characteristics. In this part we have to develop our own code for solving the Eqs. (14) with adaptive tracking of characteristics.

4.3. Solving nonlinear ODEs along characteristics

To solve the nonlinear ODEs we actually solve along the approximate characteristics. We apply Euler and Runge–Kutta methods for our nonlinear ODEs with a given accuracy. For the flexibility we use different time-steps on different characteristics to get more exact solutions.

4.4. Solving parabolic problem via FEM

For the numerical solution of an initial boundary value problem given as a linear parabolic equation, we use a finite difference/element/volume method that is now more or less a conventional task. In this direction there are plenty of commercial or free software codes and we concentrate on the most flexible codes. We choose libMesh, developed at The University of Texas at Austin [8], and OFELI (Object Finite Element Library) developed in France [9] provide free C++ source code that is easy to implement in our programs. Further the PDE toolbox in Matlab is also a basic finite element package and does not require users to have C++ programming experience.

5. Numerical experiments

5.1. Example 1: Linear reaction

To examine our method, we first consider a 2-dimensional problem with a linear reaction. For this experiment we can find the exact solution so that we can compare the numerical and exact results. In particular, we have a rotating velocity $\mathbf{v} = (-4y, 4x)$, a constant scalar diffusion $D > 0$, a linear reaction $R(u) = Ku$ with K being a constant, and a null source/sink, i.e. $f \equiv 0$. W assume that the substances are initially normally distributed, i.e. the initial condition is specified as a Gaussian hill

$$u_0(x, y) = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \tag{19}$$

For this special case our exact solution is given by

$$u(x, y, t) = \frac{2\sigma^2}{2\sigma^2 + 4Dt}\exp\left(Kt - \frac{(x^* - x_c)^2 + (y^* - y_c)^2}{2\sigma^2 + 4Dt}\right) \tag{20}$$

where $(x^*, y^*, 0)$ is the backtracking initial point for the characteristics from (x, y, t).

We simplify by using a uniform triangular mesh. For the characteristic tracking we use the second order Runge–Kutta (or Heun) method. The finite element solver for the parabolic part, used as a DLL, is derived from a modified source code in OFELI.

In our numerical experiments, we choose $T = \pi/2$, $\Omega = [-1, 1] \times [-1, 1]$, $D = 10^{-4}$, $K = 0.1$, $(x_c, y_c) = (-0.5, -0.5)$, and $\sigma^2 = 0.01$. For the parabolic solver, we use 20 micro steps for each global time step $[t_n, t_{n+1}]$. Due to this solver, we set the maximal number of time-steps in characteristic tracking also to 20. Table 1 lists some results for the numerical solution at the final time. We still obtain very good numerical solutions, even though the use of a flexible large global time steps.

5.2. Example 2: Nonlinear reaction

The second example is a simplified model for single-species biodegradation: $R(u) = au|(u + b)$. We consider

Table 1
Numerical results of example 1 with $\Delta t = \pi/8$

Mesh size h	L^∞-error	L^1-error	L^2-error
1/20	1.266×10^{-2}	1.247×10^{-4}	3.138×10^{-4}
1/40	1.031×10^{-2}	5.061×10^{-5}	2.085×10^{-4}
1/50	9.984×10^{-3}	4.153×10^{-5}	1.923×10^{-4}
1/60	9.796×10^{-3}	3.613×10^{-5}	1.825×10^{-4}

Table 2
Numerical results of example 2

Δt	h	U_{min}	U_{max}	Δt	h	U_{min}	U_{max}
0.25	1/20	0.0	1.5159	0.125	1/40	0.0	1.5251
0.25	1/40	0.0	1.5176	0.10	1/20	0.0	1.5248
0.25	1/60	0.0	1.5179	0.10	1/50	0.0	1.5268

a two-dimensional problem with a constant velocity filed (V_1, V_2), a scalar diffusion $D > 0$, and we have no sources in the domain. The initial condition is a Gaussian hill, as introduced in Eq. (19).

For numerical runs we chose $T = 1$, $\Omega = [-1, 1] \times [-1, 1]$, $(V_1, V_2) = (1, 1)$, $D = 10^{-4}$, $a = b = 1$, $(x_c, y_c) = (-0.5, -0.5)$, and $\sigma^2 = 0.01$. In Table 2 we present some results of the numerical solution at the final time.

For this problem an exact solution is not known. But from the results presented in Table 2 we can observe that the operator-splitting method is stable, and keeps positivity of the solution.

6. Conclusions

In this paper we propose an operator-splitting method for transport equations with nonlinear reactions. The error-analysis for a first-order method is presented. To implement the improved solver and discretization methods, we incorporate some existing commercial and free software components into our own program. By integrating functionalities of existing software components the time period of software development is significantly shortened. The numerical results correspond to our theoretical results and first nonlinear computations are presented. In our future work we will focus on error analysis for the coupled methods with respect to the physical behavior of the equations. The use of such flexible splitting methods allows us to simulate complex physical models.

References

[1] Lanser D, Verwer JG. Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modeling. J Comput Appl Math 1999;111:201–216.
[2] Verwer JG, Sportisse B. A note on operator splitting in a stiff linear case. MAS-R9830, 1998.
[3] Geiser J. R^3T: Radioactive-Retardation-Reaction-Transport-Program for the Simulation of radioactive waste disposals. Technical report, Institute for Scientific Computation, Texas A&M University, College Station, 2004.
[4] Wang H, Ewing RE, Celia MA. Eulerian–Lagrangian localized adjoint methods for reactive transport with biodegradation. Numer Meth Partial Diff Eqn 1995;11:229–254.
[5] Celia MA, Kindred JS, Herrera I. Contaminant transport and biodegradation 1. A numerical model for reactive transport in porous media. Water Resources Research 1989; 25:1141–1148.
[6] Karlsen KH, Risebro NH. An operator splitting method for nonlinear convection-diffusion equations. Numer Math 1997;77:365–382.
[7] Component object model: http://www.microsoft.com/com/
[8] libMesh: http://libmesh.sourceforge.net/
[9] OFELI: http://ofeli.sourceforge.net/