Applied Mathematics and Computation 332 (2018) 76–89

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Efficient boundary corrected Strang splitting

Lukas Einkemmer[a], Martina Moccaldi[b], Alexander Ostermann[a,*]

[a] Department of Mathematics, University of Innsbruck, Innsbruck, Austria
[b] Department of Mathematics, University of Salerno, Fisciano (SA), Italy

ARTICLE INFO

Keywords:
Strang splitting
Dirichlet boundary conditions
Neumann boundary conditions
Diffusion-reaction equations
Overcoming order reduction

ABSTRACT

Strang splitting is a well established tool for the numerical integration of evolution equations. It allows the application of tailored integrators for different parts of the vector field. However, it is also prone to order reduction in the case of non-trivial boundary conditions. This order reduction can be remedied by correcting the boundary values of the intermediate splitting step. In this paper, three different approaches for constructing such a correction in the case of inhomogeneous Dirichlet, Neumann, and mixed boundary conditions are presented. Numerical examples that illustrate the effectiveness and benefits of these corrections are included.

1. Introduction

In recent years, splitting schemes have been widely used to integrate evolution equations (see, for instance, [3,8,12]). The main advantage of the splitting approach, compared to standard integrators, is the fact that it allows the separate treatment of different terms of the vector field. This usually reduces the computational effort required to solve the problem. Moreover, this strategy facilitates in many cases a parallel implementation and may lead to numerical methods having better geometric properties. For all these reasons, splitting methods are highly suitable for the numerical treatment of models describing complex phenomena. For instance, in turbulent combustion simulations with finite-rate kinetics, hundreds of differential equations (one for each chemical species) have to be integrated in addition to the Navier–Stokes equations for momentum and energy. Moreover, such problems are typically stiff, so they require implicit solvers. In these examples, splitting separately treats the chemical kinetics and the convection-diffusion terms. This allows one to use appropriate numerical methods for each part and can thus considerably reduce the computational cost [14].

This paper is focused on diffusion-reaction systems. Such problems are efficiently integrated by splitting schemes [4,9]. In particular, we will consider equations where the diffusion is modelled by a linear elliptic differential operator and the reaction by a nonlinear function. The splitting approach takes into account the different nature of the components and replaces the nonlinear system of differential equations with a linear system and a set of ordinary differential equations, making the implementation easier and reducing the computational burden. Moreover, splitting methods preserve positivity and invariant sets provided that the integrator of each flow has the corresponding property [11].

When periodic or homogeneous Dirichlet boundary conditions are employed, the well-known Strang splitting scheme is second-order accurate, in general. However, it has been proved in the literature that the order of Strang splitting is reduced in the case of non trivial boundary conditions, e.g. inhomogeneous Dirichlet, Neumann, Robin boundary conditions or mixed

* Corresponding author.
E-mail addresses: lukas.einkemmer@uibk.ac.at (L. Einkemmer), mmoccaldi@unisa.it (M. Moccaldi), alexander.ostermann@uibk.ac.at (A. Ostermann).

https://doi.org/10.1016/j.amc.2018.03.006
0096-3003/

ones (see, for instance, [5,6,12]). One possibility to overcome this order reduction is to introduce a smooth correction function in splitting schemes such that the new reaction flow is compatible with the prescribed boundary conditions [5,6]. For time-invariant Dirichlet boundary conditions, this correction has to be computed only once at the beginning of the simulation. However, for time-dependent Dirichlet, Neumann, or Robin boundary conditions, the correction is time-dependent. In these cases, the correction function has to be computed at each time step, which could noticeably increase the computational cost. Hence, an open problem is finding an efficient procedure to construct this function in order to preserve the aforementioned computational advantages of the splitting scheme.

In this work, we present and compare different strategies to construct the correction function. First of all, we propose to select the correction function as the solution of an elliptic problem endowed with appropriate boundary conditions [5,6]. A similar elliptic problem has to be solved by the splitting method in each time step. Therefore, the computational overhead of this additional solution is moderate. Nevertheless, we investigate here other approaches which, depending on the situation, are less expensive. E.g., an effective procedure consists in computing the correction function with the help of the actual numerical solution. For Dirichlet boundary conditions, the correction is computed at each time step as $f(u_n)$, where f is the vector field of the reaction and u_n is the numerical solution at the beginning of the current time step. This is a good option for problems where the application of f is cheap. However, there are situations in which additional evaluations of the vector field f significantly increase the computational effort, as it is the case in turbulent combustion simulations [14]. Therefore, we present two further techniques which are more convenient for problems having expensive reactions. For Dirichlet boundary conditions, we employ a widely used low-pass filter for noise reduction in linear image processing [1,13], the moving average, which we extend with multiple grid levels to further increase smoothness. In the case of Neumann boundary conditions, we use a crude approximation of the elliptic problem on a hierarchy of grids, which results in a multigrid-like algorithm.

The outline of the paper is as follows. In Section 2 we present the splitting approach, explain the related problem of order reduction and the use of a correction function to avoid it. Sections 3 and 4 are devoted to the description of efficient algorithms to construct the correction function for Dirichlet and Neumann boundary conditions, respectively. Moreover, we report some numerical tests that prove the effectiveness of the proposed approaches. Finally, we discuss some conclusions in Section 5.

2. Model problem

In this paper, we deal with the numerical integration of diffusion-reaction systems modelled by the following initial-boundary value problem

$$\partial_t u = Du + f(u), \tag{2.1a}$$

$$Bu|_{\partial\Omega} = b, \tag{2.1b}$$

$$u(0) = u_0, \tag{2.1c}$$

where $u : [0, T] \times \Omega \to \mathbb{R}$. Here, $\Omega \subset \mathbb{R}^d$ is a bounded domain with sufficiently smooth boundary $\partial\Omega$, the reaction term f is a real smooth function, typically nonlinear. Moreover, D is a second-order elliptic differential operator (e.g. the Laplacian), the boundary data b: $[0, T] \times \partial\Omega$ may be time-dependent and

$$B = \beta(x)\partial_n + \alpha(x) \tag{2.2}$$

is a first-order differential operator with sufficiently smooth coefficients. The functions b and u_0 are also assumed to be sufficiently smooth. According to the values of the coefficients in B, different boundary conditions can be modelled. For example, if $\beta(x) = 0$ and $\alpha(x) \geq c > 0$, then (2.1b) corresponds to Dirichlet boundary conditions, whereas $\alpha(x) = 0$ and $\beta(x) \geq c > 0$ give Neumann boundary conditions.

For the numerical solution of (2.1) we split the system into the linear diffusion equation

$$\partial_t v = Dv, \quad Bv|_{\partial\Omega} = b \tag{2.3a}$$

and the nonlinear reaction equation

$$\partial_t w = f(w). \tag{2.3b}$$

Let u_n be the numerical approximation to the exact solution u of (2.1) at time $t = t_n$. To step from t_n to $t_{n+1} = t_n + \tau$, where τ is the step size, we use the well-known Strang splitting approach

$$u_{n+1} = \mathcal{S}_\tau u_n = \varphi_{\tau/2}(\psi_\tau(\varphi_{\tau/2}(u_n))), \tag{2.4}$$

where φ_τ and ψ_τ are the exact flows of the first (2.3a) and second (2.3b) sub-problem, respectively. In other words, Strang splitting for the integration of problem (2.1) consists in computing the solution $v(\frac{\tau}{2})$ of (2.3a) with initial value $v(0) =$