Iterative Operator Splitting Methods with embedded Multi-grid methods

Jürgen Geiser [a,*],

[a] *Department of Mathematics, Humboldt-Universität zu Berlin, Unter den Linden 6, D-10099 Berlin, Germany*

Abstract

In this article a new approach is considered for implementing iterative operator splitting methods for differential equations. The underlying idea is to embed fast Multi-grid methods to accelerate the iterative splitting schemes. The main problem are fast iterative solvers for multi-scale physics, while different scales, we have to apply multi-grid methods to obtain the optimal scale, which can be solved by an iterative splitting schemes. Here we discuss the embedding of such spatial- and time-scale methods, e.g. Multi-grid (spatial) and BDF (time) methods, to taken into account the different scales.

Key words: numerical analysis, operator splitting method, initial value problems, iterative solver method, stability analysis, convection-diffusion-reaction equation, multi-grid methods
1991 MSC: [2008] 35K15 35K57 47F05 65M60 65N30.

1. Introduction

We are motivated on solving multi-scale problems that are given for example in transport problem in porous media.

In the last years, the interest in numerical simulations with multi-scale problems, that can be used to model different physical behavior, e.g. potential damage events, has significantly increased, while also the methodology is delicate to adapt for such problems, see [39] and [40].

We concentrate on simplified models and taken into the ideas to adapt the underlying splitting schemes, see [15]. While the splitting schemes worked optimal for single scale

* Corresponding author.
Email address: geiser@mathematik.hu-berlin.de (Jürgen Geiser).

problems, we have to embed multi-grid ideas or multi-step ideas to extend our schemes to multi-scale solvers. We deal with iterative splitting methods and decomposing such scale problems and accelerate the convergence rates with multi-scale methods, such that we can overcome to such scaling problems.

Moreover the embedding part is important, while we dealing with multi-grid methods, we have to extend the underlying spitting analysis and discuss the algorithmic ideas.

The novelty is to embed cheaply the multi-grid and multi-step methods and apply only cheap iterative schemes to achieve higher order results.

In general, we discuss an elegant way of embedding recently survey on methods for multi-scale problems with respect to iterative splitting schemes.

In the following, we describe our model problem. The model equation for the multi-scale equations are given as coupled partial differential equations.

We concentrate on a far-field model for a plasma reactor, see [16] and [30] and assume a continuum flow, and that the transport equations can be treated with a convection-diffusion-reaction equation, due to a constant velocity field, see:

$$\frac{\partial u}{\partial t} + \nabla \cdot F u = 0, \text{ in } \Omega \times [0, t] \tag{1}$$

$$F = \mathbf{v} - D\nabla,$$

$$c(x, t) = c_0(x), \text{ on } \Omega, \tag{2}$$

$$\frac{\partial c(x, t)}{\partial n} = 0, \text{ on } \partial\Omega \times [0, t], \tag{3}$$

where c is the particle density of the ionized species. F the flux of the species. $\mathbf{v}$ is the flux velocity through the chamber and porous substrate which is influenced by the electric field. D is the diffusion matrix. The initial value is given as c_0 and we assume a Neumann boundary condition.

The aim of this paper is to present a novel iterative splitting method, that embed multi-scale methods

The paper is outlined as follows: In Section 2, we present the underlying splitting methods. The stability analysis of the embedded Multi-grid schemes with the iterative schemes are given in Section 3. The assembling and algorithms to the embedded multi-grid method is given in Section 4. Numerical verifications are given in Section 5. In Section 6, we briefly summarize our results.

2. Iterative splitting methods

The iterative splitting methods are developed in the early 90'ies, see [28], [41].

The idea is to decouple the equations in two or more equations to save computational times, while letting equation part unchanged with previous results. We obtain inhomogeneous partial differential equations and solve them with appropriate methods.

We consider the following the differential equation problem, while the operators A and B are given spatial discretized operators:

$$\frac{\partial u}{\partial t} = Au + Bu, \tag{4}$$

where the initial conditions are $u^n = u(t^n)$. The operators A and B are spatially discretized operators, e.g. they correspond in space to the discretized convection and diffusion operators (matrices). Hence, they can be considered as bounded operators with a sufficient large spatial step $\Delta x > 0$.

In the following we discuss the different methods.

2.1. *The sequential iterative splitting method*

The classical sequential operator splitting, known as Lie-splitting or Strang-Marchuk splitting methods, have several drawbacks besides their benefits, see [38], [31] and [**?**]. For instance, for non-commuting operators there might be a very large constant in the splitting error which requires the use of an unrealistically small time step. Also, splitting the original problem into the different subproblems with one operator, i.e. neglecting the other components, is physically questionable.

In order to avoid these problems, one can use the iterative operator splitting method on an interval $[0, T]$. This algorithm is based on the iteration with fixed splitting discretization step-size τ. On every time interval $[t^n, t^{n+1}]$ the method solves the following subproblems consecutively for $i = 0, 2, \ldots 2m$.

$$\begin{aligned}
& \frac{\partial u_i(x,t)}{\partial t} = A u_i(x,t) \; + \; B u_{i-1}(x,t), \text{ with } \; u_i(t^n) = u^n = v_0 && (5) \\
& u_0(x,t^n) = u^n \; , \; u_{-1} = 0, \\
& \text{and } u_i(x,t) = u_{i-1}(x,t) = v_1 \; , \text{ on } \partial\Omega \times (0,T) \; , \\
& \frac{\partial u_{i+1}(x,t)}{\partial t} = \; A u_i(x,t) \; + \; B u_{i+1}(x,t), && (6) \\
& \text{with } \; u_{i+1}(x,t^n) = u^n = v_0 \; , \\
& \text{and } u_i(x,t) = u_{i-1}(x,t) = v_1 \; , \text{ on } \partial\Omega \times (0,T) \; ,
\end{aligned}$$

where u^n is the known split approximation at the time level $t = t^n$ (see [9]). This algorithm constitutes an iterative method which involves in each step both operators A and B. Hence, there is no real separation of the different physical processes in these equations.

Remark 1 *For the presented iterative splitting method, we have a serial algorithm and we can not use the method parallel in this version. Because of the efficiency, we modify the method with respect to the parallelization.*

2.2. *The parallel iterative splitting method*

Taken into account to parallelize the standard iterative splitting method, see Section 2.1, we propose a decoupled version.

The iterative scheme can be done in a Jacobian form, so that the two parts of the algorithm can be computed independently.

In the next iteration steps, we apply the previous solutions and improve them in the next iteration steps.

We apply the parallel iterative splitting method for $i = 0, 2, \ldots 2m$, and we have :

$$\frac{\partial u_i(x,t)}{\partial t} = A u_i(x,t) \ + \ B u_{i-1}(x,t), \text{ with } \ u_i(t^n) = u^n = v_0 \tag{7}$$
$$u_0(x,t^n) = u^n \ , \ u_{-1} = 0,$$
$$\text{and } u_i(x,t) = u_{i-1}(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$
$$\frac{\partial u_{i+1}(x,t)}{\partial t} = \ A u_{i-2}(x,t) \ + \ B u_{i+1}(x,t), \tag{8}$$
$$\text{with } \ u_{i+1}(x,t^n) = u^n = v_0 \ , \ u_{-2} = 0,$$
$$\text{and } u_i(x,t) = u_{i-1}(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$

where u^n is the known split approximation at the time level $t = t^n$. This algorithm constitutes an iterative method which involves in each step both operators A and B. Hence, there is no real separation of the different physical processes in these equations.

The first iterative steps are given as :

$$\frac{\partial u_0(x,t)}{\partial t} = A u_0(x,t), \text{ with } \ u_0(t^n) = v_0,$$
$$\text{and } u_0(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$
$$\frac{\partial u_1(x,t)}{\partial t} = \ B u_1(x,t), \tag{9}$$
$$\text{with } \ u_1(x,t^n) = u^n = v_0,$$
$$\text{and } u_1(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$

and

$$\frac{\partial u_2(x,t)}{\partial t} = A u_2(x,t) + B u_1(x,t), \text{ with } \ u_2(t^n) = v_0,$$
$$\text{and } u_2(x,t) = u_1(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$
$$\frac{\partial u_3(x,t)}{\partial t} = A u_0(x,t) + B u_3(x,t), \tag{10}$$
$$\text{with } \ u_3(x,t^n) = u^n = v_0,$$
$$\text{and } u_3(x,t) = u_0(x,t) = v_1 \ , \text{ on } \partial\Omega \times (0,T) \ ,$$

Remark 2 *The effect of the parallelization is to obtain accuracy of the sequential iterative splitting with one more iterative step.*

2.3. *The Multi-grid Algorithm*

For describing the multi-grid method we first initiate the two-grid method. Afterwards we extend it recursively to the multi-grid method. The smoother on grid level l is denoted by S_l. The two-grid method is defined as follows:

$$M_l^{ZG} \ := \ S_l^{\nu_2} \, M_l^{ZGG} \, S_l^{\nu_1}. \tag{11}$$

where ν_1 denotes the pre-smoothing steps and ν_2 the post-smoothing steps. The correction M_l^{ZGG} on the coarse grid is defined by:

$$M_l^{ZGG} \ := \ \mathbb{1} - p \, A_{l-1}^{-1} \, r \, A_l. \tag{12}$$

Linear multi-grid cycle

$MG(x_l, b_l, l)$
```
{
  if (l == 0)
    {
  x_0 = A_0^{-1} b_0 ;    exact solving on coarse grid.
  }
  else
  {
      x_l = S^{ν1}(x_l, b_l) ;    ν1 pre-smoothing steps.
        b_{l-1} = r b_l ;    defect restricted on next coarser grid.
      x_{l_1} = 0 ;
      for (i = 0; i < γ; i++)
      {
          c_{l-1} = 0 ;
            MG(c_{l-1}, b_{l-1}, l-1) ;    γ-fold recursive invoking of
                                              correction of coarse grid.
          x_{l-1} = x_{l-1} + c_{l-1} ;
          b_{l-1} = b_{l-1} - A_{l-1} c_{l-1} ;
      }
      c_l = p x_{l-1} ;    interpolation of correction of the
                              next coarse grid.
      x_l = x_l + c_l ;
      b_l = b_l - A_l c_l ;
      x_l = S^{ν2}(x_l, b_l) ;    ν2 post-smoothing steps.
  }
}
```

The junction to the multi-grid method is done in such a way, that matrix A_{l-1} of the coarse grid in Equation (12) is not inverted exactly, but the two-grid method is invoked γ-times to solve the equation systems at grid level $l-1$. The equation system is only solved on the coarsest grid.

The multi-grid method is defined as:

$$M_0^{MG} := 0, \tag{13}$$

$$M_1^{MG} := M_1^{ZG}, \tag{14}$$

$$M_l^{MG} := M_l^{ZG} + \ S_l^{\nu_2}\ p\ (M_{l-1}^{MG})^{\gamma}\ A_{l-1}^{-1}\ r\ A_l\ S_l^{\nu_1}, \tag{15}$$

where ν_1 denotes the pre-smoothing steps and ν_2 the post-smoothing steps. The corrections M_l^{MGG} of the coarse grid are defines as:

$$M_l^{MGG} \ := \ \mathbb{1} - p\ (I - (M_{l-1}^{MG})^{\gamma})\ A_{l-1}^{-1}\ r\ A_l. \tag{16}$$

This approach is called multi-grid method. For the choice of $\gamma = 1$ one speaks of a V-cycle, for $\gamma = 2$ it is a W-cycle.

The multi-grid algorithm is given by: The refinement is done using the strategy of [18]. By assuming a linear effort for the smoothers as well as for the grid-transfer operators, one obtains a linear effort for Algorithms 2.3, if $\gamma \leq 3$, confer [18]. The proofs of convergence for the W-cycle were don in [17]. The topic of convergence will not be further discussed, for an overview we refer to [44].

Subsequently, the multi-grid cycles are illustrated in Figure 1. For a further consolida-

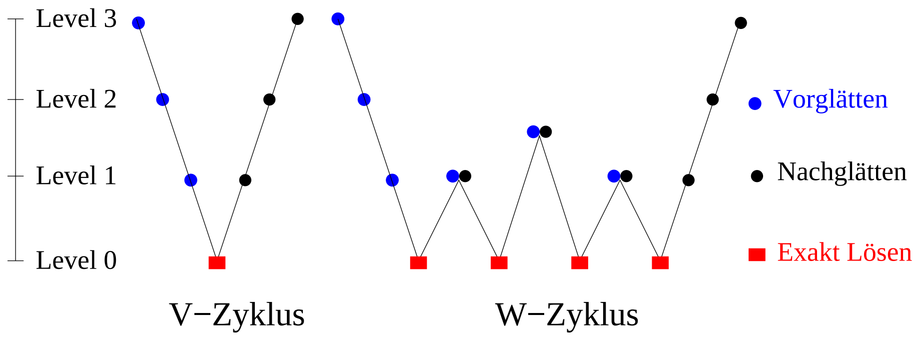

Fig. 1. Multi-grid cycles: V- and W-cycle.

tion of the topic of multi-grid methods we refer to [17], [2] and [18].

2.4. *Iterative splitting method with embedded Multi-grid method*

The following algorithm is based on embedding the multi-grid method to the operator splitting method. The iteration with fixed splitting discretization step-size τ. On the time interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for $i = 0, 2, \dots 2m$. (cf. [24] and [9].)

$$\frac{\partial c_i(t)}{\partial t} = A c_i(t) \; + \; P^{l_A - l_B} B c_{i-1}(t), \text{ with } \; c_i(t^n) = c^n \tag{17}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A - l_B} A c_i(t) \; + \; B c_{i+1}(t), \text{ with } \; c_{i+1}(t^n) = c^n \; , \tag{18}$$

where $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at the time level $t = t^n$. We assume A is the fine spatial discretized operator on level l_A, where B i the coarse discretized operator on level l_B.

The operators are coupled by the restriction and prolongation operators :

$$A_{coarse} = R^{l_A - l_B} A, \tag{19}$$

$$B_{fine} = P^{l_A - l_B} B, \tag{20}$$

where R is the restriction and P the prolongation operator.

2.5. *The Waveform-Relaxation Method*

A further method to solve large coupled differential equations are the wave-form relaxation scheme.

We assume two spatial operators A, B, which are discretized by finite difference of finite element methods.
We solve the time-discretization of our equations

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \ + \ P^{l_A - l_B} B c_{i-1}(t), \text{ with } \ c_i(t^n) = c^n \tag{21}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A - l_B} A c_i(t) \ + \ Bc_{i+1}(t), \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{22}$$

where $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at time-level $t = t^n$. with time-integration and obtain:

$$(I - A)c_i(t^{n+1}) = c_i(t^n) \ + \ P^{l_A - l_B} B c_{i-1}(t^{n+1}), \text{ with } \ c_i(t^n) = c^n \tag{23}$$

$$(I - B)c_{i+1}(t) = c_{i+1}(t^n) \ + R^{l_A - l_B} A c_i(t^{n+1}), \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{24}$$

We have the multi-grid equations :

$$L_i c_i(t^{n+1}) = c_i(t^n) \ + \ P^{l_A - l_B} B c_{i-1}(t^{n+1}), \text{ with } \ c_i(t^n) = c^n \tag{25}$$

$$L_{i+1} c_{i+1}(t) = c_{i+1}(t^n) \ + R^{l_A - l_B} A c_i(t^{n+1}), \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{26}$$

The iterative method was discussed in [41] and can be done either with Gauss- or Jacobian form.

We deal with the following ordinary differential equation or assume a semi-discretized partial differential equation :

$$u_t \ = f(u, t), \text{ in } (0, T) \ ,$$

$$u(0) = v_0$$

where $u = (u_1, \ldots, u_m)^t$ and $f(u, t) = (f_1(u, t), \ldots, f_m(u, t))^t$.

We apply the Waveform-Relaxation method for $i = 0, 1, \ldots m$ and have :

$$\frac{\partial u_{1,i}(x,t)}{\partial t} = f_1(u_{1,i}, u_{2,i-1}, \ldots, u_{m,i-1}) \text{ with } \ u_{1,i}(t^n) = u_1(t^n) \tag{27}$$

$$\frac{\partial u_{2,i}(x,t)}{\partial t} = f_2(u_{1,i-1}, u_{2,i}, u_{3,i-1} \ldots, u_{m,i-1}) \text{ with } \ u_{2,i}(t^n) = u_2(t^n) \tag{28}$$

$$\vdots$$

$$\frac{\partial u_{m,i}(x,t)}{\partial t} = f_2(u_{1,i-1}, \ldots, u_{m-1,i-1}, u_{m,i}) \text{ with } \ u_{m,i}(t^n) = u_m(t^n) \tag{29}$$

where for the initialization of the first step we have $u_{1,-1}(t) = u_1(t^n), \ldots, u_{m,-1}(t) = u_m(t^n)$.

We reduce to 2 equations and reformulate the method to our iterative splitting methods.

So we deal with :

$$\frac{\partial u_1}{\partial t} \ = f_{11}(u_1, t) + f_{12}(u_2, t), \text{ in } (0, T) \ , \tag{30}$$

$$\frac{\partial u_2}{\partial t} \ = f_{21}(u_1, t) + f_{22}(u_2, t), \text{ in } (0, T) \ , \tag{31}$$

$$u(0) = v_0 \tag{32}$$

where $u = (u_1, u_2)^t$.

Our notation for the operator equation is given as :

$$\frac{\partial u}{\partial t} = A(u) + B(u), \text{ in } (0,T) \ , \tag{33}$$

$$u(0) = v_0 \tag{34}$$

where

$$A(u) = \begin{pmatrix} f_{11}(u_1) \\ f_{21}(u_1) \end{pmatrix} \tag{35}$$

$$B(u) = \begin{pmatrix} f_{12}(u_2) \\ f_{22}(u_2) \end{pmatrix} \tag{36}$$

The iterative splitting method as Waveform-Relaxation method written for $i = 0, 1, \ldots m$ as :

$$\frac{\partial u_i}{\partial t} = A(u_{1,i}, u_{2,i-1}) + B(u_{1,i-1}, u_{2,i}) \tag{37}$$

$$\text{with } \ u_{1,i}(t^n) = u_1(t^n) \text{ and } u_{2,i}(t^n) = u_2(t^n)$$

where for the initialization of the first step we have $u_{1,-1}(t) = u_1(t^n), u_{2,-1}(t) = u_2(t^n)$.

2.5.1. *Convergence Analysis*

In the following we formulate the iterative splitting method as Waveform-Relaxation method and apply the proof-techniques of the relaxation schemes.

The Waveform-Relaxation Method is given as

$$\frac{du_i}{dt} = Pu_i + Qu_{i-1} + f, \tag{38}$$

$$u_i(t^n) = u(t^n), \tag{39}$$

where $A = P + Q$, e.g. P is the diagonal part of A (Jacobi-method).

Here the splitting method is done abstract with respect to the matrix A. The method considered an effective solver method with respect to the underlying matrices.

The reformulation of the iterative operator splitting method is given as:

$$\frac{du_i}{dt} = Pu_i + Qu_{i-1} + f, \tag{40}$$

$$u_i(t^n) = u(t^n), \tag{41}$$

$$\frac{du_{i+1}}{dt} = Pu_i + Qu_{i+1} + f, \tag{42}$$

$$u_{i+1}(t^n) = u(t^n), \tag{43}$$

where P, Q are matrices given by spatial discretization, e.g. P is the convection part of Q the diffusion part.

But we can also do an abstract decomposition, take into account $A = P + Q$, where P is the matrix with small eigenvalues and Q is the matrix with large eigenvalues.

Theorem 1 *We have the iterative operator splitting methods given as outer and inner iterations.*

Outer Iteration :

$$\frac{dU_i}{dt} = \mathcal{P}U_i + \mathcal{Q}U_{i-1} + F, \tag{44}$$
$$U_i(t^n) = U(t^n), \tag{45}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are the diagonal and outer-diagonal matrices of the splitting methods.

Inner Iteration:

$$\frac{du_j}{dt} = Pu_j + Qu_{j-1} + f, \tag{46}$$
$$u_j(t^n) = u(t^n), j = 1, \ldots, J \tag{47}$$
$$\frac{du_k}{dt} = Pu_J + Qu_k + f, \tag{48}$$
$$u_k(t^n) = u(t^n), k = 1, \ldots, K \tag{49}$$

We have a convergent scheme for $\mathcal{K}$ is bounded in a Banach-space :

$$\rho(\mathcal{K}) := \lim_{k \to \infty} ||\mathcal{K}^k||^{1/k}, \tag{50}$$

where $\rho(\mathcal{K}) \leq 1$ is the spectral radius of $\mathcal{K}$ and is given by the variation of constants of $\mathcal{P}$ and $\mathcal{Q}$.

Proof.

The outer iteration given as

$$\frac{dU_i}{dt} = \mathcal{P}U_i + \mathcal{Q}U_{i-1} + F, \tag{51}$$
$$U_i(t^n) = U(t^n), \tag{52}$$

where $\mathcal{P}$ and $\mathcal{Q}$ are the diagonal and outer-diagonal matrices of the splitting methods, can be solved by the variation of constants.

We introduce the linear integral operator

$$\mathcal{K}U(t) = \int_0^t \exp((t-s)\mathcal{P})\mathcal{Q}U(s)\, ds, \tag{53}$$

and we have

$$U_i = \mathcal{K}U_{i-1} + \phi, \tag{54}$$
$$\phi = \exp(t\mathcal{P})U(t^n) + \int_0^t \exp((t-s)\mathcal{P})F(s)\, ds, \tag{55}$$

For the convergence it is sufficient to show :

$\mathcal{K}$ is bounded in a Banach-space :

$$\rho(\mathcal{K}) := \lim_{k \to \infty} ||\mathcal{K}^k||^{1/k}, \tag{56}$$

where $\rho(\mathcal{K}) \leq 1$ is the spectral radius of $\mathcal{K}$

This is given by the bounded operators, see [33] and [34].

3. Error Analysis: Stability Theory for the iterative splitting method with Multi-grid methods

The following algorithm is based on embedding the multi-grid method in the operator splitting method. The iteration with fixed splitting discretization step-size τ. On the time interval $[t^n, t^{n+1}]$ we solve the following sub-problems consecutively for $i = 0, 2, \ldots 2m$. (cf. [24] and [9].)

$$\frac{\partial c_i(t)}{\partial t} = Ac_i(t) \ + \ P^{l_A - l_B} B c_{i-1}(t), \text{ with } \ c_i(t^n) = c^n \tag{57}$$

$$\frac{\partial c_{i+1}(t)}{\partial t} = R^{l_A - l_B} A c_i(t) \ + \ B c_{i+1}(t), \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{58}$$

where $c_0(t^n) = c^n$, $c_{-1} = 0$ and c^n is the known split approximation at time-level $t = t^n$. We assume A to be the fine spatial discretized operator on level l_A and B to be the coarse discretized operator on level l_B.

The operators are coupled by the restriction and prolongation operators:

$$A_{coarse} = R^{l_A - l_B} A, \tag{59}$$

$$B_{fine} = P^{l_A - l_B} B, \tag{60}$$

where R is the restriction and P the prolongation operator.

Theorem 2 *Let us consider the abstract Cauchy problem in a Banach space* $\mathbf{X}$

$$\begin{aligned} \partial_t c(t) &= Ac(t) + P^{l_A - l_B} Bc(t), \quad 0 < t \leq T \\ c(0) &= c_0 \end{aligned} \tag{61}$$

where $A, P^{l_A - l_B} B, A + P^{l_A - l_B} B : \mathbf{X} \to \mathbf{X}$ *are given linear operators being generators of the* C_0*-semigroup and* $c_0 \in \mathbf{X}$ *is a given element. Then the iteration process (81)–(82) is convergent and the rate of convergence is of higher order.*

Proof.

We assume $A + P^{l_A - l_B} B \in \mathcal{L}(X)$ and assume a generator of a uniformly continuous semi-group, hence the problem (61) has a unique solution $c(t) = \exp((A + P^{l_A - l_B} B)t)c_0$. Let us consider the iteration (**??**)–(**??**) on the sub-interval $[t^n, t^{n+1}]$. For the local error function $e_i(t) = c(t) - c_i(t)$ we have the relations

$$\begin{aligned} \partial_t e_i(t) &= Ae_i(t) + P^{l_A - l_B} B e_{i-1}(t), \quad t \in (t^n, t^{n+1}], \\ e_i(t^n) &= 0, \end{aligned} \tag{62}$$

and

$$\partial_t e_{i+1}(t) = R^{l_A - l_B} A e_i(t) + B e_{i+1}(t), \quad t \in (t^n, t^{n+1}],$$
$$e_{i+1}(t^n) = 0, \tag{63}$$

for $m = 0, 2, 4, \ldots$, with $e_0(0) = 0$ and $e_{-1}(t) = c(t)$. In the following we use the notations $\mathbf{X}^2$ for the product space $\mathbf{X} \times \mathbf{X}$ endowed with the norm $\|(u, v)\| = \max\{\|u\|, \|v\|\}$ $(u, v \in \mathbf{X})$. The elements $\mathcal{E}_i(t)$, $\mathcal{F}_i(t) \in \mathbf{X}^2$ and the linear operator $\mathcal{A} : \mathbf{X}^2 \to \mathbf{X}^2$ are defined as follows

$$\mathcal{E}_i(t) = \begin{bmatrix} e_i(t) \\ e_{i+1}(t) \end{bmatrix}, \quad \mathcal{F}_i(t) = \begin{bmatrix} P^{l_A - l_B} B e_{i-1}(t) \\ 0 \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} A & 0 \\ R^{l_A - l_B} A & B \end{bmatrix}. \tag{64}$$

Then using the notations (64), the relations (62) and (63) can be written in the form

$$\partial_t \mathcal{E}_i(t) = \mathcal{A}\mathcal{E}_i(t) + \mathcal{F}_i(t), \quad t \in (t^n, t^{n+1}],$$
$$\mathcal{E}_i(t^n) = 0. \tag{65}$$

Due to our assumptions, $\mathcal{A}$ is a generator of the one-parameter C_0-semi-group $(\exp \mathcal{A}t)_{t \geq 0}$, hence using the variations of constants formula, the solution to the abstract Cauchy problem (65) with homogeneous initial condition can be written as

$$\mathcal{E}_i(t) = \int_{t^n}^{t} \exp(\mathcal{A}(t - s)) \mathcal{F}_i(s) ds, \quad t \in [t^n, t^{n+1}]. \tag{66}$$

(See, e.g. [8].) Hence, using the denotation

$$\|\mathcal{E}_i\|_\infty = \sup_{t \in [t^n, t^{n+1}]} \|\mathcal{E}_i(t)\|, \tag{67}$$

we have

$$\|\mathcal{E}_i\|(t) \leq \|\mathcal{F}_i\|_\infty \int_{t^n}^{t} \|\exp(\mathcal{A}(t - s))\| ds$$
$$= \|B\| \|e_{i-1}\| \int_{t^n}^{t} \|\exp(\mathcal{A}(t - s))\| ds, \quad t \in [t^n, t^{n+1}]. \tag{68}$$

Since $(\mathcal{A}(t))_{t \geq 0}$ is a semi-group, the so called *growth estimation*

$$\| \exp(\mathcal{A}t)\| \leq K \exp(\omega t), \quad t \geq 0, \tag{69}$$

holds with some numbers $K \geq 0$ and $\omega \in \mathbb{R}$, cf. [8].

– Assume that $(\mathcal{A}(t))_{t\geq 0}$ is a bounded or a exponentially stable semi-group, i.e. (69), holds with some $\omega \leq 0$. Then obviously the estimate

$$\|\exp(\mathcal{A}t)\| \leq K, \quad t \geq 0, \tag{70}$$

holds, and hence, on base of (68) we have the relation

$$\|\mathcal{E}_i\|(t) \leq K\|P^{l_A - l_B} B\|\tau_n \|e_{i-1}\|, \quad t \in [t^n, t^{n+1}]. \tag{71}$$

– Assume that $(\exp \mathcal{A}t)_{t\geq 0}$ has an exponential growth with some $\omega > 0$. Using (69), we have

$$\int_{t^n}^{t} \|\exp(\mathcal{A}(t-s))\| ds \leq K_\omega(t), \quad t \in [t^n, t^{n+1}], \tag{72}$$

where

$$K_\omega(t) = \frac{K}{\omega}\left(\exp(\omega(t - t^n)) - 1\right), \quad t \in [t^n, t^{n+1}]. \tag{73}$$

Hence

$$K_\omega(t) \leq \frac{K}{\omega}\left(\exp(\omega\tau_n) - 1\right) = K\tau_n + \mathcal{O}(\tau_n^2). \tag{74}$$

The estimations (71) and (74) result in

$$\|\mathcal{E}_i\|_\infty = K\|P^{l_A - l_B}\|\|B\|\tau_n\|e_{i-1}\| + \mathcal{O}(\tau_n^2). \tag{75}$$

Taking into account the definition of $\mathcal{E}_i$ and the norm $\|\cdot\|_\infty$, we obtain

$$\|e_i\| = K\|P^{l_A - l_B}\|\|B\|\tau_n\|e_{i-1}\| + \mathcal{O}(\tau_n^2), \tag{76}$$

and hence

$$\|e_{i+1}\| = K_1\tau_n^2\|e_{i-1}\| + \mathcal{O}(\tau_n^3), \tag{77}$$

which proves our statement.

3.1. *Operator-splitting method with embedded Jacobian Newton iterative method*

The Newton's method is used to solve the nonlinear parts of the iterative operator-splitting method, see the linearization techniques in [24], [25]. We apply the iterative operator-splitting method and obtain:

$$\begin{aligned}
&F_1(u_i) = \partial_t u_i - A(u_i)u_i - B(u_{i-1})u_{i-1} = 0,\\
&\text{with} \quad u_i(t^n) = c^n,\\
&F_2(u_{i+1}) = \partial_t u_{i+1} - A(u_i)u_i - B(u_{i+1})u_{i+1} = 0,\\
&\text{with} \quad u_{i+1}(t^n) = c^n,
\end{aligned}$$

where the time step is $\tau = t^{n+1} - t^n$. The iterations are $i = 1, 3, \dots, 2m+1$. $c_0(t) = 0$ is the starting solution and c^n is the known split approximation at time level $t = t^n$. The results of the methods are $c(t^{n+1}) = u_{2m+2}(t^{n+1})$. The splitting method with embedded Newton's method is given as

$$
\begin{aligned}
& u_i^{(k+1)} = u_i^{(k)} - D(F_1(u_i^{(k)}))^{-1}(\partial_t u_i^{(k)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i-1}^{(k)})u_{i-1}^{(k)}), \\
& \text{with } D(F_1(u_i^{(k)})) = -(A(u_i^{(k)}) + \frac{\partial A(u_i^{(k)})}{\partial u_i^{(k)}} u_i^{(k)}), \\
& \text{and } k = 0, 1, 2, \ldots, K, \text{ with } \quad u_i(t^n) = c^n,
\end{aligned}
$$

$$
\begin{aligned}
& u_{i+1}^{(l+1)} = u_{i+1}^{(l)} - D(F_2(u_{i+1}^{(l)}))^{-1}(\partial_t u_{i+1}^{(l)} - A(u_i^{(k)})u_i^{(k)} - B(u_{i+1}^{(k)})u_{i+1}^{(k)}) c_2^{(l)}), \\
& \text{with } D(F_2(u_{i+1}^{(l)})) = -(B(u_{i+1}^{(l)}) + \frac{\partial B(u_{i+1}^{(l)})}{\partial u_{i+1}^{(l)}} u_{i+1}^{(l)}), \\
& \text{and } l = 0, 1, 2, \ldots, L, \text{ with } \quad u_{i+1}(t^n) = c^n.
\end{aligned}
$$

Remark 3 *For the iterative operator-splitting method with Newton's method we have two iteration procedures. The first iteration is Newton's method for computing the solution of the nonlinear equations, the second iteration is the iterative splitting method, which computes the resulting solution of the coupled equation systems. The embedded method is used for strong nonlinearities.*

4. Assembling of the Splitting methods with embedded multi-grid method

In the following we discuss the assembling of our methods.

4.1. *Crank-Nicolson and Two-grid Method*

In the following we apply the iterative splitting method, with a Crank-Nicolson method in time, a second order finite difference method in space and underly a two-grid method as solver.

We apply the Crank-Nicolson scheme *separately* to the two equations obtained by the two steps of the iterative operator splitting method after the discretization of the 2D heat equation. For the i-step of the iterative operator splitting method, as interpolation operator we choose the *bilinear interpolation* given by

$$
\begin{aligned}
v_{2i,2j}^h &= v_{ij}^{2h} , \\
v_{2i+1,2j}^h &= \frac{1}{2}(v_{ij}^{2h} + v_{i+1,j}^{2h}) , \\
v_{2i,2j+1}^h &= \frac{1}{2}(v_{ij}^{2h} + v_{i,j+1}^{2h}) , \\
v_{2i+1,2j+1}^h &= \frac{1}{4}(v_{ij}^{2h} + v_{i+1,j}^{2h} + v_{i,j+1}^{2h} + v_{i+1,j+1}^{2h}) , \quad 0 \leq i, j, \leq \frac{n}{2} - 1.
\end{aligned}
$$

where the superscript h is associated with the fine grid and $2h$ with the coarse grid.

Step i:

$$\frac{u^{n+1}_{i,j,k}-u^{n}_{i,j,k}}{\Delta t}=\frac{1}{2}\left[\left(D_1\frac{u^{n+1}_{i-1,j,k}-2u^{n+1}_{i,j,k}+u^{n+1}_{i+1,j,k}}{h_x^2}+PD_2\frac{u^{n+1}_{i-1,j-1,k-1}-2u^{n+1}_{i-1,j,k-1}+u^{n+1}_{i-1,j+1,k-1}}{h_y^2}\right)\right. +$$

$$\left.\left(D_1\frac{u^{n}_{i-1,j,k}-2u^{n}_{i,j,k}+u^{n}_{i+1,j,k}}{h_x^2}+PD_2\frac{u^{n}_{i-1,j-1,k-1}-2u^{n}_{i-1,j,k-1}+u^{n}_{i-1,j+1,k-1}}{h_y^2}\right)\right]$$

$$\equiv\frac{D_1}{2}\hat{\delta}_x^2u^{n+1}_{i,j,k}+\frac{D_2}{2}P\hat{\delta}_y^2u^{n+1}_{i-1,j,k-1}+\frac{D_1}{2}\hat{\delta}_x^2u^{n}_{i,j,k}+\frac{D_2}{2}P\hat{\delta}_y^2u^{n}_{i-1,j,k-1}$$

$$\Leftrightarrow Pu^{n+1}_{i-1,j-1,k-1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i-1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i,j,k}+u^{n+1}_{i,j,k}\frac{D_1\Delta t}{h_x^2}+Pu^{n+1}_{i-1,j,k-1}\frac{D_2\Delta t}{h_y^2}$$

$$+Pu^{n+1}_{i-1,j+1,k-1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)=\frac{1}{2}(D_1\hat{\delta}_x^2u^{n}_{i,j,k}+D_2P\hat{\delta}_y^2u^{n}_{i-1,j,k-1})+u^{n}_{i,j,k}$$

$$\Leftrightarrow u^{n+1}_{i-1,j-1,k}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i-1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i,j,k}+u^{n+1}_{i,j,k}\frac{D_1\Delta t}{h_x^2}+\frac{1}{2}(u^{n+1}_{i-1,j-1,k}+u^{n+1}_{i-1,j+1,k})\frac{D_2\Delta t}{h_y^2}$$

$$+\frac{1}{4}(u^{n+1}_{i-1,j+1,k}+u^{n+1}_{i,j+1,k}+u^{n+1}_{i-1,j+2,k}+u^{n+1}_{i,j+2,k})\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}D_1\hat{\delta}_x^2u^{n}_{i,j,k}+\frac{1}{2}D_2P\frac{u^{n}_{i-1,j-1,k-1}-2u^{n}_{i-1,j,k-1}+u^{n}_{i-1,j+1,k-1}}{h_y^2}+u^{n}_{i,j,k}$$

$$\Leftrightarrow u^{n+1}_{i-1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i,j,k}(1+\frac{D_1\Delta t}{h_x^2})+u^{n+1}_{i-1,j+1,k}\frac{D_2\Delta t}{2h_y^2}$$

$$+\frac{1}{4}(u^{n+1}_{i-1,j+1,k}+u^{n+1}_{i,j+1,k}+u^{n+1}_{i-1,j+2,k}+u^{n+1}_{i,j+2,k})\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}D_1\hat{\delta}_x^2u^{n}_{i,j,k}+\frac{1}{2}\frac{D_2}{h_y^2}[u^{n}_{i-1,j-1,k}-2\frac{1}{2}(u^{n}_{i-1,j-1,k}+u^{n}_{i-1,j+1,k})+\frac{1}{4}(u^{n}_{i-1,j+1,k}+u^{n}_{i,j+1,k}+u^{n}_{i-1,j+2,k}+u^{n}_{i,j+2,k})]+u^{n}_{i,j,k}$$

$$\Leftrightarrow u^{n+1}_{i-1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i,j,k}(1+\frac{D_1\Delta t}{h_x^2})+u^{n+1}_{i-1,j+1,k}\left(\frac{D_2\Delta t}{2h_y^2}\right)$$

$$+u^{n+1}_{i,j+2,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i-1,j+1,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i,j+1,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i-1,j+2,k}\left(\frac{-D_2\Delta t}{4h_y^2}\right)+u^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)$$

$$=\frac{1}{2}D_1\hat{\delta}_x^2u^{n}_{i,j,k}+\frac{1}{2}\frac{D_2}{h_y^2}[-\frac{1}{2}u^{n}_{i-1,j,k}-\frac{3}{4}u^{n}_{i-1,j+1,k}+\frac{1}{4}(u^{n}_{i,j+1,k}+u^{n}_{i-1,j+2,k}+u^{n}_{i,j+2,k})]+u^{n}_{i,j,k}$$

$$\Leftrightarrow u^{n+1}_{i,j,k}\left(\underbrace{1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{2h_y^2}}_{c}\right)+u^{n+1}_{i,j-2,k}\left(\underbrace{\frac{-D_2\Delta t}{4h_y^2}}_{a}\right)+u^{n+1}_{i-1,j,k}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)+$$

$$+u^{n+1}_{i,j+2,k}\left(\underbrace{\frac{-D_2\Delta t}{4h_y^2}}_{a}\right)+u^{n+1}_{i+1,j,k}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)$$

$$=\underbrace{\frac{1}{2}D_1\hat{\delta}_x^2u^n_{i,j,k}+\frac{1}{2}\frac{D_2}{h_y^2}[\frac{1}{2}(u^n_{i,j,k}+u^n_{i,j-2,k})-2u^n_{i,j,k}+\frac{1}{2}(u^n_{i,j+2,k}+u^n_{i,j,k})]+u^n_{i,j,k}}_{d^n_{i,j,k}}$$

$$\Leftrightarrow au_{i,j-2,k}+bu_{i-1,j,k}+cu_{i,j,k}+bu_{i+1,j,k}+au_{i,j+2,k}=d^n_{i,j,k}$$

This procedure leads to a linear system $A\mathbf{u}=\mathbf{f}$, where

$$\mathbf{u}=[u_{2,2}\;u_{3,2}\;\ldots\;u_{n/2-1,2},\;u_{2,3}\;u_{3,3}\;\ldots\;u_{n/2-1,3},\;\ldots\;u_{2,n/2-1}\;u_{3,n/2-1}\;\ldots\;u_{n/2-1,n/2-1}]^T$$

$$\mathbf{f}=[d_{2,2}\;d_{3,2}\;\ldots\;d_{n/2-1,2},\;d_{2,3}\;d_{3,3}\;\ldots\;d_{n/2-1,3},\;\ldots\;d_{2,n/2-1}\;d_{3,n/2-1}\;\ldots\;d_{n/2-1,n/2-1}]^T$$

The index $k+1$ is omitted in the vectors $\mathbf{u},\ f$ for the sake of better presentation. The matrix A is given as:

$$
A = \left[\begin{array}{ccccccccccccccccc}
c & b & 0 & 0 & 0 & 0 & 0 & a & & & & & & & & & \\
b & c & b & & & & & & a & & & & & & & & \\
0 & b & c & b & & & & & & a & & & & & & & \\
0 & & b & c & b & & & & & & a & & & & & & \\
0 & & & b & c & b & & & & & & & & & & & \\
0 & & & & b & c & b & & & & & & & & & & \\
0 & & & & & b & c & b & & & & & & & & & \\
a & & & & & & b & c & b & & & & & & & & \\
0 & a & & & & & & b & c & b & & & & & & & \\
\vdots & & \ddots & & & & & & \ddots & \ddots & \ddots & & & & \ddots & & \\
\vdots & & & \ddots & & & & & & \ddots & \ddots & \ddots & & & & \ddots & \\
 & & & & & a & & & & & b & c & b & & & & a \\
\vdots & & & & & & \ddots & & & & & \ddots & \ddots & \ddots & & & \vdots \\
 & & & & & & & & & & & & & & & & \\
 & & & & & & & & a & & & & & & b & c & b \\
0 & & & & & & & & & a & 0 & 0 & 0 & 0 & 0 & b & c
\end{array}\right]
$$

For the $(i+1)$-step of the iterative operator splitting method, as restriction operator we choose the most obvious restriction operator, the *injection*. We prefer it instead of the standard full weighting operator because it leads to a linear system with better structure that is easier to handle. Injection is defined by

$$
v_{i,j}^{2h} \; = \; v_{2i,2j}^{h}.
$$

Step $i+1$:

$$
\frac{u_{i+1,j,k+1}^{n+1} - u_{i+1,j,k+1}^{n}}{\Delta t} = \frac{1}{2}\left[\left(RD_1 \frac{u_{i-1,j,k}^{n+1} - 2u_{i,j,k}^{n+1} + u_{i+1,j,k}^{n+1}}{h_x^2} + D_2 \frac{u_{i+1,j-1,k+1}^{n+1} - 2u_{i+1,j,k+1}^{n+1} + u_{i+1,j+1,k+1}^{n+1}}{h_y^2}\right) + \right.
$$

$$
\left.\left(RD_1 \frac{u_{i-1,j,k}^{n} - 2u_{i,j,k}^{n} + u_{i+1,j,k}^{n}}{h_x^2} + D_2 \frac{u_{i+1,j-1,k+1}^{n} - 2u_{i+1,j,k+1}^{n} + u_{i+1,j+1,k+1}^{n}}{h_y^2}\right)\right]
$$

$$
\equiv \frac{D_1}{2} R\hat{\delta}_x^2 u_{i,j,k}^{n+1} + \frac{D_2}{2}\hat{\delta}_y^2 u_{i+1,j,k+1}^{n+1} + \frac{D_1}{2} R\hat{\delta}_x^2 u_{i,j,k}^{n} + \frac{D_2}{2}\hat{\delta}_y^2 u_{i+1,j,k+1}^{n}
$$

$$
\Leftrightarrow u_{i+1,j-1,k+1}^{n+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right) + Ru_{i-1,j,k}^{n+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right) + u_{i+1,j,k+1}^{n+1} + Ru_{i,j,k}^{n+1}\frac{D_1\Delta t}{h_x^2} + u_{i+1,j,k+1}^{n+1}\frac{D_2\Delta t}{h_y^2}
$$

$$+u^{n+1}_{i+1,j+1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+Ru^{n+1}_{i+1,j,k}\left(\frac{-D_1\Delta t}{2h_x^2}\right)=\frac{1}{2}(D_1R\hat{\delta}_x^2u^n_{i,j,k}+D_2\hat{\delta}_y^2u^n_{i+1,j,k+1})+u^n_{i+1,j,k+1}$$

$$\Leftrightarrow u^{n+1}_{i+1,j-1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i-1,j,k+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)+u^{n+1}_{i+1,j,k+1}+u^{n+1}_{i,j,k+1}\frac{D_1\Delta t}{h_x^2}+u^{n+1}_{i+1,j,k+1}\frac{D_2\Delta t}{h_y^2}$$

$$+u^{n+1}_{i+1,j+1,k+1}\left(\frac{-D_2\Delta t}{2h_y^2}\right)+u^{n+1}_{i+1,j,k+1}\left(\frac{-D_1\Delta t}{2h_x^2}\right)=\frac{1}{2}(D_1\hat{\delta}_x^2u^n_{i,j,k+1}+D_2\hat{\delta}_y^2u^n_{i+1,j,k+1})+u^n_{i+1,j,k+1}$$

$$\Leftrightarrow u^{n+1}_{i,j-1,k+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u^{n+1}_{i-1,j,k+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)+u^{n+1}_{i,j,k+1}\underbrace{(1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{h_y^2})}_{c}$$

$$+u^{n+1}_{i,j+1,k+1}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right)+u^{n+1}_{i+1,j,k+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right)=\underbrace{\frac{1}{2}(D_1\hat{\delta}_x^2u^n_{i,j,k+1}+D_2\hat{\delta}_y^2u^n_{i+1,j,k+1})+u^n_{i,j,k+1}}_{d^n_{i,j,k+1}}$$

$$\Leftrightarrow au_{i,j-1,k+1}+bu_{i-1,j,k,k+1}+cu_{i,j,k+1}+bu_{i+1,j,k+1}+au_{i,j+1,k+1}=d^n_{i,j,k+1}$$

$A\mathbf{u}=\mathbf{f}$, where

$$\mathbf{u}=[u_{2,2}\;u_{3,2}\;\ldots\;u_{n/2-1,2},\;u_{2,3}\;u_{3,3}\;\ldots\;u_{n/2-1,3},\;\ldots\;u_{2,n/2-1}\;u_{3,n/2-1}\;\ldots\;u_{n/2-1,n/2-1}]^T$$

$$\mathbf{f}=[d_{2,2}\;d_{3,2}\;\ldots\;d_{n/2-1,2},\;d_{2,3}\;d_{3,3}\;\ldots\;d_{n/2-1,3},\;\ldots\;d_{2,n/2-1}\;d_{3,n/2-1}\;\ldots\;d_{n/2-1,n/2-1}]^T$$

(the index $k+1$ is omitted in the vectors $\mathbf{u},\ f$ for the sake of better presentation)

$$A = \begin{bmatrix}
c & b & 0 & 0 & a & 0 & & & & & & & & \\
b & c & b & & & a & & & & & & & & \\
0 & b & c & b & & & a & & & & & & & \\
0 & & b & c & b & & & a & & & & & & \\
a & & & b & c & b & & & a & & & & & \\
0 & a & & & b & c & b & & & a & & & & \\
& & a & & & b & c & b & & & a & & & \\
\vdots & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & & \\
\vdots & & & & \ddots & & & \ddots & \ddots & \ddots & & & \ddots & \\
& & & & & a & & & b & c & b & & & a \\
\vdots & & & & & & \ddots & & & \ddots & \ddots & \ddots & & \vdots \\
& & & & & & & & & & & & & \\
& & & & & & & & a & & & b & c & b \\
0 & & & & & & & & 0 & a & 0 & 0 & b & c
\end{bmatrix}$$

The Jacobi iterations for the heat equation are given as:

$$u_{ij}^{(n+1)} = \Delta t \, D_1 \frac{-u_{i-1,j}^{(n+1)} + 2u_{ij}^{(n+1)} - u_{i+1,j}^{(n+1)}}{h_x^2} + \Delta t \, D_2 \frac{-u_{i,j-1}^{(n+1)} + 2u_{ij}^{(n+1)} - u_{i,j+1}^{(n+1)}}{h_y^2} + u_{ij}^{(n)}$$

4.2. *Implementation of the iterative splitting method with embedded two-grid method*

The implementation for the heat equation is given as:

$$\frac{u_{i,j,k}^{n+1} - u_{i,j,k}^{n}}{\Delta t} = \frac{1}{2}\left[\left(D_1 \frac{u_{i-1,j,k}^{n+1} - 2u_{i,j,k}^{n+1} + u_{i+1,j,k}^{n+1}}{h_x^2} + PD_2 \frac{u_{i,j-1,k-1}^{n+1} - 2u_{i,j,k-1}^{n+1} + u_{i,j+1,k-1}^{n+1}}{h_y^2}\right) + \left(D_1 R \frac{u_{i-1,j,k}^{n} - 2u_{i,j,k}^{n} + u_{i+1,j,k}^{n}}{h_x^2} + D_2 \frac{u_{i,j-1,k+1}^{n} - 2u_{i,j,k+1}^{n} + u_{i,j+1,k+1}^{n}}{h_y^2}\right)\right]$$

$$\equiv \frac{D_1}{2}\hat{\delta}_x^2 u_{i,j,k}^{n+1} + \frac{D_2}{2} P \hat{\delta}_y^2 u_{i,j,k-1}^{n+1} + \frac{D_1}{2} R \hat{\delta}_x^2 u_{i,j,k}^{n} + \frac{D_2}{2}\hat{\delta}_y^2 u_{i,j,k+1}^{n}$$

$$\Leftrightarrow P u_{i,j-1,k-1}^{n+1}\left(\frac{-D_2 \Delta t}{2h_y^2}\right) + R u_{i-1,j,k}^{n+1}\left(\frac{-D_1 \Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1} + u_{i,j,k}^{n+1}\frac{D_1 \Delta t}{h_x^2} + P u_{i,j,k-1}^{n+1}\frac{D_2 \Delta t}{h_y^2}$$

$$+ P u_{i,j+1,k-1}^{n+1}\left(\frac{-D_2 \Delta t}{2h_y^2}\right) + R u_{i+1,j,k}^{n+1}\left(\frac{-D_1 \Delta t}{2h_x^2}\right) + u_{i,j,k}^{n+1} = \frac{1}{2}(D_1 R \hat{\delta}_x^2 u_{i,j,k}^{n} + D_2 \hat{\delta}_y^2 u_{i,j,k+1}^{n}) + u_{i,j,k}^{n}$$

$$\Leftrightarrow u^{n+1}_{i,j-1,k}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right) + u^{n+1}_{i-1,j,k-1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right) + u^{n+1}_{i,j,k}\underbrace{(1+\frac{D_1\Delta t}{h_x^2}+\frac{D_2\Delta t}{h_y^2})}_{c}$$

$$+u^{n+1}_{i,j+1,k}\left(\underbrace{\frac{-D_2\Delta t}{2h_y^2}}_{a}\right) + u^{n+1}_{i+1,j,k+1}\left(\underbrace{\frac{-D_1\Delta t}{2h_x^2}}_{b}\right) = \underbrace{\frac{1}{2}(D_1\hat{\delta}_x^2 u^n_{i,j,k+1} + D_2\hat{\delta}_y^2 u^n_{i,j,k+1}) + u^n_{i,j,k}}_{d^n_{i,j}}$$

$$\Leftrightarrow a u^{n+1}_{i,j-1,k} + b u^{n+1}_{i-1,j,k-1} + c u^{n+1}_{i,j,k} + a u^{n+1}_{i,j+1,k} + b u^{n+1}_{i+1,j,k+1} = d^n_{i,j} \qquad (1)$$

The restriction operators are given as:
$R u^{n+1}_{i,j,k}$ lead to $\frac{1}{16}[u^{n+1}_{i-1,j-1,k-1} + 2u^{n+1}_{i,j-1,k-1} + u^{n+1}_{i+1,j-1,k-1} + 2u^{n+1}_{i-1,j,k-1} + 4u^{n+1}_{i,j,k-1} + 2u^{n+1}_{i+1,j,k-1} + u^{n+1}_{i-1,j+1,k-1} + 2u^{n+1}_{i,j+1,k-1} + u^{n+1}_{i+1,j+1,k-1}]$,

according to the interpretation of the stencil $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$.

5. Numerical Experiments

In the following examples, we deal with different test examples and their underlying multi-scale physics.

5.1. *Simple Heat Equation*

We deal with a PDE which is parabolic and has a stiffness in the diffusion part.
We have the following equation :

$$\partial_t u_1 = D_{11}\partial_{xx}u_1 + D_{21}\partial_{xx}u_2 \ ,$$

$$\partial_t u_2 = D_{12}\partial_{xx}u_1 + D_{22}\partial_{xx}u_2 \ , \qquad (78)$$

$$u_1(0) = u_{10} \ , \ u_2(0) = u_{20} \text{ (initial conditions)} \ , \qquad (79)$$

$$u_1(x,t) = u_2(x,t) = 0 \text{ (boundary conditions)} \ , \qquad (80)$$

where $D_{11}, D_{21}, D_{12}, D_{22} \in \mathbb{R}^+$ and $D_{11}, D_{12} < D_{21}, D_{22}$ are the diffusion operators.
We apply the finite difference scheme for the spatial derivatives :
$\partial_{xx} = [1 \ -2 \ 1]$ and $\partial_{yy} = [1 \ -2 \ 1]^t$.
With the standard projection and restriction :

$$R_H = 1/16\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$P_h = 4R_H$

The time-derivations are discretized by implicit Euler methods and we obtain the following linear equation system:

$$(c_i^{n+1} - c_i^n)/\Delta t = A c_i^{n+1} \ + \ PB c_{i-1}^{n+1}, \text{ with } \ c_i(t^n) = c^n \tag{81}$$

$$(c_{i+1}^{n+1} - c_{i+1}^n)/\Delta t = RA c_i^{n+1} \ + \ B c_{i+1}^{n+1}, \text{ with } \ c_{i+1}(t^n) = c^n \ , \tag{82}$$

We obtain after the insertion of the operators an linear equation system:

$$\tilde{A} U_i^{n+1} = \tilde{B} U_{i-1}^{n+1} + \tilde{C} U_i^n, \tag{83}$$

This equation system is solved by a two-grid method.

Here we have two scales and decouple :

$$\partial_t u = Au + Bu \ , \tag{84}$$

$$u(0) = (u_{10}, u_{20})^T, \tag{85}$$

where $u(t) = (u_1(t), u_2(t))^T$ for $t \in [0, T]$.

Our spitted operators are

$$A = \begin{pmatrix} D_{11}\partial_{xx} & 0 \\ D_{12}\partial_{xx} & 0 \end{pmatrix} \ , \ B = \begin{pmatrix} 0 & D_{21}\partial_{xx} \\ 0 & D_{22}\partial_{xx}. \end{pmatrix} . \tag{86}$$

We chose such an example to have $AB \neq BA$, and therefore we have a splitting error of first order for the usual sequential splitting methods, called A-B splitting.

Our numerical results based on two-grid-methods are presented in the following Table 1.

We choose $D_{11} = D_{12} = 0.5$ and $D_{21} = D_{22} = 0.05$ on the time interval [0,1]. As 2th order method we choose Crank-Nicolson with $\theta = 1/2$. As 4th order method we choose the Gauss Runge-Kutta . We apply a two-grid method, see Subsection 4.1.

The numerical results are given in Table 1. We apply the L_2 error based on the exact and the numerical solution.

Remark 4 *In the experiments, we improved the iterative splitting scheme with higher order discretization methods. Further we can accelerate the splitting scheme with an underlying two-grid method, that is embedded in the splitting scheme. Both higher order time-discretization and also two-grid methods are necessary to obtain such results.*

5.2. *Second example: Transport-reaction equation*

We deal with the following system of transport equation:

$$\partial_t u_1 + v_1 \partial_x u_1 = -\lambda u_1 + \mu u_2, \tag{87}$$

$$\partial_t u_2 + v_2 \partial_x u_2 = \mu u_1 - \lambda u_2, \tag{88}$$

where we have Dirichlet boundary conditions and $u_{1,0}, u_{2,0}$ are the initial conditions.

Number of time-partitions	Iterative Steps	err_1 (2th order)	err_2 (2th order)	err_1 (4th order)	err_2 (4th order)
2	1	4.5321e-002	3.6077e-003	4.5321e-002	3.6077e-003
2	10	3.9664e-003	4.7396e-004	3.9664e-003	4.7397e-004
2	100	3.9204e-004	4.8078e-005	3.9204e-004	4.8083e-005
3	1	4.6126e-004	3.6077e-003	4.6126e-004	3.6077e-003
3	10	7.8129e-006	2.9285e-005	7.8069e-006	2.9289e-005
3	100	8.5988e-008	2.8270e-007	8.0050e-008	2.8682e-007
4	1	4.6126e-004	2.2459e-005	4.6126e-004	2.2464e-005
4	10	4.1883e-007	4.2629e-008	4.1321e-007	4.8154e-008
4	100	5.9521e-009	5.4846e-009	4.0839e-010	4.9968e-011

Table 1
Numerical results for the first example with the iterative splitting method and 2 and 4 th order method.

We split the operator in two operators

$$A = \begin{pmatrix} -v_1 \partial_x & \\ & -v_2 \partial_x \end{pmatrix} \quad , \quad B = \begin{pmatrix} -\lambda & \mu \\ \mu & -\lambda \end{pmatrix} \tag{89}$$

We see that for $\mu \approx 0$ the operators commutate.

We choose $v_1 = 1$, $v_2 = 0.5$, $\lambda = 0.01$.
We let $t \in [0, 40]$ and $x \in [0, 40]$.
We set $\Delta t = \frac{1}{25}$ and $\Delta x = \frac{4}{10}$

In the first figure 2, we choose $\mu = 0.001$ and see that we could say $\mu \approx 0$ and obtain nearly the same results as for the A-B splitting.

In the second figure 3, we choose $\mu = 0.01$ and see that $\mu \neq 0$ and obtain more optimal results for the iterative schemes.

The result is given in the following Figures 2-4.

Remark 5 *In this example, we concentrate on the comparison between standard A-B splitting and iterative splitting methods. We obtain more accurate results for non-commuting problems. Such problems are related to multi-scale problems and we can apply our embedded multi-grid method. We have the benefit of receiving higher order results without reducing of the time-steps.*

6. Conclusions and Discussions

We present iterative splitting methods with embedded multi-grid and multi-stepping methods. Here the idea were to achieve more accurate and faster results as for standard splitting schemes (e.g. A-B splitting schemes). We obtain a benefit with the embedded

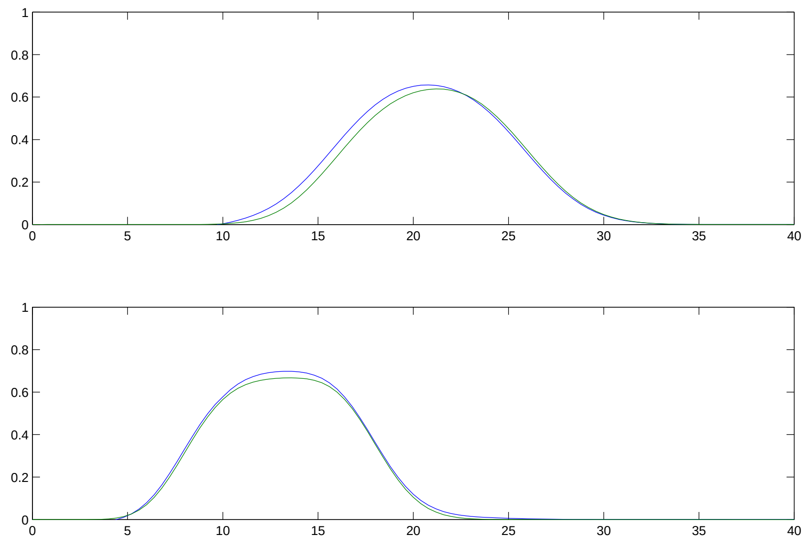

Fig. 2. Blue - Iterative Operator Splitting (4 iterations) Green - A-B Splitting

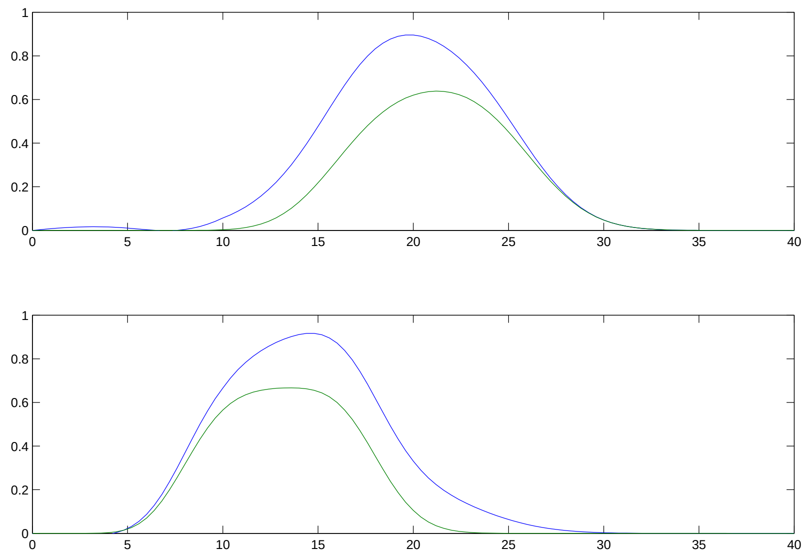

Fig. 3. T=2: Blue - Iterative Operator Splitting (4 iterations) Green - A-B Splitting

method to accelerate the iterative schemes and achieve much more accurate results. In future we concentrate on nonlinear and matrix dependent splitting scheme.

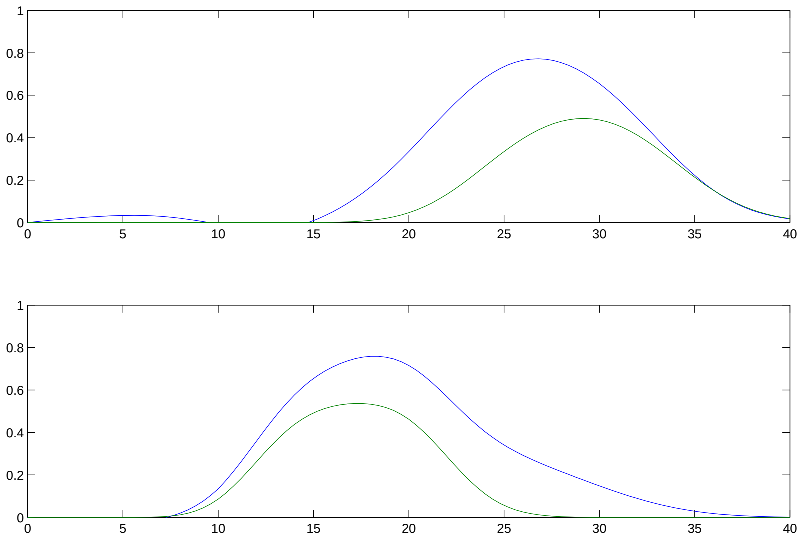

Fig. 4. T=60: Blue - Iterative Operator Splitting (4 iterations) Green - A-B Splitting

References

[1] W. Bangerth and R. Rannacher. *Adaptive finite Element Methods for differential equations*. Lectures in Mathematics, ETH Zürich, Birkhäuser Verlag, Basel-Boston-Berlin, 2003.

[2] D. Braess. *Finite Elemente*. Springer-Verlag, Berlin, Heidelberg, New York, 1992.

[3] X.C. Cai, Additive Schwarz algorithms for parabolic convection-diffusion equations, Numer. Math. 60 (1991) 41-61.

[4] X.C. Cai, Multiplicative Schwarz methods for parabolic problems, SIAM J. Sci Comput. 15 (1994) 587-603.

[5] W. Cheney, Analysis for Applied Mathematics, Graduate Texts in Mathematics., 208, Springer, New York, Berlin, Heidelberg, 2001.

[6] C. N. Dawson, Q. Du, and D. F. Dupont, A finite Difference Domain Decomposition Algorithm for Numerical solution of the Heat Equation, Mathematics of Computation 57 (1991) 63-71.

[7] P. Deuflhard, Newton Methods for Nonlinear Problems, Springer-Verlag, New-York, Berlin, 2004.

[8] K.-J. Engel and R. Nagel, One-Parameter Semigroups for Linear Evolution Equations, Springer, New York, 2000.

[9] I. Farago and J. Geiser, Iterative Operator-Splitting methods for Linear Problems, Preprint No. 1043 of Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany. International Journal of Computational Science and Engineering, accepted September 2007.

[10] M.J. Gander and H. Zhao, Overlapping Schwarz waveform relaxation for parabolic problems in higher dimension, In A. Handlovičová, Magda Komorníkova, and Karol Mikula, editors, in: Proc. Algoritmy 14, Slovak Technical University, 1997, pp. 42-51.

[11] E. Giladi and H. Keller, Space time domain decomposition for parabolic problems. Technical Report 97-4, Center for research on parallel computation CRPC, Caltech, 1997.

[12] J. Geiser, Discretisation Methods with embedded analytical solutions for convection dominated transport in porous media, in: Proc. NA&A '04, Lecture Notes in Computer Science, Vol.3401, Springer, Berlin, 2005, pp. 288-295.

[13] J. Geiser, Iterative Operator-Splitting Methods with higher order Time-Integration Methods and Applications for Parabolic Partial Differential Equations, J. Comput. Appl. Math., accepted, June 2007.

[14] J. Geiser, O. Klein, and P. Philip. Influence of anisotropic thermal conductivity in the apparatus insulation for sublimation growth of SiC: Numerical investigation of heat transfer. Crystal Growth & Design 6, 2021 - 2028, 2006.

[15] J. Geiser. *Decomposition Methods for Differential Equations: Theory and Applications.* CRC Press, Taylor and Francis Group, Numerical Analysis and Scientific Computing Series, edited by Magoules and Lai, 2009.

[16] J. Geiser and M. Arab. *Modelling and Simulation of a Chemical Vapor Deposition.* Journal of Applied Mathematics, special issue: Mathematical and Numerical Modeling of Flow and Transport (MNMFT), Hindawi Publishing Corp., New York, accepted, January 2011.

[17] W. Hackbusch. *Multi-Gird Methods and Applications.* Springer-Verlag, Berlin, Heidelberg, 1985.

[18] W. Hackbusch. *Iterative L"osung gro"ser schwachbesetzter Gleichungssysteme.* Teubner Studienb"ucher: Mathematik, B.G. Teubner Stuttgart, 1993.

[19] E. Hannsen and A. Ostermann, Dimensional splittingfor evolution equations. Numer. Math., 108, 557-570, 2008.

[20] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2003.

[21] M. Holst, R. Kozack, F. Saied and S. Subramaniam, Treatment of Electrostatic Effects in Proteins: Multigrid-based Newton Iterative Method for Solution of the Full Nonlinear Poisson-Boltzmann Equation, Proteins: Stucture, Function, and Genetics, vol. 18, 231–245, 1994.

[22] W. Hundsdorfer and J.G. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Series in Computational Mathematics Vol. 33, Springer Verlag, 2003.

[23] R. Jeltsch and O. Nevanlinna, *Stability and accuracy of time discretizations for initial value problems.* Numerische Mathematik, 40(2), 245-296, 1982.

[24] J. Kanney, C. Miller, and C.T. Kelley, Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems, Advances in Water Resources 26 (2003) 247-261.

[25] K.H. Karlsen and N. Risebro. *An Operator Splitting method for nonlinear convection-diffusion equation.* Numer. Math., 77, 3 , 365–382, 1997.

[26] K.H. Karlsen and N.H. Risebro, Corrected operator splitting for nonlinear parabolic equations, SIAM J. Numer. Anal. 37 (2000) 980-1003.

[27] K.H. Karlsen, K.A. Lie, J.R. Natvig, H.F. Nordhaug and H.K. Dahle, Operator splitting methods for systems of convection-diffusion equations: nonlinear error mechanisms and correction strategies, J. Comput. Phys. 173 (2001) 636-663.

[28] C.T. Kelly. *Iterative Methods for Linear and Nonlinear Equations.* Frontiers in Applied Mathematics, SIAM, Philadelphia, USA, 1995.

[29] P. Knabner and L. Angermann, Numerical Methods for Elliptic and Parabolic Partial Differential Equations, Text in Applied Mathematics, Springer Verlag, Newe York, Berlin, vol. 44, 2003.

[30] M.A. Lieberman and A.J. Lichtenberg. Principle of Plasma Discharges and Materials Processing., Wiley-Interscience, John Wiley & Sons, Inc Publication, Second Edition, 2005.

[31] G.I. Marchuk, Some applicatons of splitting-up methods to the solution of problems in mathematical physics, Aplikace Matematiky 1 (1968) 103-132.

[32] G.A. Meurant, Numerical experiments with a domain decomposition method for parabolic problems on parallel computers, in: R. Glowinski, Y.A. Kuznetsov, G.A. Meurant, J. Périaux and O. Widlund, (Ed.), Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA, 1991. SIAM.

[33] U Miekkala and O. Nevanlinna. *Sets of convergence and stability regions.* BIT Numerical Mathematics, 27, 554-584, 1987.

[34] U Miekkala and O. Nevanlinna. *Convergence of dynamic iteration methods for initial value systems.* SIAM, 8(4), 459-482, 1987.

[35] H.A. Schwarz, Über einige Abbildungsaufgaben, Journal für Reine und Angewandte Mathematik 70 (1869) 105-120.

[36] H. Roos, M. Stynes and L. Tobiska. Numerical Methods for Singular Perturbed Differential Equations, Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[37] H.A. Schwarz, Über einige Abbildungsaufgaben, Journal für Reine und Angewandte Mathematik 70 (1869) 105-120.

[38] G. Strang, On the construction and comparision of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506-517.

[39] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations* Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, New-York, 1997.

[40] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations* Series: Numerical Mathematics and Scientific Computation, Clarendon Press, Oxford, 1999.

[41] S. Vandewalle. *Parallel Multigrid Waveform Relaxation for Parabolic Problems.* B.G. Teubner, Stuttgart, 1993.

[42] H.A. van der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, New York, 2003.

[43] H. Yoshida, Construction of higher order symplectic integrators, Physics Letters A, Vol. 150, no. 5,6,7, 1990.

[44] H. Yserentant. *Old and New Convergence Proofs for Multigrid Methods.* Acta Numerica, 285–326, 1993.

[45] E. Zeidler. Nonlinear Functional Analysis and its Applications. II/B Nonlinear montone operators Springer-Verlag, Berlin-Heidelberg-New York, 1990.

[46] Z. Zlatev. *Computer Treatment of Large Air Pollution Models.* Kluwer Academic Publishers, 1995.