Error analysis of the numerical solution of split differential equations

P. Csomós, I. Faragó

Eötvös Loránd University,
Department of Applied Analysis and Computational Mathematics,
H–1117 Budapest, Pázmány Péter st. 1/C, Hungary

Abstract

The operator splitting method is a widely used approach for solving partial differential equations describing physical processes. Its application usually requires the use of certain numerical methods in order to solve the different split sub-problems. The error-analysis of such a numerical approach is a complex task. In the present paper we show that an interaction error appears in the numerical solution when an operator splitting procedure is applied together with a lower order numerical method. The effect of the interaction error is investigated by an analytical study and by numerical experiments made for a test problem.

Key words: Numerical simulation, operator splitting procedures, error analysis.
PACS: 02.60.Cb

1 Introduction

Complex physical processes are frequently modelled by systems of linear or non-linear partial differential equations. Due to the complexity of these equations, typically, there is no numerical method which can provide a numerical solution accurate enough while taking reasonable integrational time. In order to simplify the task, the operator splitting procedure has been introduced ([1,2]), which is widely used for solving advection–diffusion problems (see e.g. in [3,4]), Hamilton–Jacobi equation (see e.g. in [5,6]), and Navier–Stokes equation (see e.g. in [7]), including modelling turbulence and interfaces (see e.g. [8]). More applications can also be seen in [3].

Email addresses: csomos@cs.elte.hu (P. Csomós), faragois@cs.elte.hu (I. Faragó).

Preprint submitted to Elsevier Science 24 April 2006
Operator splitting means that the spatial differential operator appearing in the equations is split into a sum of different sub-operators having simpler forms, and the corresponding equations can be solved easier. The sub-operators are usually chosen regarding to the different physical processes or geometric directions. Then instead of the original problem, a sequence of sub-models is solved, which gives rise to the splitting error. The order of the splitting error can be estimated theoretically (e.g. [9]). In practice, splitting procedures are associated with different numerical methods for solving the sub-problems, which also cause a certain amount of error. This can usually lead to interaction between the two types of errors: the splitting error and the numerical error. Since the orders of the applied splitting procedure and the numerical method are known, the question is which order their common effect results in. The interaction, therefore, can be measured through the order of the total time-discretization method (which means the application of the splitting and the numerical method together for solving the equation numerically). If the numerical method is not properly chosen for the applied splitting procedure then this can lead to order reduction, therefore, loss of accuracy. Moreover, the choices of the step size of the numerical method and the time parameter of the splitting procedure play an important role, too.

Our aim is to analyze the behaviour of the different errors appearing in the solution. We investigate whether the total error of this complex method is really the sum of the numerical and the splitting errors. In practice, only the order of the splitting is taken into account when considering the accuracy (the order of the numerical method is usually not taken into consideration). This means that the order of the total error is considered to be equal to the order of the applied splitting procedure. Therefore, we investigate the conditions under which this consideration really fulfills. We are also going to find a quantity which characterizes the accuracy of the method well, without knowing the exact solution.

In the operator splitting community it is well known that there is a kind of order reduction when applying lower order numerical method than splitting procedure, however, this statement has never been examined thoroughly. In this paper we give a first step towards the systematic survey of the phenomenon, using simple mathematical and numerical tools, and through simple problems.

The paper is organised as follows. In Section 2 we give a motivation to our work through a specific example. In Section 3 the frequently applied operator splitting procedures are introduced. In Section 4 we define and investigate the different kinds of errors appearing in the numerical solving processes. In Section 5 analytical computations are presented in the case of bounded operators and first-order numerical methods. In Section 6 we perform numerical experiments for a test problem by using higher order numerical methods, as well. In Section 7 we summarize our results.
2 Illustration

In what follows we illustrate the interaction between the splitting procedure and the numerical method on a simple problem applying *sequential splitting* (see definition later on) and *explicit Euler method*.

Let \((x(t), y(t))^T\) denote a function of type \(\mathbb{R} \to \mathbb{R}^2\), and \(0 \leq t \leq T\). The time-evolution of the **harmonic oscillator** is described by the following equation (see, e.g. [10]):

\[
\begin{align*}
\dot{x}(t) &= y(t) \\
\dot{y}(t) &= -x(t),
\end{align*}
\tag{1}
\]

with \(x(0) = x_0 \in \mathbb{R}\) and \(y(0) = y_0 \in \mathbb{R}\). The **exact solution** of system (1) reads

\[
\begin{align*}
x(t) &= y_0 \sin t + x_0 \cos t \\
y(t) &= y_0 \cos t - x_0 \sin t,
\end{align*}
\tag{2}
\]

where \(x(t)\) and \(y(t)\) represent the amplitude and the velocity of the oscillator, respectively. The simplest example of an oscillating system is a mass connected to a rigid foundation with a spring.

Equation (1) can be written as:

\[
\frac{d}{dt} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}.
\tag{3}
\]

Using the decomposition (splitting) of the matrix appearing in (3):

\[
\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix},
\tag{4}
\]

and applying the sequential splitting, the **split solution** in the points \(k\tau\) \((k = 0, 1, \ldots, m)\) with \(\tau := \frac{T}{m}\) \((m \in \mathbb{N})\) has the form:

\[
\begin{align*}
x_{\text{spl}}((k+1)\tau) &= y_{\text{spl}}(k\tau) \sin \tau + x_{\text{spl}}(k\tau) \cos \tau \\
y_{\text{spl}}((k+1)\tau) &= y_{\text{spl}}(k\tau) \cos \tau - x_{\text{spl}}(k\tau) \sin \tau,
\end{align*}
\tag{5}
\]

for \(k = 0, 1, \ldots, m\), where \(x_{\text{spl}}(0) = x_0\) and \(y_{\text{spl}}(0) = y_0\). From (2) and (5) one
can easily check that
\[
\begin{align*}
x_{\text{spl}}(k\tau) &= x(k\tau) \\
y_{\text{spl}}(k\tau) &= y(k\tau),
\end{align*}
\]
for \(k = 0, 1, \ldots, m\), hence splitting does not cause any error in this case.

Let us apply explicit Euler method with step size \(\tau\) for (1). Then the numerical solution has the form:
\[
\begin{align*}
x^{k+1} &= x^k + \tau y^k \\
y^{k+1} &= y^k - \tau x^k,
\end{align*}
\]
for \(k = 0, 1, \ldots, m\), where \(x^0 = x_0\) and \(y^0 = y_0\).

If the explicit Euler method is applied to the split problems, the numerical split solution has the following form:
\[
\begin{align*}
x_{\text{spl}}^{k+1} &= x_{\text{spl}}^k + \tau y^k - \tau^2 \left( x_{\text{spl}}^k - y_{\text{spl}}^k \right) \\
y_{\text{spl}}^{k+1} &= y_{\text{spl}}^k - \tau x^k - \tau^2 \left( x_{\text{spl}}^k + y_{\text{spl}}^k \right),
\end{align*}
\]
for \(k = 0, 1, \ldots, m\), where \(x_{\text{spl}}^0 = x_0\) and \(y_{\text{spl}}^0 = y_0\).

Since the split and the exact solutions do not differ, we are expecting that if we use the same numerical method for both the split and the unsplitted problems, then the numerical solution and the numerical split solution do not differ, as well. However, comparing (6) and (7) it can be seen that there appears a certain error in the numerical split solution. Hence, we can see that there is a certain interaction between the splitting procedure and the numerical method.

In our numerical experiments we used \(\tau = \frac{2\pi}{200}\) while integrating system (1) on the time interval \([0, 4\pi]\) with and without applying splitting (the period of the harmonic oscillator is \(2\pi\)). On the left panel of Figure 1 the effect of the numerical error can be seen: the numerical solution is spiraling outwards compared to the exact solution. On the right panel of Figure 1 the effect of the interaction error can be seen: the numerical split solution is spiraling inwards compared to the exact solution (which coincide the split solution in this case). Hence, in this case the interaction between the numerical method and the splitting procedure causes the “turn” of the spiral.

Before investigating the effect of the interaction error generally, we introduce the operator splitting procedures, and give the proper definitions of the different solutions and errors.
Fig. 1. Effects of the numerical error (left panel) and interaction error (right panel).

3 Operator splitting procedures

Operator splitting procedures are commonly used in modelling various physical phenomena. The basic idea behind the procedure is to divide the spatial differential operator into a few sub-operators of simpler structure. For the sub-problems (which usually represent standard tasks) suitable numerical methods can be found. Then a sequence of the equations corresponding to the sub-operators is solved. The connections between the sub-systems are the initial conditions. As an example, in what follows, we introduce three possible splitting procedures ([11,14]).

For the sake of simplicity, we define the splitting procedures only for systems of ordinary differential equations, i.e. when $A : \mathbb{R}^N \to \mathbb{R}^N$ is a bounded operator (i.e. it can be represented as a matrix $A \in \mathbb{R}^{N \times N}$), and $A$ is a sum of two bounded operators. The generalization for the abstract Cauchy problems, and for more than two sub-operators is straightforward. Consider the following problem:

$$\begin{aligned}
\frac{du(t)}{dt} &= A u(t) = (A_1 + A_2) u(t), \\
u(0) &= u_0,
\end{aligned} \qquad (t \in (0, T])$$

(8)

where $u : (0, T] \to \mathbb{R}^N$ is the unknown function, $u_0 \in \mathbb{R}^N$ is a given element, and $A, A_1, A_2$ are operators of type $\mathbb{R}^N \to \mathbb{R}^N$. We assume that (8) has a unique solution. Let us divide the time interval $[0, T]$ into $m \in \mathbb{N}$ equal sub-intervals with length $\tau$, where $\tau$ is called the splitting timestep ($T = m\tau$).

There exist several splitting procedures (see e.g. in [1,2,11,12,14]). The simplest one is the sequential splitting, defined by the following sequence of sub-
Another splitting technique is the Strang–Marchuk splitting. For a timestep three sub-problems should be solved:

\[
\begin{aligned}
\frac{du_1^{(k)}}{dt} &= A_1 u_1^{(k)}(t), \quad t \in ((k - 1)\tau, k\tau] \\
\quad u_1^{(k)}((k - 1)\tau) &= u_{\text{spl}}((k - 1)\tau) \\
\frac{du_2^{(k)}}{dt} &= A_2 u_2^{(k)}(t), \quad t \in ((k - 1)\tau, k\tau] \\
\quad u_2^{(k)}((k - 1)\tau) &= u_1^{(k)}(k\tau) \\
\quad u_{\text{spl}}(k\tau) &:= u_2^{(k)}(k\tau)
\end{aligned}
\]

with \( k = 1, \ldots, m \), and \( u_{\text{spl}}(0) = u_0 \), where \( u_{\text{spl}}(k\tau) \) is the solution of the split problem defined on the mesh \( \{k\tau, k = 0, 1, \ldots, m\} \).

Another splitting technique is the Strang–Marchuk splitting, where for one splitting timestep three sub-problems should be solved:

\[
\begin{aligned}
\frac{du_1^{(k)}}{dt} &= A_1 u_1^{(k)}(t), \quad t \in ((k - 1)\tau, (k - \frac{1}{2})\tau] \\
\quad u_1^{(k)}((k - 1)\tau) &= u_{\text{spl}}((k - 1)\tau) \\
\frac{du_2^{(k)}}{dt} &= A_2 u_2^{(k)}(t), \quad t \in ((k - 1)\tau, (k - \frac{1}{2})\tau] \\
\quad u_2^{(k)}((k - 1)\tau) &= u_1^{(k)}(\left( k - \frac{1}{2}\right) \tau) \\
\frac{du_3^{(k)}}{dt} &= A_1 u_3^{(k)}(t), \quad t \in ((k - \frac{1}{2})\tau, k\tau] \\
\quad u_3^{(k)}\left( (k - \frac{1}{2})\tau \right) &= u_2^{(k)}(k\tau) \\
\quad u_{\text{spl}}(k\tau) &:= u_3^{(k)}(k\tau),
\end{aligned}
\]

where \( u_{\text{spl}}^0(0) = u_0, k = 1, \ldots, m \).

The weighted splitting can be obtained by using two sequential splittings, once with the order of operators \( A_1 \to A_2 \), and once with \( A_2 \to A_1 \). At time \( t = k\tau \) the numerical solution is computed as a weighted average of the solutions obtained by the two sequential splitting steps:

\[
u_{\text{spl}}(k\tau) = \Theta u_{\text{spl},A_1A_2}(k\tau) + (1 - \Theta) u_{\text{spl},A_2A_1}(k\tau),
\]

where \( \Theta \in [0, 1] \) is a given weight parameter, and \( u_{\text{spl},A_1A_2}(k\tau) \) and \( u_{\text{spl},A_2A_1}(k\tau) \) are the solutions of the two sequential splittings at time \( k\tau \), respectively. The case \( \Theta = \frac{1}{2} \) is called symmetrically weighted splitting.
We note that there is a weighted version of the Strang–Marchuk splitting, as well. Due to their natural parallelization, both types of weighted splittings can be efficiently used on parallel computers (see [13]).

4 Different kinds of errors

According to the aim of this paper, we study the mechanism of the interaction between the different errors appearing in the solution when an operator splitting procedure is applied together with a numerical method. First we introduce two kinds of numerical solving processes, and their solutions and errors. Then we investigate the behaviour of these errors individually.

4.1 Discretization of the time-continuous problem

Discretization means that the time-continuous problem defined on the interval $[0, T]$ is somehow replaced by a sequence of finite dimensional problems. In the case of the present investigations we used finite difference methods. Let us introduce two uniform meshes on $[0, T]$:

$$\omega_\tau := \{k\tau, k = 0, 1, \ldots, m\} \quad \text{and} \quad \omega_h := \{kh, k = 0, 1, \ldots, mn\},$$

where $\tau = \frac{T}{m}$ is the splitting timestep and $h = \frac{\tau}{n}$ is called the step size of the numerical method. Clearly, $\omega_\tau \subset \omega_h$. The structures of these meshes are shown in Figure 2.

![Figure 2. Meshes $\omega_\tau$ and $\omega_h$ on the interval $[0, T]$: $\tau = \frac{T}{m}$ is the splitting timestep, and $h = \frac{\tau}{n}$ is the step size of the numerical method.](image)

The numerical split solution will only be defined on $\omega_\tau$. (The numerical solution of the original problem is defined on $\omega_h$ but it can only be compared with the split solution on $\omega_\tau$.) If $n = 1$ then $\omega_\tau = \omega_h$. Let $X_{\omega_\tau}$ denote the set of the time-discretized functions of type $\omega_\tau \rightarrow \mathbb{R}^N$ with some vector norm $\| \cdot \|_{X_{\omega_\tau}}$. Our aim is to determine a suitable solution to (8) (from $X_{\omega_\tau}$). Thus, an element of $\mathbb{R}^N$ has to be assigned to each point of $\omega_\tau$. This can be done in several ways, for instance:

1. We apply a numerical method directly on the mesh $\omega_h \subset \omega_\tau$.  


First we apply a splitting procedure, and then the split sub-problems are solved by a certain numerical method. (In this case the sub-problems are solved on the finer mesh $\omega_h$, but the solution is defined on $\omega_\tau$.)

In the following we give two examples for (1) and (2):

(1) Discretization of equation (8) on $\omega_h$ in case of using the explicit Euler method results in the scheme:

\[
y^0 := u_0
\]
\[
\text{for } i = 1(1)M
\]
\[
y^{i+1} = y^i + h(A_1 + A_2)y^i
\]
\[
\text{end}
\]

(10)

Here $M = mn - 1$. In this case the numerical solution of the original problem is $y^i$ at the gridpoint $ih$.

(2) Algorithm for discretizing the split problem using the explicit Euler method and the sequential splitting:

\[
y^{(0)}_{\text{spl}} := u_0
\]
\[
\text{for } k = 1(1)m
\]
\[
y^{(k),0}_{1} := y^{(k-1)}_{\text{spl}}
\]
\[
\text{for } j = 0(1)n
\]
\[
y^{(k),i+1}_{1} = y^{(k),j}_{1} + hA_1y^{(k),j}_{1}
\]
\[
\text{end}
\]
\[
y^{(k),0}_{2} := y^{(k),n}_{1}
\]
\[
\text{for } j = 0(1)n
\]
\[
y^{(k),i+1}_{2} = y^{(k),j}_{2} + hA_2y^{(k),j}_{2}
\]
\[
\text{end}
\]
\[
y^{(k)}_{\text{spl}} := y^{(k),n}_{2}
\]
\[
\text{end}
\]

(11)

In this case the numerical split solution is $y^{(k)}_{\text{spl}}$ in the gridpoint $k\tau$.

Algorithms (10) and (11) can also be formulated in the cases of other splitting
procedures and other numerical methods.

Time-discretization means either the first or the second type of discretizing problem (8). We call the second type total time-discretization. (In this case a splitting procedure is applied for solving a differential equation, and a certain numerical method is used for solving the split sub-problems).

4.2 Definitions of the errors appearing in the numerical solution

Before defining different kinds of errors which will be investigated, we introduce four kinds of solutions (as before, \( \tau \) is the splitting timestep, and \( h \leq \tau \) denotes the numerical step size):

- \( u(t) \) denotes the exact solution of the original continuous problem (8) at time \( t \);
- \( u_{\text{spl}}(k\tau) \) denotes the exact solution of the continuous split problem (e.g. (9)) at time \( t = k\tau \), i.e. the split solution;
- \( y_{\text{spl}}^k \) denotes the numerical solution of the split problem (e.g. (11)) at time \( t = k\tau \), i.e. the numerical split solution,

where \( k = 1, \ldots, m \), and

- \( y_{\text{num}}^k \) denotes the numerical solution of the original problem (e.g. (10)) at time \( t = kh \) for \( k = 1, \ldots, mn \).

**Definition 1** Using the above notions, we define the following five kinds of errors:

\[
\begin{align*}
\text{total error:} & \quad E_{\text{tot}}(k\tau) := \|u(k\tau) - y_{\text{spl}}^k\|; \\
\text{splitting error:} & \quad E_{\text{spl}}(k\tau) := \|u(k\tau) - u_{\text{spl}}(k\tau)\|; \\
\text{practical error:} & \quad E_{\text{prac}}(k\tau) := \|y_{\text{num}}^k - y_{\text{spl}}^k\|; \\
\text{interaction error:} & \quad E_{\text{int}}(k\tau) := \|u_{\text{spl}}(k\tau) - y_{\text{spl}}^k\|,
\end{align*}
\]

where \( k = 1, \ldots, m \). Let \( E(k\tau) \) denote any of the above errors. Then \( E(\tau) \) and \( E(m\tau) = E(T) \) are called local and global errors, respectively.

The numerical error can be defined on the finer mesh \( \omega_h \), as well:

\[
\text{numerical error:} \quad E_{\text{num}}(kh) := \|u(kh) - y_{\text{num}}^k\| \quad (12)
\]

for \( k = 1, \ldots, mn \), where \( E_{\text{num}}(h) \) is the local numerical error.
We remark that in practice, we are interested in $E_{\text{tot}}(k\tau)$. However, the exact solution is not known, therefore, we are only be able to measure $E_{\text{prac}}(k\tau)$. Hence, we investigate whether it is possible to estimate $E_{\text{tot}}(k\tau)$ from $E_{\text{prac}}(k\tau)$, i.e. whether the order of $E_{\text{tot}}(k\tau)$ follows from the order of $E_{\text{prac}}(k\tau)$. It will be shown that the interaction error causes the reduction in the order of the total time-discretization method.

According to the definition of the order of the local error in the theory of numerical methods for ordinary differential equations ([15]), we give the following definition.

**Definition 2** The local error $E(\tau)$ (i.e. the corresponding method) has an order of $p$ if

$$p := \sup \left\{ q \in \mathbb{N} : \lim_{\tau \to 0} \frac{E(\tau)}{\tau^{q+1}} = c < +\infty \right\}. \quad (13)$$

The operator splitting procedure is called consistent if $p > 0$ for $E_{\text{spl}}(\tau)$. The total time-discretization method is called consistent if $p > 0$ for $E_{\text{tot}}(\tau)$.

We note that if the local error $E(\tau)$ has an order of $p$ (i.e. $E(\tau) = O(\tau^{p+1})$), then the local error behaves like $E(\tau) = \text{const} \cdot \tau^{p+1}$ for sufficiently small values of $\tau$. When the sub-operators are genuinely non-stiff, then we may assume that their norms are bounded with moderate constants, therefore, for the global error we obtain that

$$E(T) = nO(\tau^{p+1}) = \frac{T}{\tau}O(\tau^{p+1}) = O(\tau^p).$$

This means that the local order defines the order of the global error, as well. (For the stiff-case see [16].)

### 4.3 Local numerical and interaction errors

$E_{\text{num}}(h)$ represents the local numerical error of the applied numerical method of order $r$, therefore, $E_{\text{num}}(h) = O(h^{r+1})$. When $h$ and $\tau$ are connected with the relation $h = \tau^s$ ($s \geq 1$), then

$$E_{\text{num}}(h) = O(\tau^{r+s}). \quad (14)$$

Then $E_{\text{num}}(\tau)$ represents the global numerical error at time $t = \tau$. Its order can be estimated as follows:

$$E_{\text{num}}(\tau) = nO(h^{r+1}) = \frac{T}{h}O(h^{r+1}) = \tau O(h^r) = \tau O(\tau^s) = O(\tau^{r+1}). \quad (15)$$
The interaction error is originated from the fact that when a certain numerical method is used to solve the split sub-problems numerically, then (at least) the numerical error appears in the solution of each sub-problem. Since the initial condition of each sub-problem is the solution of the previously solved sub-problem, perturbed split problems are solved. Therefore, \( E_{int}(\tau) \) corresponds to a certain numerical error, as well. In order to estimate it, let us assume that \( h = \tau^s \) (\( s \geq 1 \)), and \( \tau \to 0 \). Then two cases are possible.

- Assume that \( s = 1 \), hence, \( h = \tau \). Then \( E_{int}(\tau) = E_{int}(h) \) represents the local numerical error, which is proportional to \( h^{r+1} = \tau^{r+1} \). Thus,
  \[
  E_{int}(\tau) = \mathcal{O}(\tau^{r+1}).
  \]  \( (16) \)

- Assume that \( s > 1 \). Hence, \( s = 1 + \beta \) with some \( \beta > 0 \). Therefore,
  \[
  n = \frac{\tau}{h} = \frac{\tau}{\tau^s} = \frac{\tau}{\tau^{1+\beta}} = \frac{1}{\tau^{\beta}},
  \]  so, if \( \tau \to 0 \) then \( n \to +\infty \). Consequently, \( E_{int}(\tau) \) represents the global numerical error at \( \tau \). Therefore, the same computation as in (15) shows that
  \[
  E_{int}(\tau) = \mathcal{O}(\tau^{rs+1}).
  \]  \( (17) \)

From (14), (16) and (17) we can conclude the following.

**Corollary 3** Applying a splitting procedure of order \( p \) together with a numerical method of order \( r \), and setting \( h = \tau^s \) (\( s \geq 1 \)), for the order of the local numerical and interaction errors we get:

\[
E_{num}(\tau) = \|u(\tau) - y_{num}^n\| = \mathcal{O}(\tau^{rs+1}),
\]

\[
E_{int}(\tau) = \|u_{spl}(\tau) - y_{spl}^1\| = \mathcal{O}(\tau^{rs+1}).
\]

### 4.4 Orders of the different splitting procedures

As an example, the order of the sequential splitting will be determined for bounded operators \( A_1 \) and \( A_2 \). The other cases can be found e.g. in [9,11].

The exact solution of the original problem (8) and the split problem applying sequential splitting (9) at time \( t = \tau \) read as

\[
\begin{align*}
  u(\tau) &= \exp[\tau(A_1 + A_2)]u_0 \\
  u_{spl}(\tau) &= \exp(\tau A_2)\exp(\tau A_1)u_0.
\end{align*}
\]  \( (18) \)
Hence,
\[ E_{\text{spl}}(\tau) = \| \exp [\tau(\mathbf{A}_1 + \mathbf{A}_2)] u_0 - \exp(\tau\mathbf{A}_1) \exp(\tau\mathbf{A}_2) u_0 \|. \]

Hence, using the Taylor-series expansion, we get:
\[ E_{\text{spl}}(\tau) = \frac{\tau^2}{2} \| (\mathbf{A}_1 \mathbf{A}_2 - \mathbf{A}_2 \mathbf{A}_1) u_0 \| + \mathcal{O}(\tau^3). \]

One can see that in the general case the local splitting error is of first order. Moreover, when the sub-operators commute then the local splitting error vanishes ([17]).

Usually, the sequential and the weighted splittings are of first order, the Strang–Marchuk and the symmetrically weighted splittings are of second order (see, e.g. [9]). The symmetrically weighted splitting is of third order if the condition
\[ [[\mathbf{A}_1, \mathbf{A}_2], \mathbf{A}_1 - \mathbf{A}_2] = 0 \] (19)
holds, where \([\mathbf{A}_1, \mathbf{A}_2] := \mathbf{A}_1 \mathbf{A}_2 - \mathbf{A}_2 \mathbf{A}_1\) denotes the commutator of the sub-operators \(\mathbf{A}_1\) and \(\mathbf{A}_2\) (see [14]).

4.5 Local total error

**Definition 4** Let \(q\) denote the order of the local total error, i.e. \(E_{\text{tot}}(\tau) = \mathcal{O}(\tau^{q+1})\).

Applying a splitting procedure of order \(p\) together with a numerical method of order \(r\) and using the triangular inequality, we can deduce from Definition 1 that
\[ E_{\text{tot}}(\tau) = \| u(\tau) - y_{\text{int}}^1 \| \leq \| u(\tau) - u_{\text{spl}}(\tau) \| + \| u_{\text{spl}}(\tau) - y_{\text{spl}}^1 \| = \frac{\tau^2}{2} \| (\mathbf{A}_1 \mathbf{A}_2 - \mathbf{A}_2 \mathbf{A}_1) u_0 \| + \mathcal{O}(\tau^3). \] (20)

Hence, the local total error can be estimated by the sum of the local splitting error and the interaction error. From the definition of the order of the local splitting error and Corollary 3, from (20) we can conclude the following.

**Proposition 5** Let us apply an operator splitting procedure of order \(p\) together with a numerical method of order \(r\), and set \(h = \tau^s\) \((s \geq 1)\). Then the order of the local total error is: \(q = \min\{p, rs\}\).

This means that in order to preserve the accuracy, we can control it through the parameters \(p, r,\) and \(s\) for a given \(p\). In practice, we apply a given splitting
procedure, i.e. $p$ and $\tau$ are fixed, and we want to preserve the order $p$ to the local total error ($\varrho = p$). Hence, in this case we can estimate it, as well.

From Proposition 5 it follows that the interaction error causes a reduction in the order of $E_{\text{tot}}(\tau)$ unless $rs \geq p$.

**Corollary 6** From Proposition 5 it follows that $\varrho = p$ if $rs \geq p$.

Therefore, we shall answer the following two questions for given values of $p$ and $\tau$.

1. How to choose $h$ for a given numerical method of order $r$?
2. Fixing $h = \tau$, how to choose the order $r$ of the numerical method?

From Proposition 5, the following answers can be stated.

**Proposition 7** When a given splitting procedure of order $p$ is applied together with a given numerical method of order $r \leq p$, and the numerical step size is set as $h = \tau^s$, then the exponent $s$ has to be chosen as $s = \frac{p}{r}$ in order to keep $\varrho = p$. For $r > p$ the identity $\varrho = p$ holds independently of the choice of $h$.

**Remark 8** For the case $r > p$, clearly, the choice $h = \tau$ is optimal, because in this case the integration of the model needs the least computational work.

**Proposition 9** When a given splitting procedure of order $p$ is applied together with a certain numerical method of order $r$, and the numerical step size is varied as $h = \tau^s$ ($s \geq 1$), then $r$ has to be chosen as $r = \left\lceil \frac{p}{s} \right\rceil + 1 \in \mathbb{N}$ in order to keep $\varrho = p$.

**Remark 10** Higher order numerical methods could be chosen, as well, but it would not lead to a higher order total time-discretization. It would only need more computational work.

### 4.6 Local practical error

Since the order of $E_{\text{tot}}(\tau)$ cannot be derived when the exact solution is not known, we can only treat with the local practical error $E_{\text{prac}}(\tau)$. Hence, we shall investigate whether the order of $E_{\text{prac}}(\tau)$ differs from the order of $E_{\text{tot}}(\tau)$. Using the definitions of the local errors, and the triangular inequality, we get:

\[
|E_{\text{prac}}(\tau) - E_{\text{tot}}(\tau)| = \left\| y_{\text{num}}^n - y_{\text{spl}}^1 \right\| - \left\| y_{\text{spl}}^1 - u(\tau) \right\| \leq \left\| u(\tau) - y_{\text{num}}^n \right\| = E_{\text{num}}(\tau). \quad (21)
\]
Using Corollary 3 we obtain that

$$|E_{prac}(\tau) - E_{tot}(\tau)| = O(\tau^{s+1}).$$

**Corollary 11** $E_{prac}(\tau)$ and $E_{tot}(\tau)$ only differ in the order of the global numerical error. This means that we can characterize the accuracy of the total time-discretization method with the local practical error, as well. It becomes important when the exact solution is not known.

In the next section the order of the above introduced local errors will be examined through an example.

## 5 Analytical computations

In this section we investigate the behaviour of the previously studied errors. As before, we only treat with bounded operators, because the splitting error, and the order of the splitting is usually defined and investigated only for them. We note that the original and the split problems are solved by using the same numerical method.

### 5.1 Expressions of the solutions

To demonstrate the effect of the interaction error, we compute the expressions of the exact, the numerical, and the numerical split solutions of (8) up to $O(\tau^{3})$, using the explicit and implicit Euler method.

The exact solution of the original problem (8) and the split solutions obtained by the different splitting procedures, can be written in the following way:

**Exact solution:**

$$u(\tau) = \exp[\tau(A_1 + A_2)] u_0$$

**Sequential splitting:**

$$u^{sq}_{spl}(\tau) = \exp(\tau A_2) \exp(\tau A_1) u_0$$

**Strang–Marchuk sp.:**

$$u^{MS}_{spl}(\tau) = \exp \left( \frac{\tau}{2} A_1 \right) \exp(\tau A_2) \exp \left( \frac{\tau}{2} A_1 \right) u_0$$

**Weighted splitting:**

$$u^{\Theta}_{spl}(\tau) = [\Theta \exp(\tau A_2) \exp(\tau A_1) + (1 - \Theta) \exp(\tau A_1) \exp(\tau A_2)] u_0.$$  

(22)

Applying the **explicit Euler** method, the numerical and numerical split solu-
tions can be written in the following form (using the notation \( y_{\text{spl}} := y_{\text{spl}}^1 \)):

**Numerical solution:** \[ y_{\text{num}}^n = [I + h(A_1 + A_2)]^n u_0 \]

**Sequential splitting:** \[ y_{\text{sq}}^{\text{spl}} = ([I - hA_2]^{-1})^n ([I - hA_1]^{-1})^n u_0 \]  \hspace{1cm} (23)

**Strang–Marchuk splitting:** \[ y_{\text{MS}}^{\text{spl}} = ([I - hA_1]^{-1})^n ([I - hA_2]^{-1})^n \times ([I - hA_1]^{-1})^n u_0 \]  \hspace{1cm} (24)

**Weighted splitting:** \[ y_{\text{\Theta}}^{\text{spl}} = \Theta \left[ ([I - hA_2]^{-1})^n ([I - hA_1]^{-1})^n + (1 - \Theta) ([I - hA_2]^{-1})^n ([I - hA_1]^{-1})^n \right] u_0. \]

Applying the *implicit Euler* method, the numerical and numerical split solutions can be written in the following form (using the notation \( y_{\text{spl}} := y_{\text{spl}}^1 \)):

**Numerical solution:** \[ y_{\text{num}}^n = [I - h(A_1 + A_2)]^{-1} u_0 \]

**Sequential splitting:** \[ y_{\text{sq}}^{\text{spl}} = ([I - hA_2])^{-1} ([I - hA_1])^{-1} u_0 \]

**Strang–Marchuk splitting:** \[ y_{\text{MS}}^{\text{spl}} = ([I - hA_1])^{-1} ([I - hA_2])^{-1} \times ([I - hA_1])^{-1} u_0 \]  \hspace{1cm} (24)

**Weighted splitting:** \[ y_{\text{\Theta}}^{\text{spl}} = \{ \Theta ([I - hA_2])^{-1} ([I - hA_1])^{-1} + (1 - \Theta) ([I - hA_2])^{-1} ([I - hA_1])^{-1} \} u_0. \]

After expanding the exponential function in (22) into Taylor-series up to \( O(\tau^3) \), and using the binomial theorem to expand the other terms in (23) and (24), the solutions \( u(\tau) \), \( y_{\text{num}}^n \) and \( y_{\text{spl}} \) can be written as follows:

\[
\begin{align*}
\begin{pmatrix}
    u(\tau) \\
y_{\text{num}}^n \\
y_{\text{spl}}
\end{pmatrix} &= \left[ I + \tau(A_1 + A_2) + \frac{\tau^2}{2} W + O(\tau^3) \right] u_0, \hspace{1cm} (25)
\end{align*}
\]

where the term \( W \) has the following meaning in the different cases, using
explicit and implicit Euler methods:

Exact solution \((u(\tau))\):
\[
A_1^2 + A_2^2 + A_1A_2 + A_2A_1
\]

Numerical solution \((y_{\text{num}}^n)\):
\[
\left(1 \mp \frac{1}{n}\right)(A_1^2 + A_2^2) + \left(1 \mp \frac{1}{n}\right)(A_1A_2 + A_2A_1)
\]

Sequential splitting \((y_{\text{spl}}^n)\):
\[
\left(1 \mp \frac{1}{n}\right)(A_1^2 + A_2^2) + 2A_2A_1
\]

Strang–Marchuk sp. \((y_{\text{MS}}^n)\):
\[
\left(1 \mp \frac{1}{n}\right)(A_1^2 + A_2^2) + A_1A_2 + A_2A_1
\]

Weighted splitting \((y_{\text{spl}}^\Theta)\):
\[
\left(1 \mp \frac{1}{n}\right)(A_1^2 + A_2^2) + (1 - \Theta)A_1A_2 + \Theta A_2A_1,
\]

where the upper sign refers to the case of the explicit, and the lower sign to the case of the implicit Euler method.

**Corollary 12** All of the introduced splitting techniques combined with the Euler methods are consistent for arbitrary \(h = \frac{\tau}{n}\). The consistency of the total time-discretization method is guaranteed by the uniformity of expressions (25), because the disappearance of the term \(I + \tau(A_1 + A_2)\) in \(E_{\text{tot}}(\tau)\) is needed to satisfy \(p > 0\) in Definition 2.

**Remark 13** We note that from formulae (26) one can derive a higher order time-discretization method, as well. Let the total time-discretization method be chosen as the average of the terms obtained by using the explicit and implicit Euler methods in \(y_{\text{num}}^n\). Then we get the formula of the exact solution up to \(O(\tau^3)\), therefore, this method is of (at least) second order.

**Remark 14** We also note that if we average the terms obtained by using the explicit and implicit Euler methods in the numerical split solutions \((y_{\text{spl}}^n, y_{\text{MS}}^n, \text{and } y_{\text{spl}}^\Theta)\), we get the formulae of the split solutions up to \(O(\tau^3)\).

5.2 Expression of the local errors

From the previously derived formulae (22) and (26), and taking into account that \(h = \frac{\tau}{n}\) one can compute the expressions of the local errors up to \(O(\tau^3)\).

**Proposition 15** From formulae (22) and (26), simple calculations show the followings.
\[ \begin{align*}
E_{\text{spl}}(\tau) &= \gamma \frac{\tau^2}{2} (A_2 A_1 - A_1 A_2) + O(\tau^3); \\
E_{\text{int}}(\tau) &= \chi \frac{\tau^2}{2} (A_1^2 + A_2^2) + O(\tau^3); \\
E_{\text{tot}}(\tau) &= \left\| \left[ \gamma \frac{\tau^2}{2} (A_2 A_1 - A_1 A_2) + \chi \frac{\tau^2}{2} (A_1^2 + A_2^2) \right] u_0 \right\| + O(\tau^3) \\
&\leq E_{\text{spl}}(\tau) + E_{\text{int}}(\tau) + O(\tau^3); \\
E_{\text{prac}}(\tau) &= \left\| \left[ \gamma \frac{\tau^2}{2} (A_2 A_1 + A_1 A_2) - \chi \frac{\tau^2}{2} (A_1^2 + A_2^2) \right] u_0 \right\| + O(\tau^3),
\end{align*} \]

where \( \chi = 1 \) and \( \chi = -1 \) for the explicit and implicit Euler methods, respectively. The values of \( \gamma \) are: 1, 0, and \( \Theta - \frac{1}{2} \) for the sequential, the Strang–Marchuk, and the weighted splitting, respectively.

These results show that estimation (20) cannot be sharpened, i.e. the local total error is less than or equal to the sum of the local splitting and the interaction errors up to \( O(\tau^3) \). Furthermore, the step size of the numerical method has to be chosen as stated in Proposition 7. For the first-order Euler methods this means that for the first-order splittings there is no restriction for the numerical step size due to \( h \leq \tau \). However, in the case of the second-order splittings the numerical step size has to be chosen as \( h = O(\tau^2) \) in order to keep the order of the interaction error (i.e. the local total error) to equal to \( p \).

The explanation of the observed behaviour is that both the explicit and the implicit Euler methods are first-order numerical methods.

One can see that there appear both \( h \) and \( \tau \) in the expression of \( E_{\text{int}}(\tau) \) in (27), i.e. this error is caused by the interaction between the numerical and the splitting errors, indeed. This is the reason why we call it “interaction error”.

**Corollary 16** Although the local total and practical errors do not have the same value, their orders equal up to \( O(\tau^3) \). This is in a good agreement with Corollary 11.

From Corollary 16 it follows that for investigating the order of the total time-discretization method, the local practical error can be computed instead of the local total error.

Finally, we remark that in this section we computed the expressions of the local errors only up to \( O(\tau^3) \). Therefore, we can only state that for the third-order symmetrically weighted splitting with condition (19) the step size has to be chosen at least \( h = O(\tau^2) \). Based on Proposition 7 we are expecting, however, that the suitable choice for the numerical step size should be \( h = O(\tau^3) \).

In the following section we investigate the order of the local total and practical errors by numerical simulation. In this case we can determine the effect of the interaction error also for higher order numerical methods. We note again that the above computations were done using only first order numerical methods.
6 Numerical experiments

We continue our computations related to the order of the local errors by using numerical simulations. We determine the order of $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$ for a test problem, namely, for the system of ordinary differential equations corresponding to the model of the harmonic oscillator.

6.1 Test problem

The model of the harmonic oscillator has already been introduced in Section 2. We choose this model because its exact solution is bounded and periodic, therefore, we can investigate the local errors also for a large range of values of the numerical step size. For our numerical experiments we define the following one-parameter splitting family for the matrix appearing in (3):

$$
\begin{pmatrix}
0 & 1 \\
-1 & 0
\end{pmatrix} = \begin{pmatrix}
\alpha & 0.5 \\
1 & 0
\end{pmatrix} + \begin{pmatrix}
-\alpha & 0.5 \\
-2 & 0
\end{pmatrix},
$$

with the parameter $\alpha \in \mathbb{R}$. With these sub-matrices, the already mentioned splitting schemes can be applied: sequential, Strang–Marchuk, weighted, symmetrically weighted (with $\alpha \neq 0$, in the numerical experiments $\alpha = 0.3$), and symmetrically weighted splitting with condition (19) (with $\alpha = 0$). We used five different numerical methods for solving the sub-problems: explicit Euler, implicit Euler, second-order trapezoidal rule, third- and fourth-order Runge-Kutta method.

6.2 Determination of the local orders

Throughout this section $E(\tau)$ denotes either $E_{\text{tot}}(\tau)$ or $E_{\text{prac}}(\tau)$. Previously the orders of $E(\tau)$ were computed analytically for the explicit and implicit Euler method up to $O(\tau^3)$. Numerically, they can be determined in two ways. As a first way, the order can be computed as a limit. In the second way, the order can be determined by fitting a line on the numerical data of the function $E(\tau)$ in logarithmic scale.
6.2.1 First method

We can apply Definition 2 of the order, with the distinction that \( q \in \mathbb{R} \). Using the notation

\[
H_q(\tau) := \frac{E(\tau)}{\tau^{q+1}},
\]

the following limit has to be computed numerically for different fixed values of \( q \):

\[
\lim_{\tau \to 0} H_q(\tau).
\]

We note that for a fixed step size \( h \ll \tau \), limit (29) should be understood as \( \tau \to h \), therefore, we select \( h \) to be close to the computer zero.

**Definition 17** The order of the local total and practical errors obtained by numerical experiments is called numerical order of the corresponding local error, and it is denoted by \( \varrho_{\text{num}} \).

**Remark 18** It will be shown that \( \varrho_{\text{num}} \) is the same for the local total and practical errors.

By numerical experiments we can only determine a numerical order \( \varrho_{\text{num}} \) of the total time-discretization method. Therefore, we investigate the conditions for \( \varrho_{\text{num}} = p \) instead of \( q = p \). We assume, however, that \( \varrho_{\text{num}} \approx q \).

In the case of the “First method”, the numerical orders of \( E_{\text{tot}}(\tau) \) and \( E_{\text{prac}}(\tau) \) are determined as the supremum of those values of \( q \) for which the limit in (29) is finite. The term \( H_q(\tau) \) has been computed for several values of \( \tau \) and \( q \). In Figure 3 our results are shown for \( E_{\text{prac}}(\tau) \) applying sequential splitting with the explicit Euler method (with a fixed step size \( h = 10^{-6} \ll \tau \)), for the following 20 values of \( q \): 0.9, 0.91, 0.92, . . . , 1.1. The splitting timestep is varied from \( \frac{2\pi}{200} \) to \( \frac{2\pi}{10} \) by \( \frac{2\pi}{200} \) (20 values). We obtained similar figures also in the case of \( E_{\text{tot}}(\tau) \), and for all kinds of splitting techniques and numerical methods. Figure 3 shows the validity of the following statement.

**Corollary 19** There exists a threshold \( q_0 \) such that for \( q \leq q_0 \), function \( H_q(\tau) \) is convergent, and for \( q > q_0 \) it is divergent as \( \tau \) tends to zero.

Threshold \( q_0 \) can be considered as the numerical order of the total time-discretization method (\( E_{\text{tot}}(\tau) \) or \( E_{\text{prac}}(\tau) \)), i.e. \( q_0 = \varrho_{\text{num}} \). According to the orders of the splitting procedure and the numerical method, different values of \( q_0 \) belong to different splittings and numerical methods.

In Figure 4 an example can be seen in order to show how to determine the approximative value of threshold \( q_0 \). In this case the Strang–Marchuk splitting
Fig. 3. Values of the term $H_q(\tau)$ (for $E_{\text{prac}}(\tau)$) as a function of $\tau$ applying sequential splitting procedure with the explicit Euler method, for 20 different values of $q$. (Similar figures for $E_{\text{tot}}(\tau)$, and for all kinds of splittings and numerical methods.)

is used together with the fourth-order Runge-Kutta method (with $h = 10^{-6} \ll \tau$), for $E_{\text{prac}}(\tau)$. The splitting timestep is varied as before. The different values of $q$ are chosen around the value of the order of the splitting procedure ($p = 2$). The last value of $q$ for which the limit is still finite, is considered to be an approximation of $q_0$. We obtained similar figures also in the cases of other splitting procedures and other numerical methods, for $E_{\text{tot}}(\tau)$ as well as for $E_{\text{prac}}(\tau)$.

Fig. 4. Values of the term $H_q(\tau)$ (for $E_{\text{prac}}(\tau)$) as a function of $\tau$, in the cases of sequential splitting using the fourth-order Runge-Kutta method, for three different values of $q$. (Similar figures for $E_{\text{tot}}(\tau)$, and for all kinds of splittings and numerical methods.)

For our experiments we choose the numerical step size as (i) $h = 10^{-6} \ll \tau$ fixed, and (ii) $h = \tau^s$ ($s = 1, 2, 3$). The first choice means that the applied numerical method can be considered very accurate, therefore, its error is negligible compared to the local total and practical errors. However, this choice is inefficient from the computational point of view. Hence, in the second case the value of $h$ is varying with the splitting timestep. (We remark that for the Strang–Marchuk splitting $h = \tau^s$ means $h = \frac{1}{2}\tau^s$.)

In Tables 1, 2 and 3 our results are collected obtained by finding the approximative value of $q_0$. We set the numerical step size $h = 10^{-6} \ll \tau$, $h = \tau$, and
$h = \tau^2$, respectively. We note that we get the same results for $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$. In the case of $h = \tau^3$ there is only one interesting value: when symmetrically weighted splitting with condition (19) is applied using first-order numerical method. Then the orders of $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$ equal $2.99$. The other values are the same as in the case of $h = \tau^2$, i.e. in Table 2.

Table 1
Results on the numerical orders of $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$ obtained by numerical experiments, using different numerical methods of order $r$, with fixed step size $h = 10^{-6} \ll \tau$.

<table>
<thead>
<tr>
<th>Splitting</th>
<th>$r = 1$</th>
<th>$r = 2$</th>
<th>$r = 3$</th>
<th>$r = 4$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>1</td>
</tr>
<tr>
<td>Strang – Marchuk</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>2</td>
</tr>
<tr>
<td>Weighted in general</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>1</td>
</tr>
<tr>
<td>Symmetrically weighted</td>
<td>1.99</td>
<td>1.98</td>
<td>1.95</td>
<td>1.95</td>
<td>2</td>
</tr>
<tr>
<td>Symm. weighted with cond. (19)</td>
<td>2.90</td>
<td>2.99</td>
<td>2.99</td>
<td>2.90</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2
Results on the numerical orders of $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$ obtained by numerical experiments, using different numerical methods of order $r$, with varying step size $h = \tau$.

<table>
<thead>
<tr>
<th>Splitting</th>
<th>$r = 1$</th>
<th>$r = 2$</th>
<th>$r = 3$</th>
<th>$r = 4$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>1.00</td>
<td>0.97</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
</tr>
<tr>
<td>Strang – Marchuk</td>
<td>1.00</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>2</td>
</tr>
<tr>
<td>Weighted in general</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>1</td>
</tr>
<tr>
<td>Symmetrically weighted</td>
<td>1.00</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>2</td>
</tr>
<tr>
<td>Symm. weighted with cond. (19)</td>
<td>1.00</td>
<td>2.00</td>
<td>2.91</td>
<td>2.99</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 3
Results on the numerical orders of $E_{\text{tot}}(\tau)$ and $E_{\text{prac}}(\tau)$ obtained by numerical experiments, using different numerical methods of order $r$, with varying step size $h = \tau^2$.

<table>
<thead>
<tr>
<th>Splitting</th>
<th>$r = 1$</th>
<th>$r = 2$</th>
<th>$r = 3$</th>
<th>$r = 4$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>1</td>
</tr>
<tr>
<td>Strang – Marchuk</td>
<td>$\approx 2$</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>2</td>
</tr>
<tr>
<td>Weighted in general</td>
<td>0.98</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>1</td>
</tr>
<tr>
<td>Symmetrically weighted</td>
<td>2.05</td>
<td>$\approx 2$</td>
<td>$\approx 2$</td>
<td>$\approx 2$</td>
<td>2</td>
</tr>
<tr>
<td>Symm. weighted with cond. (19)</td>
<td>2.00</td>
<td>$\approx 3$</td>
<td>$\approx 3$</td>
<td>$\approx 3$</td>
<td>3</td>
</tr>
</tbody>
</table>
However, this method for determining the numerical order of the total time-discretization method is not so accurate, because the results are slightly subjective. The inaccuracy appears because \( q_0 \) is obtained “by eye”. This is the reason why the symbol “≈” was written in Table 3 in some cases, and that \( q \) is varied only by 0.01 and for instance not by \( 10^{-4} \). In order to improve the data we introduce the “Second method”.

### 6.2.2 Second method

As before, let \( E(\tau) \) denote either \( E_{\text{tot}}(\tau) \) or \( E_{\text{prac}}(\tau) \). From Definition 2 of the order it follows that the total time-discretization method has an order of \( \varrho \) if it is the supremum of those numbers \( q \) for which

\[
\frac{E(\tau)}{\tau^{q+1}} \approx c < +\infty, \tag{30}
\]

for \( \tau \) small enough. Let us consider \( q \in \mathbb{R} \) in order to determine the numerical order. Taking the logarithm of both sides in (30), we obtain:

\[
\log E(\tau) \approx (q + 1) \log \tau + \log c, \tag{31}
\]

which defines a linear function in logarithmic scale. One can see that the slope \( q + 1 \) of this line corresponds to the numerical order of the error. We made several numerical experiments, and fitted a line to the results, using 20 values of \( \tau \) chosen as follows:

\[
\tau_0 = \frac{2\pi}{200} \quad \text{and} \quad \tau_i = \frac{10}{11} \tau_{i-1}, \quad \text{for} \quad i = 1, \ldots, 19.
\]

From the fitted lines we can calculate the slope of the line, which can be regarded as the numerical order of the local total or practical error.

We apply all the introduced splitting procedures and numerical methods. The time step of the numerical method is chosen as \( h = \tau^s \) (\( s = 1, \ldots, 6 \)). In the case of the ”second method” we are also able to investigate the behaviour of the orders \( \varrho_{\text{num}} \) obtained by numerical experiments also as a function of the exponent \( s \). Our results are shown in Figures 5–7, where the orders \( \varrho_{\text{num}} \) obtained by numerical experiments are plotted as a function of exponent \( s \).

On the left panels of the figures the cases of the local total errors can be seen, while on the right panels the behaviour of the local practical errors are shown.

The figures show that similarly to the case of \( q_0 \) (see Proposition 19), there is a critical value of \( s \), as well.

**Corollary 20** Let us apply an operator splitting procedure of order \( p \) together with a given numerical method of order \( r \). Then there exists a threshold \( s_0 \)
Fig. 5. Numerical orders in the cases of the local total (left panel) and the local practical (right panel) errors as a function of $s$ ($h = \tau^s$). The sequential splitting was applied. (Similar figure for the general weighted splitting.)

Fig. 6. Numerical orders in the cases of the local total (left panel) and the local practical (right panel) errors as a function of $s$ ($h = \tau^s$). The Strang–Marchuk splitting was applied. (Similar figure for the symmetrically weighted splitting.)

Fig. 7. Numerical orders in the cases of the local total (left panel) and the local practical (right panel) errors as a function of $s$ ($h = \tau^s$). The symmetrically weighted splitting with condition (19) was applied. Depending on $r$ such that for $s < s_0$: $\rho_{num} < p$, and for $s \geq s_0$: $\rho_{num} = p$.

Approximative values of $s_0$ are shown on the right panel of Table 4. On the
left panel the results according to Proposition 7 \( (s_0 = \frac{p}{r}) \) can be seen.

Table 4
Approximative values of the threshold \( s_0 \) in the cases of operator splitting procedures of order \( p \) and numerical methods of order \( r \). Results from Proposition 7 \( (s_0 = \frac{p}{r}) \) (left panel) and Figures 5–7 (right panel).

<table>
<thead>
<tr>
<th>( s_0 )</th>
<th>( r = 1 )</th>
<th>( r = 2 )</th>
<th>( r = 3 )</th>
<th>( r = 4 )</th>
<th>( r = 1 )</th>
<th>( r = 2 )</th>
<th>( r = 3 )</th>
<th>( r = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p = 1 )</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( p = 2 )</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( p = 3 )</td>
<td>3</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

6.3 Results on the numerical local orders

We can summarize the numerical results as follows. By numerical experiments we can only determine a numerical order \( \varrho_{num} \) of the total time-discretization method. Therefore, our aim is to keep \( \varrho_{num} = p \). Towards this, the numerical step size has to be chosen small enough to reduce the effect of the interaction error, i.e. to adjust the order of the interaction error also equal to \( p \). Hence, for a fixed splitting timestep we can choose \( h \ll \tau \) also fixed. In this case \( \varrho_{num} \approx p \) (see Table 1). However, this choice of \( h \) is not efficient from the computational point of view, because the integration of the model takes long time. Therefore, next we set \( h = \tau^s \) \( (s = 1, \ldots, 6) \) and investigated \( \varrho_{num} \) as a function of the exponent \( s \). From Tables 2 and 3) it can be seen that if \( p \leq r \) then \( h \) can be chosen to equal \( \tau \). However, if \( p > r \) then the higher order splitting is applied in vain (from the point of view of the order) unless \( h \) is chosen small enough. This behaviour can be considered as the effect of the interaction error.

Analyzing Table 4, one can see that the results in Figures 5–7 are almost the same as we have already stated in Proposition 7. The only difference appears when the third-order splitting procedure is applied together with the second-order numerical method. The reason of this difference is that only integer values of \( s \) were considered in the numerical computations.

From these results we can deduce the following statements.

**Corollary 21** Let us apply an operator splitting procedure of order \( p \) together with a numerical method of order \( r \), and set \( h = \tau^s \) \( (s \geq 1) \). Then \( \varrho_{num} = \min \{p, rs\} \).

**Corollary 22** From Proposition 21 it follows that \( \varrho_{num} = p \) if \( rs = p \).

The above propositions are the analogues of Propositions 5 and 6 for the numerical orders.
We note that from Figures 5–7 it can be seen that $E_{\text{prac}}(\tau)$ behaves very similarly to $E_{\text{tot}}(\tau)$. Therefore, $E_{\text{prac}}(\tau)$ is a convenient quality in characterizing the total time-discretization method.

From these results one can see that the propositions stated in the fourth and the fifth sections are also valid for numerical experiments and for higher order numerical methods.

7 Conclusions

For solving differential equations, operator splitting procedures are usually applied together with numerical methods. Our aim was to achieve the order of this total time-discretization method to be equal to the order of the applied splitting procedure. In this case the total error of the numerical solution can be estimated.

In the present paper we showed that interaction error appeared in the numerical solution. It can be considered as the effect of the interaction between the splitting and the numerical errors: a certain type of global numerical error. Beside the above types of errors, the local practical error was defined. We showed that also the order of the local practical error characterizes the accuracy of the total time-discretization well. We deduced that the order of the total time-discretization equals the minimum of the order of the splitting, and the product of the order of the applied numerical method and the exponent which describes the variation of the numerical step size as a power function of the splitting timestep. This means that in practice there are two ways to achieve the order of the total time-discretization method to be equal to the order of the applied splitting procedure: we use numerical methods accurate enough, or we set the numerical step size small enough.

We completed our analytical results with numerical experiments. First, methods were introduced how to determine the order of the local errors, and then using these methods we could supplement the results also with higher order numerical methods.

Among our future plans there is the generalization of these results for unbounded operators, stiff, and non-linear problems, as well.
Acknowledgements

The authors kindly acknowledge the useful comments and suggestions of the unknown referees. This research was supported by the Hungarian Scientific Research Fund under the grant OTKA T043765, and NATO Collaborative Linkage Grant EST CLG 980505.

References


