Effectivity Analysis of Operator Splitting and the Average Method

Lívia Boda and István Faragó

Abstract In mathematics there are several problems which can be described by differential equations of a certain very complicated structure. Most of the time, we cannot produce the exact (analytical) solution of these problems, so we have to approximate them numerically by using some approximating method. In this paper we analyse one of such approximation methods, namely, operator splitting, which is a widely and successfully used method in numerical analysis. We introduce and demonstrate the method on a general Cauchy problem. In Sect. 1 of this paper we discuss the two most popular splitting methods, which are the sequential splitting (SS) and the Strang–Marchuk (SM) splitting, and describe the Average Method (AM) obtained by using splitting methods. Here we also discuss the possible reduction of the terms needed for the Average Method by using a matrix decomposition of pairwise commuting matrices.

In Sect. 2 we describe an aerodynamical model of flutter, which serves as our example problem. The advantage of the Average Method is shown in Sect. 3, where tables about runtimes and errors are given.

1 Operator Splitting and Average Method

We consider the following Cauchy-problem in $\mathbb{R}^m$:

$$\begin{cases} \dot{y}(t) = Ay(t) = \sum_{i=1}^{d} A_i y(t), \; t \in (0, T], \\ y(0) = y_0, \end{cases} \tag{1}$$

where $y\colon [0, T] \to \mathbb{R}^m$ is the unknown function, $y_0 \in \mathbb{R}^m$ is the given initial vector and $A_i \in \mathbb{R}^{m \times m}$ $(i = 1, \ldots, d)$ are given matrices.

L. Boda (✉) · I. Faragó
Department of Differential Equations, Budapest University of Technology and Economics, Budapest, Hungary
e-mail: boda.livia@ttk.bme.hu; istvan.farago@ttk.elte.hu

M. Ehrhardt, M. Günther (eds.), *Progress in Industrial Mathematics at ECMI 2021*, The European Consortium for Mathematics in Industry 39,
https://doi.org/10.1007/978-3-031-11818-0_6

The exact solution of the Cauchy-problem (1) can be written directly as $y(t) = \exp(tA)y(0)$. Since the exact representation of the exponential matrix $\exp(tA)$ is typically impossible (or, at least, a very time-consuming task), our aim is to approximate the exact solution numerically by some suitable approximation of this exponential matrix on the grid

$$\omega_h = \left\{t_n = n \cdot h, h = \frac{T}{N}, \ n = 0, 1, \ldots, N\right\}. \tag{2}$$

We can do it by the so-called operator splitting, which means the following. We decompose the original (complex) problem into a series of simpler Cauchy problems, linked through their initial conditions. By applying this method it can be easier to find a numerical solution to the original problem.

The two most popular splitting methods include the sequential splitting (SS) and the Strang-Marchuk (SM) splitting. The algorithm of sequential splitting in case of two subproblems is as follows. In this case the decomposition of A is $A = A_1 + A_2$. If we use the sequential splitting to solve (1) on the grid ω_h, it means that the following two subproblems are solved in every step:

$$\begin{cases} \dot{y}_1(t) = A_1 y_1(t), \ t \in (t_i, t_{i+1}], \\ y_1(t_i) = x_{sp}(t_i), \end{cases} \tag{3}$$

$$\begin{cases} \dot{y}_2(t) = A_2 y_2(t), \ \ t \in (t_i, t_{i+1}], \\ y_2(t_i) = y_1(t_{i+1}). \end{cases} \tag{4}$$

where $i = 0, \ldots, N-1$, $x_{sp}(t_{i+1}) = y_2(t_{i+1})$ and $x_{sp}(t_0) = y_0$.

The main difference between the sequential and Strang-Marchuk splitting is that the latter computes values in the midpoints of the subintervals. The algorithm of SM splitting means solving the following subproblems:

$$\begin{cases} \dot{y}_1(t) = A_1 y_1(t), \ t \in \left(t_i, t_{i+\frac{1}{2}}\right], \\ y_1\left(t_i\right) = x_{sp}(t_i), \end{cases} \tag{5}$$

$$\begin{cases} \dot{y}_2(t) = A_2 y_2(t), \quad t \in \left(t_i, t_{i+1}\right], \\ y_2\left(t_i\right) = y_1\left(t_{i+\frac{1}{2}}\right), \end{cases} \tag{6}$$

$$\begin{cases} \dot{y}_1(t) = A_1 y_1(t), \qquad t \in \left(t_{i+\frac{1}{2}}, t_{i+1}\right], \\ y_1\left(t_{i+\frac{1}{2}}\right) = y_2\left(t_{i+1}\right). \end{cases} \tag{7}$$

where $i = 0, \ldots, N-1$, $x_{sp}(t_{i+1}) = y_1(t_{i+1})$ and $x_{sp}(t_0) = y_0$.

Remark 1 The sequential splitting is a first-order of consistency method, the Strang-Marchuk splitting is a second-order of consistency method.

As an alternative to the classical splitting methods, we introduce the Average Method with sequential splitting (AM_{SS}) which is based on the idea to divide the

Cauchy problem (1) into d subproblems, using sequential splitting in all possible ordering sequences. Then we define the numerical solution of each split subproblem, taking their arithmetic mean, and we define the new numerical solution in ω_h.

Let $\mathcal{P}^d$ denote the set of the permutations of the indices $\{1, 2, \ldots, d\}$ and for $p = \{p_1, p_2, \ldots, p_d\} \in \mathcal{P}^d$ we introduce the notation

$$\exp\{p_1, p_2, \ldots, p_d\} = \exp(hA_{p_1})\exp(hA_{p_2}) \cdot \ldots \cdot \exp(hA_{p_d}). \tag{8}$$

Theorem 1 *Solving the Cauchy-problem* (1) *using sequential splitting for all possible permutations and then averaging the resulting numerical solutions yields a second-order method, i.e.*

$$\exp\big(h(A_1 + \ldots + A_d)\big) = \frac{1}{d!}\sum_{p\in\mathcal{P}^d} \exp\{p_1, p_2, \ldots, p_d\} + O(h^3). \tag{9}$$

So instead of using a second-order method once, we just use a first-order method more than once and we get a second-order numerical solution. Hence the main advantage is that a first-order method requires less computational demand than a second-order numerical method. However using the AM_{SS} method to solve Cauchy problem (1), we have to calculate $d!$ numerical solutions. Even with a relatively small value of d, we have to produce many numerical solutions and the computational demand may increase greatly.

If we find a decomposition for Cauchy problem (1) that includes commuting matrices, the number of subproblems can be significantly reduced. Let $A = A_1 + A_2 + \ldots + A_d$, and suppose that $\exists i, j \in \mathbb{N}$, and $i \neq j$ such that $[A_i, A_j] = 0$. Then instead of all the $d!$ permutations, we have $d! - (d-1)! = (d-1)(d-1)!$ elements. If the decomposition includes more commuting pairs of matrices, the reduction might be more significant. The other advantage of the Average Method is that the $d!$ numerical solutions can be independently calculated, i.e. the computation is parallelizable.

2 Application to the Aerodynamics

We investigated the efficiency of the Average Method on a physical problem which describes the aerodynamics of an airplane wing. The model is based on a wind tunnel experiment in which the lift force of an airplane wing was examined as a function of the inclination of the wing. This piecewise-linear model of flutter was investigated in [2]. Motivated by this model, we consider the 4-dimensional Cauchy problem

$$\begin{cases} \dot{\mathbf{x}}(t) = A_k\mathbf{x}(t), \\ \mathbf{x}(0) = \mathbf{x}_0. \end{cases} \tag{10}$$

Table 1 Parameters of the model

Parameter	c_0	c_1	c_2	p_1	p_2	p_3	p_4
Value	5.932	-6.846	2.662	0.1485	0.0147	0.0540	0.2748

where the affine model equations contain the three system matrices ($k = 0, 1, 2$)

$$A_k = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -(p_1 + p_2\mu c_k) & -\mu^2 c_k p_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & c_k\mu & -(p_4 - c_k\mu^2) & -p_3 \end{pmatrix},$$

with the model parameters given in Table 1, and $\mu \in (0, \infty)$ represents the nondimensional wind speed.

We analyzed several decompositions of matrix A_k, the most important of them being the following, which contains commuting matrices:

$$A_k = A_{k_{(1)}} + A_{k_{(2)}} + A_{k_{(3)}}, \tag{11}$$

where

$$A_{k_{(1)}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & -(p_1 + p_2\mu c_k) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_{k_{(2)}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -(p_4 - c_k\mu^2) & -p_3 \end{pmatrix},$$

$$A_{k_{(3)}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -\mu^2 c_k p_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & c_k\mu & 0 & 0 \end{pmatrix}.$$

Clearly, that matrices $A_{k_{(1)}}$ and $A_{k_{(2)}}$ are commuting matrices, therefore:

$$\exp\left(hA_{k_{(1)}}\right) \cdot \exp\left(hA_{k_{(2)}}\right) = \exp\big(h(A_{k_{(1)}} + A_{k_{(2)}})\big). \tag{12}$$

Then we introduce the notation

$$A_{k_{(4)}} = A_{k_{(1)}} + A_{k_{(2)}} = A_{k_{(2)}} + A_{k_{(1)}}. \tag{13}$$

Solving the Cauchy problem (1) using the AM_{SS} method with decomposition (11), which includes commuting matrices, and using property (12) and notation (13), the number of subproblems can be reduced from six to four, and we have to calculate the following four numerical solutions:

$$x_1(h) = \exp(hA_{k_{(1)}})\exp(hA_{k_{(3)}})\exp(hA_{k_{(2)}}) \cdot x_0, \tag{14}$$

$$x_2(h) = \exp(hA_{k_{(2)}})\exp(hA_{k_{(3)}})\exp(hA_{k_{(1)}}) \cdot x_0, \tag{15}$$

$$x_3(h) = \exp(hA_{k_{(4)}})\exp(hA_{k_{(3)}}) \cdot x_0, \tag{16}$$

$$x_4(h) = \exp(hA_{k_{(3)}})\exp(hA_{k_{(4)}}) \cdot x_0, \tag{17}$$

where (14), (15) have three subproblems, and in case of (16), (17) the number of subproblems was reduced from three to two which further simplifies the solution process and reduces computational demand, too. Then based on Theorem 1, we have

$$\exp\left(h(A_{k_{(1)}} + A_{k_{(2)}} + A_{k_{(3)}})\right) = \frac{x_1(h) + x_2(h) + 2x_3(h) + 2x_4(h)}{6} + \mathcal{O}(h^3). \tag{18}$$

3 Numerical Results

During the numerical implementation, the numerical solutions (14)–(17) were computed using sequential splitting, which has first order, and the subproblems were solved using the first-order explicit Euler method. Then averaging these first-order solutions we get a second-order numerical solution. The essence of the AM_{SS} method is that we have to implement some first-order approximating methods, which we can easily implement, then the average of first-order numerical solutions should be taken, which is not a very expensive operation, either, then we get a second-order method.

The numerical solutions (14)–(17) can be independently calculated, i.e. the computation is parallelizable. When we have four processors, we can compute the solutions (14)–(17) at the same time. We can simulate the parallel run as follows. Consider that we have four processors to calculate the numerical solutions in parallel. We can see the runtimes of every calculation of the solutions (14)–(17) in Table 2. We can calculate the whole runtime as follows: choose the maximum of the four runtimes (red coloured) and then add the runtime of the averaging. The last column of Table 2 shows the full runtime of the Average Method in case of four processors.

Now we consider the case where three processors are available to solve system (10) using the AM_{SS} method. In this case the main problem is to partition the subproblems well. On the one hand we saw in Sect. 2 that in case of solutions (14) and (15) there are three subproblems with matrices $A_{k_{(1)}}$, $A_{k_{(2)}}$ and $A_{k_{(3)}}$ while in case of solutions (16) and (17) there are only two subproblems with matrices $A_{k_{(3)}}$ and $A_{k_{(4)}}$. And on the other hand Table 2 shows that solutions (16) and (17) can be computed faster than solutions (14) and (15). Therefore, it is reasonable to partition

Table 2 Runtimes (in seconds) of the AM_{SS} during a parallel run with four processors

h	x_1	x_2	x_3	x_4	Average	Runtime
1	$7.20 \cdot 10^{-5}$	$6.12 \cdot 10^{-5}$	$2.10 \cdot 10^{-5}$	$2.22 \cdot 10^{-5}$	$1.02 \cdot 10^{-6}$	$7.30 \cdot 10^{-5}$
0.1	$8.24 \cdot 10^{-5}$	$8.90 \cdot 10^{-5}$	$7.56 \cdot 10^{-5}$	$7.14 \cdot 10^{-5}$	$2.21 \cdot 10^{-5}$	$1.11 \cdot 10^{-4}$
0.01	$8.18 \cdot 10^{-3}$	$7.52 \cdot 10^{-3}$	$1.71 \cdot 10^{-3}$	$2.03 \cdot 10^{-3}$	$5.62 \cdot 10^{-4}$	$8.74 \cdot 10^{-3}$
0.001	$4.02 \cdot 10^{-2}$	$3.89 \cdot 10^{-2}$	$1.37 \cdot 10^{-2}$	$1.45 \cdot 10^{-2}$	$9.72 \cdot 10^{-4}$	$4.11 \cdot 10^{-2}$

Table 3 Runtimes (in seconds) of AM_{SS} during a parallel run with three processors

h	x_1	x_2	x_3 and x_4	Average	Runtime
1	$7.20 \cdot 10^{-5}$	$6.12 \cdot 10^{-5}$	$4.32 \cdot 10^{-5}$	$4.02 \cdot 10^{-6}$	$7.30 \cdot 10^{-5}$
0.1	$8.24 \cdot 10^{-5}$	$8.90 \cdot 10^{-5}$	$1.47 \cdot 10^{-4}$	$2.21 \cdot 10^{-5}$	$1.69 \cdot 10^{-4}$
0.01	$8.18 \cdot 10^{-3}$	$7.52 \cdot 10^{-3}$	$3.74 \cdot 10^{-3}$	$5.62 \cdot 10^{-4}$	$8.74 \cdot 10^{-3}$
0.001	$4.02 \cdot 10^{-2}$	$3.89 \cdot 10^{-2}$	$2.82 \cdot 10^{-2}$	$9.72 \cdot 10^{-4}$	$4.11 \cdot 10^{-2}$

Table 4 Runtimes (in seconds) of AM_{SS} during a parallel run with two processors

h	x_1 and x_2	x_3 and x_4	Average	Runtime
1	$9.31 \cdot 10^{-5}$	$8.34 \cdot 10^{-5}$	$1.02 \cdot 10^{-6}$	$9.40 \cdot 10^{-5}$
0.1	$1.58 \cdot 10^{-4}$	$1.60 \cdot 10^{-4}$	$2.21 \cdot 10^{-5}$	$1.82 \cdot 10^{-4}$
0.01	$9.89 \cdot 10^{-3}$	$9.55 \cdot 10^{-3}$	$5.62 \cdot 10^{-4}$	$1.04 \cdot 10^{-2}$
0.001	$5.39 \cdot 10^{-2}$	$5.34 \cdot 10^{-2}$	$9.72 \cdot 10^{-4}$	$5.48 \cdot 10^{-2}$

as follows: solutions (14) and (15) are computed by two separate processors, and solutions (16) and (17) are computed one after the other by the third processor. Table 3 shows the runtimes of this case.

It is worth examining the case where we have two processors to compute the numerical solution of (10). Similarly to the three-processor case, proper partitioning will be the main task. The most reasonable partition is as follows: calculate solutions (14) and (16) one after the other with one processor, meanwhile solutions (15) and (17) can be calculated one after the other using the other processor. In this case Table 4 shows the runtimes.

And in order to see the practical usefulness of the AM_{SS} method, we solved the whole Cauchy-problem (10) without any splitting process using the improved Euler method, which is the same second-order method as the AM_{SS} method, and we compared the runtime of the AM_{SS} with two, three and four processors with the runtime of the improved Euler method. Table 5 shows this comparison and we can see that on average, the AM_{SS} method is one-two orders of magnitude faster than the improved Euler method.

Table 6 shows the comparison of errors in case of AM_{SS} and the improved Euler method. It can be seen the second-order convergence in both cases, furthermore the error is approximately the same in both cases.

Table 5 Comparison of runtimes (in seconds) in case of AM_{SS} with two, three and four processors and the improved Euler method

h	AM_{SS} + 2 proc.	AM_{SS} +3 proc.	AM_{SS} + 4 proc.	Improved Euler
1	$9.40 \cdot 10^{-5}$	$7.30 \cdot 10^{-5}$	$7.30 \cdot 10^{-5}$	$8.18 \cdot 10^{-3}$
0.1	$1.82 \cdot 10^{-4}$	$1.69 \cdot 10^{-4}$	$1.11 \cdot 10^{-4}$	$1.96 \cdot 10^{-2}$
0.01	$1.04 \cdot 10^{-2}$	$8.74 \cdot 10^{-3}$	$8.74 \cdot 10^{-3}$	$8.44 \cdot 10^{-2}$
0.001	$5.48 \cdot 10^{-2}$	$4.11 \cdot 10^{-2}$	$4.11 \cdot 10^{-2}$	$1.13 \cdot 10^{0}$

Table 6 Comparison of errors in case of AM_{SS} and the improved Euler method

h	AM_{SS} method	Improved Euler
1	$2.08 \cdot 10^{-2}$	$4.77 \cdot 10^{0}$
0.1	$2.44 \cdot 10^{-4}$	$2.04 \cdot 10^{-4}$
0.01	$2.44 \cdot 10^{-6}$	$2.01 \cdot 10^{-6}$
0.001	$2.44 \cdot 10^{-8}$	$2.01 \cdot 10^{-8}$

Acknowledgments This research has been supported by the National Research, Development and Innovation Office - NKFIH, grant no. K137699.

References

1. Faragó, I., Havasi, Á.: Operator splittings and their applications. In: Nova Science Publishers., (2009)
2. Kalmár-Nagy, T., Csikja, R., Elgohary, T.A.: Nonlinear analysis of a 2-dof piecewise linear aeroelastic system. Nonlinear Dynamics, vol. 85, no. 2, pp. 739–750 (2016)