International Journal of Parallel, Emergent and Distributed Systems
Vol. 00, No. 00, October 2006, 1–15

Computational Complexity of Weighted Splitting Schemes on Parallel Computers

P. CSOMÓS ¶ , I. DIMOV*‡ §, I. FARAGÓ¶, A. HAVASI† and TZ. OSTROMSKY‡

† Department of Meteorology, Eötvös Loránd University, Pázmány P. sétány 1/a, H-1117 Budapest, Hungary
‡ IPP, Bulgarian Academy of Sciences, Acad. G. Bonchev Str. 25 A, Sofia 1113, Bulgaria
§ ACET, The University of Reading, Whiteknights, PO Box 217, Reading, RG6 6AH, UK
¶ Department of Applied Analysis, Eötvös Loránd University, Pázmány P. sétány 1/c, H-1117 Budapest, Hungary

(received November 2005)

In models of complicated physical-chemical processes operator splitting is very often applied in order to achieve sufficient accuracy as well as efficiency of the numerical solution. The recently rediscovered weighted splitting schemes have the great advantage of being parallelizable on operator level, which allows us to reduce the computational time if parallel computers are used.

In this paper the computational times needed for the weighted splitting methods are studied in comparison with the sequential splitting and the Marchuk-Strang splitting and are illustrated by numerical experiments performed by use of simplified versions of the Danish Eulerian Model.

Keywords: Operator splitting; Computational complexity; Air pollution modeling; Distributed memory; Shared memory

1 Introduction

Huge systems of equations arise in several fields of applied mathematics. An example is modeling the long-range transport of air pollutants, where the number of ODE's to be solved can be as many as a few hundred thousand in each time-step of the numerical time integration. In such a case, in addition

* Corresponding author. Email: I.T.Dimov@reading.ac.uk

The International Journal of Parallel, Emergent and Distributed Systems
ISSN 1744-5760 print/ ISSN 1744-5779 online ©2005 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/1744-5760xxxxxxxxxxxx

to the requirement for sufficient accuracy, there is a great demand also for the minimization of the computational costs (efficiency). There is a need for optimizing the existing algorithms and developing new ones, in which parallel algorithms and their implementation on parallel computers play an increasingly significant role [1].

In models of such complex phenomena as air pollution transport the simultaneous effect of several different sub-processes has to be described. The direct numerical treatment of such complicated problems is very difficult, therefore operator splitting is usually applied. The main point in operator splitting is the replacement of the original model with one in which the sub-processes take place successively in each time-step. This allows us to solve a few simpler sub-problems instead of the original one (for more details see [8–10]).

Numerous splitting schemes have been constructed and used in practice. These have been analyzed in several papers from the viewpoint of accuracy (in terms of the local splitting error, see for example [4,5,8]). In this paper the computational properties of four selected splitting procedures are compared and conditions are formulated which can help programmers choose the optimal splitting scheme when huge systems of equations are to be solved by applying operator splitting.

The paper is organized as follows. In Section 2 the splitting schemes selected for our study are introduced. In Section 3 the computational complexity for these splitting methods is estimated. Finally, in Section 4, some numerical results, showing the computational costs of these splitting methods when applied to two simplified versions of the Danish Eulerian Model [16,17] are presented. The results are from sequential as well as from parallel runs (with shared and distributed memory parallelization on the operator level) where appropriate. In Section 5 the practicality of our approach as well as the limitations to its parallel implementation (together with those of the previously implemented optional levels of parallelism) are discussed. Some conclusions are drawn in Section 6.

2 Description of the splitting schemes

In this section four basic splitting methods are illustrated in a model problem with two operators. These methods are as follows:

- the sequential splitting,
- the Marchuk-Strang splitting,

- the symmetrically weighted sequential splitting,
- the symmetrically weighted Marchuk-Strang splitting.

Consider the following initial value problem

$$\left.\begin{aligned} \frac{dw(t)}{dt} &= Aw(t), \quad t \in (0,T] \\ w(0) &= w_0, \end{aligned}\right\}$$

and assume that the operator A can be presented as a sum of two operators, i.e., $A = A_1 + A_2$. Let us divide the time interval $[0,T]$ into $n \in I\!\!N$ subintervals of length $\tau = T/n$, the so-called splitting time-step.

2.1. The **sequential splitting** is the simplest and most natural splitting scheme, determined by the following systems of equations:

$$\left.\begin{aligned} \frac{dw_k^{(1)}(t)}{dt} &= A_1 w_k^{(1)}(t), \quad t \in ((k-1)\tau, k\tau] \\ w_k^{(1)}((k-1)\tau) &= w_{k-1}^{(2)}((k-1)\tau) \end{aligned}\right\},$$

$$\left.\begin{aligned} \frac{dw_k^{(2)}(t)}{dt} &= A_2 w_k^{(2)}(t), \quad t \in ((k-1)\tau, k\tau] \\ w_k^{(2)}((k-1)\tau) &= w_k^{(1)}(k\tau) \end{aligned}\right\}, \tag{1}$$

for $k = 1, 2, \ldots, n$, where $w_0^{(2)}(0) = w_0$. (The superindices in parentheses identify the number of the sub-problem in the splitting.) So, as a first step we solve the system with operator A_1 using the initial condition of the original problem and then, applying the obtained solution at time τ as an initial condition, we solve the system with operator A_2. This procedure is then performed cyclicly. The method can be extended to more than two operators in a natural way.

The sequential splitting has a first order local splitting error, which means that ignoring the other sources of error (inexact data, rounding, etc.) the difference between the solution of the original problem and the sequence of problems (2)–(1) at the end of the time interval is $\mathcal{O}(\tau)$ (the splitting error at each time-step is $\mathcal{O}(\tau^2)$).

This algorithm is not parallelizable on the operator level, because each step needs the solution of the previous step as an initial value. However, each step may be parallelizable itself (which is usually the case), so in general parallelization on lower level is possible. Parallelism of that kind is not considered in this paper, we will rather concentrate on the operator-level parallelism (sometimes both levels can be used together on a big parallel system in order to exploit its full capacity). Examples of efficient parallelization of a large air-pollution model that uses sequential splitting (where each operator is parallelized separately) can be found in [6,12,13,16,17].

2.2. Marchuk-Strang splitting. This is a more accurate (second order – $\mathcal{O}(\tau^2)$) method, defined by the following sequence of problems:

$$\left.\begin{aligned} \frac{dw_k^{(1)}(t)}{dt} &= A_1 w_k^{(1)}(t), \quad t \in ((k-1)\tau, (k-\tfrac{1}{2})\tau] \\ w_k^{(1)}((k-1)\tau) &= w_{k-1}^{(3)}((k-1)\tau) \end{aligned}\right\}, \tag{2}$$

$$\left.\begin{aligned} \frac{dw_k^{(2)}(t)}{dt} &= A_2 w_k^{(2)}(t), \quad t \in ((k-1)\tau, k\tau] \\ w_k^{(2)}((k-1)\tau) &= w_k^{(1)}((k-\tfrac{1}{2})\tau) \end{aligned}\right\}, \tag{3}$$

$$\left.\begin{aligned} \frac{dw_k^{(3)}(t)}{dt} &= A_1 w_k^{(3)}(t), \quad t \in ((k-\tfrac{1}{2})\tau, k\tau] \\ w_k^{(3)}((k-\tfrac{1}{2})\tau) &= w_k^{(2)}(k\tau) \end{aligned}\right\}, \tag{4}$$

for $k = 1, 2, \ldots, n$, where $w_0^{(3)} = w_0$. This scheme is non-symmetric with respect to A_1 and A_2: in each time-step we begin and end the computations with operator A_1 (two steps with a distance $\frac{\tau}{2}$), putting A_2 in the middle (just one step with a distance τ). The method was proposed by Marchuk [11] and Strang [15] independently in 1968. It is easy to see that the number of operations performed in the Marchuk-Strang splitting is greater than that in the sequential scheme. Similarly to the sequential splitting, the Marchuk-Strang splitting cannot be parallelized on operator level.

If three operators are present, then we can apply Marchuk-Strang splitting by some symmetric ordering of the operators, e.g. $A_1A_2A_3A_2A_1$, stepping τ with the middle operator A_3 and $\frac{\tau}{2}$ with the others. For the sake of efficiency the computationally most expensive operator is usually put in the middle.

2.3. Symmetrically weighted sequential splitting. An alternative of the above splitting techniques can be obtained by symmetrizing the sequential splitting in the following way. In each time-step we apply sequential splitting both in the order $A_1 \to A_2$ and $A_2 \to A_1$ independently, and at the end of the time-steps we combine both solutions by taking a weighted average of the results. So, in the k-th time-step we compute the solution $u_k^{(2)}(k\tau)$ of the

$A_1 \to A_2$ splitting

$$\left.\begin{aligned}\frac{du_k^{(1)}(t)}{dt} &= A_1 u_k^{(1)}(t), \quad t \in ((k-1)\tau, k\tau] \\ u_k^{(1)}((k-1)\tau) &= \hat{w}((k-1)\tau)\end{aligned}\right\}, \tag{5}$$

$$\left.\begin{aligned}\frac{du_k^{(2)}(t)}{dt} &= A_2 u_k^{(2)}(t), \quad t \in ((k-1)\tau, k\tau] \\ u_k^{(2)}((k-1)\tau) &= u_k^{(1)}(k\tau)\end{aligned}\right\}, \tag{6}$$

and the solution $v_k^{(2)}(k\tau)$ of the $A_2 \to A_1$ splitting

$$\left.\begin{aligned}\frac{dv_k^{(1)}(t)}{dt} &=A_2 v_k^{(1)}(t), \quad t \in ((k-1)\tau, k\tau] \\ v_k^{(1)}((k-1)\tau) &= \hat{w}((k-1)\tau)\end{aligned}\right\}, \tag{7}$$

$$\left.\begin{aligned}\frac{dv_k^{(2)}(t)}{dt} &= A_1 v_k^{(2)}(t), \quad t \in ((k-1)\tau, k\tau] \\ v_k^{(2)}((k-1)\tau) &= v_k^{(1)}(k\tau)\end{aligned}\right\}, \tag{8}$$

where $\hat{w}(0) = w_0$, and then compute a weighted average of the solutions as

$$\hat{w}(k\tau) = \Theta u_k^{(2)}(k\tau) + (1-\Theta) v_k^{(2)}(k\tau), \tag{9}$$

where $\Theta \in (0,1)$ is some weight parameter. The scheme belonging to $\Theta = \frac{1}{2}$ has been introduced by Strang [14] and is called symmetrically weighted sequential splitting. It has also second order ($\mathcal{O}(\tau^2)$) local error [3]. Unlike the sequential and Marchuk-Strang splittings, this scheme is inherently parallel, because the computations of the sequential splittings (5)–(6) and (7)–(8) are independent of each other [2].

When there are three operators we can order them in more than two ways. Any two can be chosen, their solutions are weighted symmetrically and the average is taken for the next time-step.

2.4. Weighted Marchuk-Strang splitting. This splitting scheme is defined by computing a weighted average of the solutions obtained by applying Marchuk-Strang splitting by two different orderings of the operators ($A_1 A_2 A_1 - A_2 A_1 A_2$). This is again a parallelizable method, but the number of operations to perform is greater. The local error of this scheme is of $\mathcal{O}(\tau^2)$ by any value of the weight parameter, see [7].

If there are three operators, then we can order them in several different ways. The computationally most expensive operator is usually put in the middle and the other two are altered in order to get the least expensive (and computationally balanced) scheme. The two solutions are weighted symmetrically and the average is taken for the next time-step.

We should mention that in the cases of weighted sequential splitting and weighted Marchuk-Strang splitting the accuracy can be of even higher than second order if certain conditions are imposed to the operators, see [3, 7]. However, these conditions are rather strict and are not satisfied in the model problems studied in this paper. The numerical methods applied to solve the ODE systems are not discussed in this paper and the numerical errors are not taken into account.

3 Computational complexity estimations

In this section we analyze the computational times for the different splitting procedures introduced in Section 2. Two- and three-operator schemes are considered. Tables 1 and 2 allow us to make a full comparison of all schemes under consideration from the viewpoint of efficiency. The run-time differences for different splitting schemes can be seen from these tables (see column "Computational time").

Splitting scheme	**Proc.**	**Computational time**
Sequential	1	$T_1 + T_2$
Marchuk-Strang ($A_1A_2A_1$)	1	$2T_1 + T_2$
Weighted sequential	1	$2(T_1 + T_2) + T_a$
	2-SM	$q(T_1 + T_2) + T_a$
	2-DM	$T_1 + T_2 + T_c + T_a$
Weighted Marchuk-Strang ($A_1A_2A_1 - A_2A_1A_2$)	1	$3(T_1 + T_2) + T_a$
	2-SM	$q\max\{2T_1 + T_2, T_1 + 2T_2\} + T_a$
	2-DM	$\max\{2T_1 + T_2, T_1 + 2T_2\} + T_c + T_a$

Table 1. Computational times needed for the splitting schemes under consideration with two operators. The number of processes and the type of parallelization (if present) is given in the middle column: 1 – one process (sequential), 2 – two parallel processes, SM – shared memory, DM – distributed memory. $q > 1$ is a slowdown parameter of the shared memory due to memory conflicts.

Let the computational time for the sub-problem A_i be denoted by T_i and T_a be the time for averaging in the weighted splittings. For simplicity, in our estimations we assume that in case of parallel implementation T_1 and T_2 are the same for all the processors, which is not exactly so, see the numerical experiments in Section 4. Table 1 contains the computational times for the different splitting schemes described in the previous section. For the two weighted splitting algorithms the times for parallel execution are also given in the table. Two parallel processes (distributed or shared memory) are only used, enough to exploit the operator-level parallelism. Possible parallelization on a lower level is not considered in this study. The memory can be either distributed or shared. In the distributed memory case each process uses only its own memory during the computations. Separate communication procedures are used in order to exchange data between the two processes when necessary (in our case these are called usually before and after the averaging). The communication time (denoted by T_c) is pretty much the same in all weighted schemes, as the amount of data exchanged is the same. If the memory is shared, then both processes can access the entire memory. There are no explicit communication procedures, the data is exchanged via the shared memory. This means that some memory conflicts can arise at any time throughout the parallel computations, which results in certain slowdown of the computational steps, performed in parallel. For the sake of simplicity we assume that one and the same slowdown factor $q > 1$ applies to the times of all computational steps in the schemes with shared memory parallelization. Note that the factor q does not apply to the averaging time T_a as that stage is not performed in parallel. We should mention that the factor q depends on the particular computer used and on the quality of the parallel implementation. It can vary within broad bounds in practice. A smaller (close to 1) value of q indicates an efficient shared memory implementation. In the numerical examples, given in the next section, it appears that $q \approx 1.2$.

Let us emphasize on the following facts:

- The time need of the parallelized weighted sequential splitting is always greater than that of the sequential splitting; the difference is

  $$(q-1)(T_1+T_2)+T_a \quad \text{if shared memory is used,}$$

  $$\text{and} \qquad T_c+T_a \quad \text{if distributed memory is used.}$$

  However, at the cost of the greater time required we can expect a more accurate solution.
- The parallelized weighted sequential splitting takes less time than the Marchuk-Strang splitting if

  $$q(T_1+T_2)+T_a < 2T_1+T_2 \quad \text{in case of shared memory}$$

$$\text{and if} \qquad T_c + T_a < T_1 \qquad \text{in case of distributed memory.}$$

- The time need of the parallelized weighted Marchuk-Strang scheme is always greater than that of the traditional Marchuk-Strang scheme; the extra time required is

$$q(\max\{2T_1 + T_2, T_1 + 2T_2\}) - 2T_1 - T_2 + T_a \qquad \text{in case of shared memory}$$

$$\text{and} \qquad \max\{T_1, T_2\} - T_1 + T_c + T_a \qquad \text{in case of distributed memory.}$$

- It is more advantageous to use shared memory instead of distributed memory for the weighted sequential splitting if the following relation is satisfied:

$$(q-1)(T_1 + T_2) < T_c$$

- Similarly, for the weighted Marchuk-Strang splitting it is more advantageous to use shared instead of distributed memory if the following relation is satisfied:

$$(q-1)\max\{2T_1 + T_2, T_1 + 2T_2\} < T_c$$

Splitting scheme	**Proc.**	**Computational time**
Sequential	1	$T_1 + T_2 + T_3$
Marchuk-Strang ($A_1A_2A_3A_2A_1$)	1	$2(T_1 + T_2) + T_3$
Weighted sequential	1	$2(T_1 + T_2 + T_3) + T_a$
	2-SM	$q(T_1 + T_2 + T_3) + T_a$
	2-DM	$T_1 + T_2 + T_3 + T_c + T_a$
Weighted Marchuk-Strang	1	$4(T_1 + T_2) + 2T_3 + T_a$
	2-SM	$2q(T_1 + T_2) + pT_3 + T_a$
($A_1A_2A_3A_2A_1 - A_2A_1A_3A_1A_2$)	2-DM	$2(T_1 + T_2) + T_3 + T_c + T_a$

Table 2. Computing times needed for the splitting schemes under consideration with three operators. The number of processors and the type of parallelization (if present) is given in the middle column: 1 – one processor (sequential), 2 – two parallel processors, SM – shared memory, DM – distributed memory. $q > 1$ is a slowdown parameter of the shared memory due to memory conflicts.

The corresponding times for these splitting schemes in case of three operators are given in Table 2. The possible operator-level parallelizations are also included.

Comparing the times, given in the table, we can conclude that the weighted sequential splitting (which is always more expensive than the similarly accurate Marchuk-Strang splitting in sequential runs) is faster than Marchuk-Strang if

$q(T_1+T_2+T_3)+T_a < 2(T_1+T_2)+T_3$ on two processors with shared memory

and if $T_a+T_c < T_1+T_2$ on two processors with distributed memory.

4 Numerical experiments

In order to show the differences between the splitting schemes presented in Section 2 (in terms of computational time and parallelization abilities), all of them were implemented into some simplified versions of the Danish Eulerian Model (DEM). This model has been created for studying long-range transport of air pollutants over the European region. The model is described in detail in [16].

All experiments are performed on a Linux cluster of four nodes MAC PowerPC. Each node is a shared memory computer with two processors type G4. However, each node has its private memory, i.e., the memory is distributed on node level. This makes it possible to use both shared and distributed memory programming models on the cluster. Standard parallelization tools (MPI and OpenMP) are used for the distributed memory and the shared memory parallel implementations, respectively.

The DEM is able to treat numerically all important sub-processes of the air pollution transport: advection, diffusion, emission, chemical reactions and deposition. In the computer program of the model the advection is treated together with the diffusion and the emission together with the chemistry. There is also a three-dimensional version of the model, in which the atmosphere is stratified into several layers and the vertical transport of the pollutants (including the sub-processes of convection and vertical diffusion) is also taken into account. In the first group of experiments, presented in Tables 3 and 4, a simplified two-dimensional version of DEM is used with only two subproblems present: the advection-diffusion subproblem and the emission subproblem. These are to illustrate the two-operator splitting schemes. In Tables 5 and 6 the three-operator splitting schemes are presented by using the corresponding three-dimensional version of DEM (with the vertical transport subproblem switched on). For the sake of simplicity only one chemical species was taken into account. The numbers in all tables of this section are CPU times in seconds, averaged over several runs and rounded to the first digit after the decimal point. In the experiments below $\Theta = 0.5$ has been used for the weighted splitting schemes. This choice of Θ affects only the accuracy, so very similar computational times will be obtained with any other value of Θ.

Stage	S	MSt-AEA	MSt-EAE	W	WMSt
Data input	0.3	0.3	0.4	0.3	0.3
Parameter setup	6.8	7.0	6.9	7.5	7.1
Advection (T_1)	42.1	86.7	43.3	86.5	130.3
Emission (T_2)	1.3	1.3	2.7	2.6	4.0
Data output	18.3	18.0	18.1	18.2	18.2
Total	**68.8**	**113.3**	**71.3**	**117.4**	**162.6**

Table 3. Sequential computational times (in seconds) for different splitting schemes applied to the advection-emission problem (S – sequential splitting, MSt – Marchuk-Strang splitting, A – advection and diffusion, E – emission, W – weighted sequential splitting, WMSt – weighted Marchuk-Strang splitting).

Table 3 contains the sequential execution times of the main stages and the total times of the sequential (S), Marchuk-Strang (MSt), weighted sequential (W) and weighted Marchuk-Strang (WMSt) splitting schemes, applied to the simplified two-dimensional version of DEM. Each row corresponds to one of the stages as follows. *Data input* refers to the reading of input data (such as emission and meteorological data fields) from the corresponding input files. *Parameter setup* includes the initialization of some parameters (chemistry, emission, deposition and diffusion coefficients; pointers to certain work arrays; initial values of the concentrations; etc). The *Advection* and *Emission* rows contain the accumulated times for the solutions of the advection-diffusion and the emission subproblems on each time-step, as required by the corresponding splitting scheme. The time for writing the results in files is given in the *Data output* row. The times for data input, parameter setup and data output are obviously independent of the applied splitting procedure. The small differences between the values in these rows is a result of random factors. As seen from Table 3, the weighted sequential splitting takes longer time than the Marchuk-Strang splitting even if the latter one starts and ends with the longer-working advection-diffusion operator, which is not usually the case in practice. The Marchuk-Strang scheme starting with emission takes about 38 seconds (37%) less time, and so hardly more than the sequential splitting. It is also clear from this table that the weighted sequential splitting is not worth using if all computations are performed sequentially.

Table 4 contains the parallel execution times of the above problem for the parallelizable on operator level splitting schemes. Each submodel of DEM has efficient lower level parallelization, which is beyond the scope of this paper (see for details [6,12,13,17]). The lower level parallelization is not used in our

	Weighted sequential		**Weighted MSt**	
Stage	Process 1	Process 2	Pr.1 - AEA	Pr.2 - EAE
Data input	0.4	0.4	0.4	0.4
Parameter setup	8.0	7.0	7.9	7.0
Advection (T_1)	44.8	43.9	87.3	42.0
Emission (T_2)	1.5	1.7	1.3	2.4
Data output	19.8	18.8	19.9	19.4
Communication (T_c)	28.1	48.3	27.8	47.8
Averaging (T_a)	0.6	0.4	0.5	0.5
Total	**122.2**		**156.6**	

Table 4. Parallel execution times on two proc. (distributed memory) for the weighted sequential and weighted Marchuk-Strang splitting schemes applied to the advection-emission problem (A – advection and diffusion, E – emission, MSt – Marchuk-Strang splitting)

experiments. It should be mentioned, however, that the two levels of parallelism are not exclusive, they can be used together on a huge parallel system in order to exploit better its capacity. Results of parallelization on two distributed memory processes are presented in Table 4. The *Communication* time given there includes also some idle time (when one of the processes has to wait for the other to respond in order to complete certain communication procedure). That is why it is different for the two processes within one and the same task (the weighted sequential scheme or the weighted Marchuk-Strang scheme). The total time, however, is always equal for both processes. It should be higher than or at least equal to the sum of the stage times of the either process in the task.

In the experiment, presented in Table 4, the communication times are rather long. The communication stage in Process 2 of the weighted sequential splitting takes more than the average time needed for the advection stage. This is the reason why the parallelization of the weighted sequential splitting is not advantageous in this example (its running time exceeds that of the sequentially computed weighted sequential splitting if distributed memory is used). In the case with the weighted Marchuk-Strang splitting the total computational time is reduced by 6 seconds (3.7%). This is either not a practical advantage of the parallelization. Despite the large communication time, another problem here is the big difference between the times of Advection and Emission stages, which makes the two processes rather imbalanced. This is an inherent disadvantage of the weighted Marchuk-Strang splitting in case of two operators with different costs.

A more complex (but still simplified version of the air pollution model, without the non-linear chemistry submodel) is obtained by including the vertical exchange subproblem (describing the convection and the vertical diffusion

Stage	MSt-EVAVE	MSt-AVEVA
Data input	0.4	0.4
Parameter setup	58.4	61.1
Advection (T_1)	878.6	1021.3
Emission (T_2)	26.9	17.4
Vertical exchange (T_3)	844.5	938.8
Data output	169.2	186.4
Total	**1558.7**	**2225.4**

Table 5. Sequential execution times for two different Marchuk-Strang splitting schemes, applied to the three-dimensional simplified version of DEM (A – advection and diffusion, V – vertical exchange, E – emission, MSt – Marchuk-Strang splitting).

Stage	**Weighted sequential splitting**			
	Distributed mem.		**Shared memory**	
	Process 1	Process 2	Thread 1	Thread 2
Data input	0.4	0.4	0.4	0.4
Parameter setup	65.5	64.4	71.8	67.6
Advection (T_1)	446.6	436.6	525.6	533.0
Emission (T_2)	21.6	14.0	27.8	15.1
Vertical exchange (T_3)	412.3	411.9	487.8	481.6
Data output	172.3	170.3	192.7	260.2
Communication (T_c)	440.5	472.8	—	—
Averaging (T_a)	13.8	4.3	13.9	6.5
Total	**1580.1**		**1410.5**	

Table 6. Execution times for the distributed memory and shared memory parallel models (on two processors) of the weighted sequential splitting (EVA - AVE, A – advection and diffusion, V – vertical exchange, E – emission) applied to the three-dimensional simplified version of DEM.

processes). This model is three-dimensional and contains ten layers in vertical direction, which makes the computations about ten times more expensive than in the above two-dimensional version.

Table 5 shows the sequential computational times obtained by applying Marchuk-Strang splitting schemes with two different orders of the three subproblems and Table 6 – the parallel computational times of the weighted splitting scheme, obtained on two processors by using either distributed or shared memory parallelization techniques.

It is seen that the weighted sequential splitting is more advantageous than the Marchuk-Strang splitting if shared memory is used (we can gain almost 150 seconds (9.5%)), while for distributed memory the results for the weighted sequential splitting are somewhat worse than but comparable with those obtained by applying Marchuk-Strang splitting.

5 Practicality and limitations of our parallelization approach

Splitting approach is commonly used in the solution of large and complex PDE systems as those arising in environmental modeling. Development of new highly accurate splitting algorithms that can be performed efficiently on the up-to-date parallel supercomputers is an important task in the efficient numerical solution of such problems. Such algorithms have been discussed, implemented and tested in this paper. We have applied them to a particular large-scale air pollution model - DEM, in order to examine their practical accuracy and parallel properties. The DEM itself is optimized to work efficiently on a wide spectrum of parallel machines [6, 12, 13, 17], so each potential drawback in the performance of the proposed algorithms would easily show up. There are four distinct levels of parallelism in the current version of DEM, listed below (from finer to coarser):

(i) **instruction level parallelism** (called sometimes software pipelining), is strongly dependent on the hardware type. It is achieved automatically by the compiler if certain optimization level is enabled;

(ii) **loop level parallelism** - either shared memory or virtual shared memory via OpenMP is necessary;

(iii) **algorithm level parallelism** - MPI is mandatory for this level;

(iv) **splitting method parallelism**, achieved also by MPI.

In this paper we pay attention only to the last (coarsest) level of parallelism that can be applied if an appropriate (parallelizable) splitting method is used. The other levels are not considered here. More details on the two lower levels of parallelism with a lot of experimental results can be found in [6, 12, 13]. The numerical experiments here are meant to show just the potential of the splitting method parallelism. They prove the quality of the proposed splitting algorithms and their place in the model. We can expect similar results if these splitting techniques are applied to any other environmental, or even large-scale computational model, based on similar PDE systems.

6 Conclusions

We compared the computational complexity of the traditional splitting methods (sequential and Marchuk-Strang splitting) with those of the corresponding weighted splitting methods by using three different ways of computation: (i) sequential run, (ii) parallel run by using shared memory, (iii) parallel run by using distributed memory. Numerical experiments were performed in simplified 2D and 3D versions of the Danish Eulerian Model, however, we expect qualitatively similar results for other model problems, too. Only operator-level parallelism was used in both (ii) and (iii), lower level parallelization possibilities were not considered.

All experiments were performed on a Linux cluster of four nodes MAC PowerPC. (We remark that the results may be different if we use other computers.) The computational times of the different schemes depended very much on the way of computation.

Our main conclusions with respect to the results, presented in Section 4, can be summarized as follows.

- The weighted splittings are not worth using if all computations are performed sequentially.
- If distributed memory is used and the communication between the processors takes too long time, it is not advantageous to parallelize the weighted methods.
- If the memory conflicts do not slow down very much the computations when shared memory is used, weighted sequential splitting becomes more efficient than Marchuk-Strang splitting. We remark that in our case (two processors only) the slowdown is usually not yet significant.
- In the weighted Marchuk-Strang splitting the parallelized computation may be inefficient if the costs of the two operators differ too much.

Acknowledgements

The experimental results were obtained during the visit of P. Csomós and Á. Havasi in the Institute for Parallel Processing, Bulgarian Academy of Sciences, Sofia within the framework of the project Center of Excellence (grant ICA1-CT-2000-70016). The work was also supported in part by EC grant $BIS21^{++}$, as well as by the Bulgarian Academy of Sciences and Hungarian Academy of Sciences (contract No. 6/2003). Ágnes Havasi is a grantee of the Bolyai János Scholarship.

References

[1] Alexandrov, V. N., Owczarcz, W. Thomsen, P.G. and Zlatev, Z., 2004, Parallel runs of a large air air pollution model on a grid of SUN computer, *Math. Comp. in Simul.,* **65**, 557-577.

[2] Bochev, M., Faragó, I., Havasi, Á., 2004, Testing weighted splitting schemes on a one-column transport-chemistry model, *Int. J. Environment and Pollution*, **22**, 3–16.

[3] Csomós, P., Faragó, I. and Havasi, Á., 2005, Weighted sequential splittings and their analysis, *Comp. Math. Appl.*, **50**, 1017–1031.

[4] Dimov, I., Faragó, I., Havasi, Á. and Zlatev, Z., 2001, L-commutativity of the operators in splitting methods for air pollution models, *Annal. Univ. Sci. Sec. Math.*, **44**, 127–148.

[5] Dimov, I., Faragó, I., Havasi, Á. and Zlatev, Z., 2004, Operator splitting and commutativity analysis in the Danish Eulerian Model, *Math. Comp. Sim.* **67**, 217-233.

[6] Dimov, I., Ostromsky, Tz. and Zlatev, Z., 2005, Challenges in using splitting techniques for large-scale environmental modeling, in: *Advances in Air Pollution Modeling for Environmental Security (Faragó, I., Georgiev, K., Havasi, Á. - eds.) NATO Science Series*, **54**, Springer, 115–132.

[7] Faragó, I. and Havasi, Á., 2005, On the convergence and local splitting error of different splitting schemes, *Progress in Computational Fluid Dynamics*, **5**, 495–504.

[8] Havasi, Á., Bartholy, J. and Faragó, I., 2001, Splitting method and its application in air pollution modeling, *Időjárás,* **105**, No. 1, 39–58.

[9] Hundsdorfer, W.H. and Verwer, J., 2003, Numerical solution of time-dependent advection-diffusion-reaction equations, Springer, Berlin.

[10] Lanser, A. and Verwer J.G., 1999, Analysis of Operator Splitting for Advection-Diffusion-Reaction Problems from Air Pollution Modeling, *J. Comput. Appl. Math.,* **111**, 201–216.

[11] Marchuk, G. I., 1968. Some application of splitting-up methods to the solution of mathematical physics problems. *Applik. Mat.,* **13(2)**, 103–132.

[12] Ostromsky, Tz. and Zlatev, Z., 2001, Parallel Implementation of a Large-scale 3-D Air Pollution Model, *Large Scale Scientific Computing (Margenov, S., Wasniewski, J., Yalamov, P. - eds.), LNCS-2179*, Springer, 309–316.

[13] Ostromsky, Tz. and Zlatev, Z., 2002, Flexible Two-level Parallel Implementations of a Large Air Pollution Model, *Numerical Methods and Applications (I.Dimov, I.Lirkov, S. Margenov, Z. Zlatev - eds.), LNCS-2542*, Springer, 541–550.

[14] Strang, G., 1963, Accurate partial difference methods I: Linear Cauchy problems. *Archive for Rational Mechanics and Analysis*, **12**, 392–402.

[15] Strang, G., 1968, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.*, **5**, No. 3, 506–517.

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

[17] Zlatev, Z., Dimov, I. and Georgiev, K., 1996, Three-dimensional version of the Danish Eulerian Model, *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM),* **76**, S4, 473–476.