Additional degrees of parallelism within the Adomian decomposition method

Andreas Schmitt[1,2], Martin Schreiber[3], and Michael Schäfer[1,2]

[1] TU Darmstadt, Graduate School of Computational Engineering, Dolivostr. 15,
D-64293 Darmstadt, Germany,
aschmitt@gsc.tu-darmstadt.de,
WWW home page: http://www.graduate-school-ce.de

[2] TU Darmstadt, Institute of Numerical Methods in Mechanical Engineering,
Dolivostr. 15, D-64293 Darmstadt, Germany

[3] University of Exeter, Mathematics / Computer Science, Harrison Building,
North Park Road, Exeter, EX4 4QF, United Kingdom

Abstract. The trend of future massively parallel computer architectures challenges the exploration of additional degrees of parallelism also in the time dimension when solving continuum mechanical partial differential equations. The Adomian decomposition method (ADM) is investigated to this respect in the present work. This is accomplished by comparison with the Runge-Kutta (RK) time integration and put in the context of the viscous Burgers equation.
Our studies show that both methods have similar restrictions regarding their maximal time step size. Increasing the order of the schemes leads to larger errors for the ADM compared to RK. However, we also discuss a parallelization within the ADM, reducing its runtime complexity from $O(n^2)$ to $O(n)$. This indicates the possibility to make it a viable competitor to RK, as fewer function evaluations have to be done in serial, if a high order method is desired. Additionally, creating ADM schemes of high-order is less complex than it is with RK.

Keywords: Adomian decomposition, Burgers' equation, Runge-Kutta, parallel in time

1 Introduction

Simulations play an ever increasing role in science and industry. They are necessary, for instance, to simulate systems which cannot be studied experimentally such as planetary movements, to predict the weather of the next days, or to simulate molecular interactions for development of pharmaceutics. Very often, hard time constraints require to get simulation results within a reasonable time frame. Weather simulation is probably one of the most obvious examples. Additionally, driven by the stagnation in processor speed and the increasing degrees of on-chip parallelization over the last decade, exploring additional degrees of parallelism in the time domain is of steadily increasing interest. This is one of the main driving factors for parallel-in-time methods, see [22] for a review and overview.

Frequent choices of time stepping schemes from the family of the Runge-Kutta methods are the explicit, implicit, or implicit-explicit form [9, 19]. The application of these time stepping methods was already studied intensively. A less used and investigated method is the Adomian decomposition method (ADM) [6]. So far, the ADM was only considered as a time stepping method, but to our best knowledge no investigation was done regarding its properties of extracting additional degrees of parallelism. This motivates the present work in which we investigate the ADM in the context of parallelization and compare it to the explicit Runge-Kutta (RK) method for the viscous Burgers equation. Here, the focus lies on both the accuracy, which can be gained, and the time step restrictions.

1.1 Related Work

Adomian himself used the decomposition to calculate the exact solution for a specific initial velocity distribution [7]. Many authors approximated the exact solution by applying the Adomian decomposition using a truncated series instead of the originally used infinite series. Their work [8, 20, 27, 28, 32] shows that taking only a few terms of the series into account yields highly accurate results for the Burgers' equation. In [15, 21, 26] it is noted that truncating the series yields a small convergence radius, such that the maximal time step is bounded. Applying the ADM to an equation which is discretized in space reduces the radius even further [32].

The problem with the convergence radius was circumvented in [2] by using the ADM as a time stepping scheme with a stable time step size. Applying the ADM each time step makes spatial derivatives necessary for each interim solution. A reduction of complexity was used by approximating the interim solutions by an easily derivable series. With this method a good approximation of the exact solution was found. In [35] it is shown that the ADM is also able to yield very accurate results when the spatial domain is discretized and the ADM is used as a numerical time stepping scheme.

This work focuses on a comparison of the ADM with the RK method which was already done for other models. These comparisons were carried out numerically with a continuous spatial domain. For the Lorenz equation it was shown in [24] that the ADM with four terms of the series allows for larger time steps than the classical RK method and reaching the same order of accuracy. Another comparison based on the Lorenz equation compared the ADM with 15 terms of the series with Runge-Kutta-Verner schemes of 5^{th} and 6^{th} order which showed the ADM to be more accurate [33]. Using the ADM with 4 terms and comparing it to the classical RK method displayed errors of the same order for both methods tested on different linear and non-linear equations [31].

1.2 Contribution of this Work

Besides the standard stability and convergence requirements, a viable time stepping scheme in productive simulations has to fulfill two important properties: accurate solutions even for larger time step sizes and, related to larger time step sizes, a wall-clock time as small as possible. Considering the ADM, previous work has mainly focused on its accuracy. Our focus is on a reduction of the wall-clock time. This reduction can be achieved by (a) large time steps and by (b) speeding up the time stepping method by exploiting additional degrees of parallelism in the time dimension. Both parts are less investigated so far regarding the ADM as a time stepping method and this is the main focus of this work. By comparing the ADM to the RK method we investigate whether the discrete ADM is a viable method regarding real-world scenarios and if it is competitive to other existing time stepping methods.

In Sect. 2 we present a short introduction to the viscous Burgers equation. This is followed by the comparison of the Runge-Kutta and discrete Adomian decomposition methods in Sect. 3, where the discrete ADM is also described in more detail. The additional degrees of parallelism of the ADM are then discussed in Sect. 4, followed by results of the numerical studies in Sect. 5 and conclusions in Sect. 6.

2 Burgers' Equation

The Navier-Stokes equations (NSE) are the fundamental equations of many computational fluid dynamics (CFD) problems [34]. In their incompressible form they read

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} &= -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\mathbf{F}\ , \\ \nabla\cdot\mathbf{u} &= 0\ , \end{aligned} \tag{1}$$

where $\mathbf{u}$ denotes the velocity, p the pressure, ρ the density, ν the kinematic viscosity, and $\mathbf{F}$ the external forces.

The viscous Burgers equation, which was introduced by Bateman [11] and extensively studied by Burgers, e.g. [12], can be derived from (1). To gain Burgers' equation, we drop the internal sources $-\nabla p/\rho$ and external sources $\mathbf{F}/\rho$ leaving us with

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{u} = \nu\nabla^2\mathbf{u}\ .$$

This simplification of the NSE still contains the non-linearity, which is one of the terms of interest when it comes to developing numerical schemes for the NSE.

For a better presentation of our results, we use the one dimensional formulation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}\ . \tag{2}$$

3 Comparison of Time Integration Methods

In this section, we compare the Runge-Kutta method to the discrete Adomian decomposition method and provide a more detailed description of both methods.

3.1 Runge-Kutta Method

In our comparison we use the well-known explicit Runge-Kutta method (RK) [29] and employ schemes of order $p = 1, \ldots, 4$. The RK method solving $\mathrm{d}u/\mathrm{d}t = f(t, u)$ reads

$$\begin{aligned} u_{n+1} &= u_n + \Delta t \sum_{i=1}^{s} b_i k_i, \\ k_1 &= f(t_n, u_n), \\ k_i &= f\left(t_n + c_i \Delta t, u_n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j \right), \qquad \text{for } i = 2, \ldots, s, \end{aligned}$$

where u_n denotes the approximation of $u(t_0 + n\Delta t)$, and the coefficients a, b, c are specific to the used s-stage RK scheme. These coefficients are usually arranged in a Butcher tableau [13]

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & \\ \hline & b_1 & \cdots & b_{s-1} & b_s \end{array} .$$

For an s-stage RK method of order p, $s \geq p$ holds and if $p \geq 5$, then $s > p$ [14]. From this follows, that as many or more function evaluations are necessary per time step as the order of the scheme is.

Since the RK method is of explicit nature it has a limited stability region. The time step is bounded by the condition that the physical propagation of information has to be equal to, or slower than the propagation of information on the discretized grid. This is reflected in the CFL condition [18]. With the Burgers equation the fastest propagation of information can either be by advection or diffusion making the maximal time step width proportional to $\Delta t_{\max} \propto \Delta x$ or $\Delta t_{\max} \propto \Delta x^2$, respectively.

3.2 Discrete Adomian Decomposition Method

Next, we describe the discrete Adomian decomposition method (DADM). For this purpose, we first provide an overview of the Adomian decomposition method, before we explain its discretized version. Finally, we apply the DADM to the Burgers equation.

Adomian Decomposition Method The basis of the DADM is the Adomian decomposition method which was developed by Adomian in the '80s and '90s, e.g. [6]. The main idea is to decompose the non-linearities of equations into a series of Adomian polynomials. With the ADM it is possible to calculate the analytical solution or, if this is not possible, to gain an approximation with a fast convergence to the actual solution. This is possible as no discretization (in case of the analytical version described here), linearization, or perturbation theory has to be applied to the non-linear term [3, 4, 6, 7].

Following the notation in [5] the method is in general applied to an equation $\mathcal{F}(u) = g$, where $\mathcal{F}$ is a (non-)linear operator. This operator can be decomposed in its linear and non-linear parts $\mathcal{L} + \mathcal{R}$ and $\mathcal{N}$, respectively. Here, $\mathcal{L}$ denotes an easily invertible operator of highest order and $\mathcal{R}$ comprises the remaining linear parts. Using this notation we get

$$\mathcal{L}u + \mathcal{R}u + \mathcal{N}(u) = g \; . \tag{3}$$

Let the inverse of $\mathcal{L}$ be $\mathcal{L}^{-1}$. Solving (3) for $\mathcal{L}u$ and applying the inverse yields

$$\mathcal{L}^{-1}\mathcal{L}u = \mathcal{L}^{-1}g - \mathcal{L}^{-1}\mathcal{R}u - \mathcal{L}^{-1}\mathcal{N}(u) \; . \tag{4}$$

The left hand side of this equation can be evaluated to $\mathcal{L}^{-1}\mathcal{L}u = u + C$, where C are the integration constants given either by initial or boundary conditions.

The idea of the ADM is now to expand the unknown into the series $u = \sum_{i=0}^{\infty} u_i$ and write the non-linearity as $\mathcal{N}(u) = \sum_{i=0}^{\infty} A_i$, where the A_i are the Adomian polynomials. With the series representation we can identify $u_0 = \mathcal{L}^{-1}g - C$. Now, we can write (4) as

$$\sum_{i=0}^{\infty} u_i = u_0 - \mathcal{L}^{-1}\mathcal{R}\sum_{i=0}^{\infty} u_i - \mathcal{L}^{-1}\sum_{i=0}^{\infty} A_i \; . \tag{5}$$

Therefore, we have the recursive relation

$$\begin{aligned} u_1 &= -\mathcal{L}^{-1}\mathcal{R}u_0 - \mathcal{L}^{-1}A_0 \\ u_2 &= -\mathcal{L}^{-1}\mathcal{R}u_1 - \mathcal{L}^{-1}A_1 \\ &\vdots \\ u_{i+1} &= -\mathcal{L}^{-1}\mathcal{R}u_i - \mathcal{L}^{-1}A_i \\ &\vdots \end{aligned} \tag{6}$$

for the u_i of the expansion series.

The Adomian polynomials can formally be written as

$$A_i(u_0, u_1, \ldots, u_i) = \frac{1}{i!}\left[\frac{\mathrm{d}^i}{\mathrm{d}\lambda^i}\mathcal{N}\left(\sum_{j=0}^{\infty} \lambda^j u_j\right)\right]_{\lambda=0} \; , \tag{7}$$

(see [1, 3, 4, 6, 16]). We notice that A_i is only dependent on $u_0, u_1, \dots, u_i$ and other independent variables. This is important for the discussion of the parallelization in Sect. 4. In the end, the formulation of the A_i can be interpreted as a generalization of the Taylor series in the neighborhood of the function u_0

$$\mathcal{N}(u) = \sum_{i=0}^{\infty} A_i = \sum_{i=0}^{\infty} \frac{1}{i!} \left(u - u_0\right)^i \mathcal{N}^{(i)}\left(u_0\right) \ .$$

Discretization of the Adomian Decomposition Method Using the ADM numerically requires a truncation to a finite series. As stated in [15, 21, 26] this truncation leads to a small convergence radius of the ADM. In [35] this small convergence radius was circumvented by applying the ADM as a time stepping scheme iteratively on a discretized spatial domain. Since our ansatz is applying discretization in space and time as-well, we adapt the name discrete Adomian decomposition method.

The truncated version of the decomposition (5) to approximate the solution reads

$$\sum_{i=0}^{\infty} u_i \approx \sum_{i=0}^{p} u_i = u_0 - \mathcal{L}^{-1}\mathcal{R}\sum_{i=0}^{p-1} u_i - \mathcal{L}^{-1}\sum_{i=0}^{p-1} A_i \ ,$$

where p denotes the order of the approximation. The truncated version can be used as a time stepping scheme, by repeatedly applying the ADM with a stable time step size $\varDelta t$. The starting point of each new time step is given by the result of the previous time step. We denote the approximation at time $t_n = t_0 + n\varDelta t$ by $\hat{u}^n$. With this we can define the time stepping scheme of order p as

$$\hat{u}^{n+1} = \hat{u}^n + \sum_{i=1}^{p} u_i \ . \tag{8}$$

For one time step of this scheme p evaluations of the linear operator are necessary and the non-linearity has to be evaluated in the order of $O(p^2)$ times because of (7).

DADM Applied to Burgers' Equation We compare the DADM with the RK method applied to the Burgers equation. For the Burgers equation (2) we can identify $\mathcal{L}u = \partial u/\partial t$, $\mathcal{R}u = -\nu\partial^2 u/\partial x^2$ and $\mathcal{N}(u) = u\partial u/\partial x$. Using this and having (8) in mind the recursion (6) reads

$$\begin{aligned} u_0 &= \hat{u}^n \\ u_1 &= -\frac{\varDelta t}{1}(\mathcal{R}u_0 + A_0) \\ u_2 &= -\frac{\varDelta t}{2}(\mathcal{R}u_1 + A_1) \\ &\ \vdots \\ u_p &= -\frac{\varDelta t}{(p-1)}(\mathcal{R}u_{p-1} + A_{p-1}) \ . \end{aligned} \tag{9}$$

Here, we utilized $\mathcal{L}^{-1} = \int_{t_n}^{t_{n+1}} \cdot \, \mathrm{d}t$ and $\hat{u}^n$ being independent on t. Equation (7) can be written as

$$A_i = \sum_{j=0}^{i} u_j \frac{\partial u_{i-j}}{\partial x} \tag{10}$$

in this context.

3.3 Comparison of DADM and RK

Both methods investigated in this comparison are explicit methods, which can be formulated with different orders of accuracy. We introduce the notation DADMp and RKp, where p denotes the order of the method. For the purpose of readability, we also introduce the notation $\mathcal{N}(x(u), y(u)) = x(u)\partial y(u)/\partial x$ for the non-linearity.

First Order We compare DADM1 with the explicit Euler method, which is equivalent with RK1. Writing out DADM1 yields

$$\hat{u}^{n+1} = \hat{u}^n + \sum_{i=1}^{1} u_i = \hat{u}^n - \Delta t \left(\mathcal{R}\hat{u}^n + \mathcal{N}\left(\hat{u}^n, \hat{u}^n\right)\right) \ ,$$

and using the same operator notation for RK1, we get

$$u^{n+1} = u^n - \Delta t \left(\mathcal{R}u^n + \mathcal{N}\left(u^n, u^n\right)\right) \ .$$

As we can see, both methods are exactly the same.

Second Order Writing $\hat{u}^{n+1}$ as a function of $\hat{u}^n$ for DADM2 yields

$$\begin{aligned}
\hat{u}^{n+1} &= \hat{u}^n + \sum_{i=1}^{2} u_i \\
&= \hat{u}^n - \Delta t \left(\mathcal{R}\hat{u}^n + \mathcal{N}\left(\hat{u}^n, \hat{u}^n\right)\right) \\
&\quad + \frac{\Delta t^2}{2} \left(\mathcal{R}^2 \hat{u}^n + \mathcal{R}\mathcal{N}\left(\hat{u}^n, \hat{u}^n\right) + \mathcal{N}\left(\mathcal{R}\hat{u}^n + \mathcal{N}\left(\hat{u}^n, \hat{u}^n\right), \hat{u}^n\right) \right. \\
&\quad \left. + \mathcal{N}\left(\hat{u}^n, \mathcal{R}\hat{u}^n + \mathcal{N}\left(\hat{u}^n, \hat{u}^n\right)\right)\right) \ .
\end{aligned}$$

We can see, that this is the 2^{nd} order exact representation of the generalized Taylor series developed around the function $\hat{u}^n$, as expected with the ADM.

The RK2 method, which is the midpoint method, can also be written without intermediate stages

$$\begin{aligned}
u^{n+1} &= u^n - \Delta t \left(\mathcal{R}u^n + \mathcal{N}\left(u^n, u^n\right)\right) \\
&\quad + \frac{\Delta t^2}{2} \left(\mathcal{R}^2 u^n + \mathcal{R}\mathcal{N}\left(u^n, u^n\right) + \mathcal{N}\left(\mathcal{R}u^n + \mathcal{N}\left(u^n, u^n\right), u^n\right) \right. \\
&\quad \left. + \mathcal{N}\left(u^n, \mathcal{R}u^n + \mathcal{N}\left(u^n, u^n\right)\right)\right) \\
&\quad + \frac{\Delta t^3}{4} \left(\mathcal{N}\left(\mathcal{N}\left(u^n\right), \mathcal{R}u^n\right) + \mathcal{N}\left(\mathcal{R}u^n, \mathcal{N}\left(u^n, u^n\right)\right) \right. \\
&\quad \left. + \mathcal{N}\left(\mathcal{N}\left(u^n\right), \mathcal{N}\left(u^n\right)\right) + \mathcal{N}\left(\mathcal{R}u^n, \mathcal{R}u^n\right)\right) \ .
\end{aligned}$$

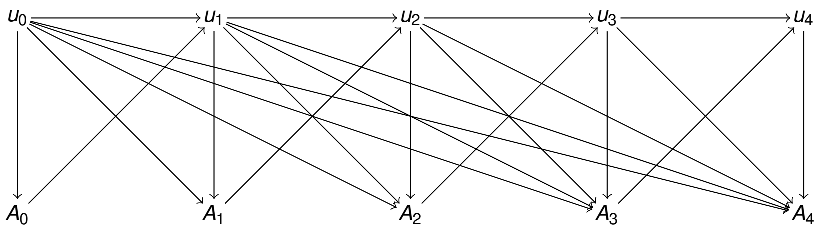

Fig. 1. Data dependencies indicated by the arrows for the first four Adomian polynomials A_i are shown to underline the fact that each polynomial is dependent on all previously calculated velocities u_i

In this formulation we recognize all terms of DADM2. In addition, there are some terms of 3^{rd} order. These terms do not represent the third order terms of the generalized Taylor series and, therefore, impact the error but not the convergence order of the numerical scheme.

Higher Order Higher order methods show a comparable picture as the 2^{nd} order methods. The DADMp is, as intended, always accurate up to order p regarding the generalized Taylor series. RKp is based on the same terms as DADMp, but also has additional terms of orders higher than p.

Similar solutions are expected due to this from the numerical schemes. The remaining question is, whether the additional terms are beneficial for the stability (maximal stable time step width).

4 Degrees of Parallelism

The goal of this work is to investigate whether DADM is a viable method and competitive to other time steppers like RK. For this purpose, we have already compared both methods in Sect. 3. There, we found that both methods are very similar, but need different numbers of function evaluations. One could argue that the large amount of evaluations of the non-linear term in DADM makes this method less efficient than RK.

Taking a closer look at the data dependencies of the calculation of the Adomian polynomials, we can recognize (see also Sect. 3.2) that these are a sum over already calculated variables, sketched in Fig. 1. We can use this to parallelize the calculation of the terms in (7), respectively (10).

To our best knowledge, these additional degrees of parallelism were not studied in the literature so far. A strictly sequential evaluation of a p-stage DADM, leads to $O(p)$ function evaluations plus $\sum_{n=1}^{p} n = \frac{1}{2}p(p+1) = O(p^2)$ additional evaluations of the non-linear term following (9) and (10). However, each stage leads to additional independent terms which increases the degree of parallelism for higher stages over the sum. Assuming that the evaluations of each term in the stage are computationally more expensive compared to the reduction over

Table 1. This table shows distribution of workload on multi-core system. The left most column indicates the necessary communication. The columns headed by P_i denote the processors and show the values on the processor. In the last column the dependencies and calculations are given. Here, the calculation for the first three velocities is shown to demonstrate the idea

	P_0	P_1	P_2	P_3	Calculations & dependencies
Bcast	u_0				
	A_0	A_0	A_0	A_0	$A_0 = (u_0 \cdot \nabla)u_0$
	u_1	u_1	u_1	u_1	$u_1 = -\Delta t(\nu\nabla^2 u_0 + A_0)$
Reduce	$u_0\nabla \cdot u_1$	$u_1\nabla \cdot u_0$			
	A_1	A_1	A_1	A_1	$A_1 = \sum_{i=0}^{1}(u_i \cdot \nabla)u_{1-i}$
	u_2	u_2	u_2	u_2	$u_2 = -\Delta t/2(\nu\nabla^2 u_1 + A_1)$
Reduce	$u_0\nabla \cdot u_2$	$u_1\nabla \cdot u_1$	$u_2\nabla \cdot u_0$		
	A_2	A_2	A_2	A_2	$A_2 = \sum_{i=0}^{2}(u_i \cdot \nabla)u_{2-i}$
	u_3	u_3	u_3	u_3	$u_3 = -\Delta t/3(\nu\nabla^2 u_2 + A_2)$

the sum, this reduces the runtime complexity to $O(p)$. In Table 1 the workload distribution on a multi-core system is depicted. In addition to the distribution, we highlight the required data dependencies, hence the communication.

We would also like to point out that some terms of (10) can already be computed at an earlier stage, e.g. the term $u_1\nabla\cdot u_1$ belonging to A_2 can be evaluated at the time the terms for A_1 are evaluated. Hence, a formulation as a directed acyclic graph (DAG) scheduling problem could lead to further improvements regarding the wall-clock time and the required computational resources.

5 Numerical Studies

The numerical studies described in this section are used to expand the comparison of RKp and DADMp. We start with a description of the benchmark used for the studies. With this benchmark scenario we compare the maximal time step size and the convergence order of both schemes.

5.1 Spatial Discretization

As the interest of our study is the comparison of the time stepping methods, we apply spectral methods for the discretization of our spatial domain. With this we are able to minimize the errors induced by the spatial discretization of the equation.

The method of choice is the periodic Fourier basis, e.g. [19]. This basis allows to evaluate linear operators element-wise in spectral space, hence without loss

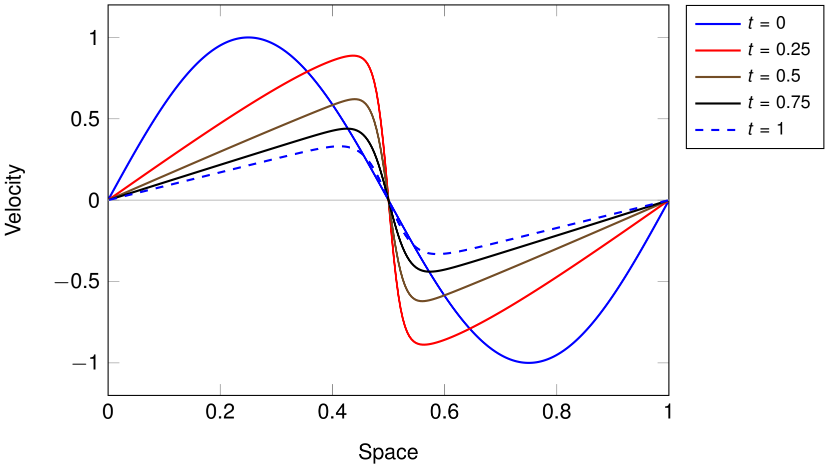

Fig. 2. Snapshots of the analytical solution to the viscous Burgers equation for the velocity over the spatial domain with $\nu = 0.01$

of accuracy. Therefore, the error of this operation is in the order of machine precision, if the function can be represented exactly in spectral space with the given number of modes.

Non-linear operators are evaluated in a pseudo-spectral fashion [10, 23] in physical space to reduce the complexity of the calculation, which would be given with the necessary convolution of all spectral series. A standard anti-aliasing technique is used to filter the spurious modes created in physical space by the pseudo-spectral calculation, e.g. [30].

5.2 Benchmark Scenario

For our benchmark the spatial domain is initialized at $t = t_0$ with a one dimensional sinus wave as shown in Fig. 2. The space time domain, on which the benchmark is run, is $x, t \in [0, 1]^2$. Even with an initial distribution as simple as this, it would not be very easy to apply the ADM numerically without a spatial discretization. This is due to the convergence radius requiring multiple steps of the ADM and the intermediate solutions are not easily differentiable.

As the focus of the Adomian decomposition is the non-linearity, we use a viscosity of $\nu = 0.01$, which reduces the influence of the diffusion compared to the advection. With this choice the snapshots in Fig. 2 show that both parts have a visible influence on the solution. The spatial domain is discretized with a grid spacing of $\Delta x = 1/256$, which corresponds to $N = 170$ spectral modes. For the comparison of the convergence order, we run the schemes with time step widths of $\Delta t = \{1.25 \times 10^{-4}, 2.5 \times 10^{-4}, 5 \times 10^{-4}\}$. All errors are the difference between the numerical and analytical solution, which is given by the Hopf-Cole transformation [17, 25].

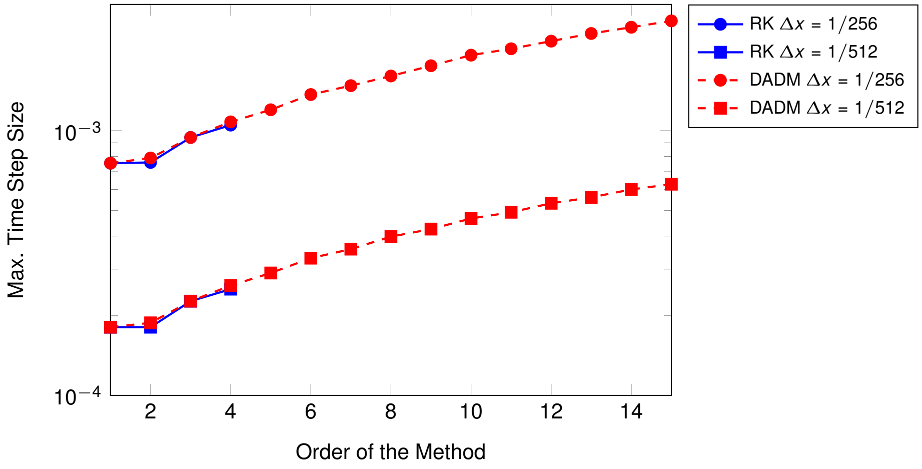

Fig. 3. Maximal time step Δt_{max} for converged calculations in time interval $t \in [0, 1]$ for RK (solid) and DADM (dashed) plotted over the order of the method

5.3 Stable Time Step Size Limitations

Since one of the possibilities to reduce the calculation time is to use large time step sizes, it is of interest to investigate the largest possible time step size with both methods. Therefore, we compare the maximal stable time step Δt_{max} for different orders of the DADM and RK method applied to the benchmark. The maximal stable time step for this investigation is the Δt with which a solution at the final time $t = 1$ was found without significant deviations from the analytical solution. The calculations were done with schemes of the orders $p = \{1, \ldots, 4\}$ for RK and with schemes of the order $p = \{1, \ldots, 15\}$ for DADM. Δt_{max} was determined with up to three significant digits.

In Fig. 3 the upper two curves show the results for the mentioned settings. Considering the schemes of order $p = \{1, \ldots, 4\}$, we can observe that both RK and DADM show very similar maximal stable time step sizes. For the even orders DADM allows time steps which are approximately 3% larger than those with RK. With an increasing order slightly larger time steps are possible. This trend continues for DADM for the other tested orders $p = \{5, \ldots, 15\}$.

The two bottom curves in Fig. 3 were calculated with $N = 341$ spectral modes. The smaller Δt_{max} is expected, because both schemes are of explicit nature and their time step size is bounded by the CFL condition. Increasing the number of spatial discretization points by a factor of two in this case leads to a reduction of the maximal stable time step width by a factor of four. From this we can deduce that the time step width is bounded by the diffusive information transport. In this case the relation $\Delta t_{max} \propto \Delta x^2$ holds.

From these results we can conclude that there seems to be no striking difference between RK and DADM considering Δt_{max}. Therefore, applying the parallelization described in Sect. 4 can make DADM competitive against RK for

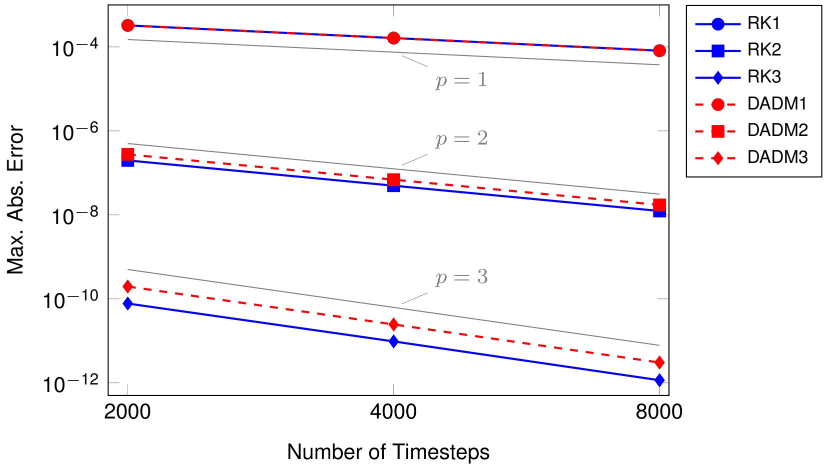

Fig. 4. Maximal absolute error at $t = 1$ plotted over the number of time steps for RK and DADM of order $p = 1, \ldots, 3$. Additionally, slopes of order $p = 1, \ldots, 3$ are shown to validate the order of the methods

high order methods, as there is no disadvantage in time to solution regarding the time step width.

5.4 Convergence Order

In addition to the maximal time step size, we take a look at the errors produced by the schemes to investigate the assumption made in Sect. 3.3, where we stated that the additional higher order terms of RK will reduce the error compared to DADM. Here, we compare the errors for the orders $p = 1, \ldots, 3$ of both schemes and show that the schemes converge with the given order.

In Fig. 4 the maximal absolute error at the final time $t = 1$ is shown for the three different time step widths of the benchmark. As expected from the comparison in Sect. 3.3, the first order methods have exactly the same error and converge according to their order. The errors of DADM2 are slightly larger than those of RK2. The difference is a result of the additional terms of higher order which appear in RK2. Both schemes converge with 2^{nd} order accuracy. For the third order schemes we get a comparable picture. Although both methods reach the expected order, the DADM3 produces larger errors than the RK3. Taking all three orders into account, we can see that the error difference increases with increasing order.

From a serial point of view this and having comparable maximal time step sizes suggests to prefer the RK method over the DADM. With the discussed parallelization DADM is still viable. For smaller orders the error difference is not significant. In case of the higher orders increasing the order from p to $p + 1$ for the DADM needs still as many function evaluations as RK for $p \geq 5$ and even less for $p \geq 7$ [14].

6 Conclusion and Discussion

In this work, we have discussed how the Adomian decomposition method (ADM) can be applied as an explicit time stepping scheme. We were able to extract additional degrees of parallelism within this method to reduce the runtime complexity from $O(n^2)$ to $O(n)$.

This reduction makes the ADM a viable competitor to the Runge-Kutta (RK) method for high-order schemes. In this case the number of function evaluations which have to be computed in serial is smaller for the ADM than for the RK method. A comparison based on numerical studies has shown that both methods have comparable maximal time step widths. The larger errors of the ADM in the high orders can be circumvented by increasing the order of the ADM from p to $p + 1$. This still leaves the ADM with less function evaluations done in serial than the RK method of order p. This increase of order is easy to accomplish, as it is straight forward to obtain higher order ADM methods. These results support the statement, that the ADM can be a viable time stepping method.

It is mentioned, e.g. in [32], that the maximal time step width of the ADM could be increased with Padé's rational approximation. This might increase the viability of the ADM as an explicit time stepping scheme further while still allowing to exploit the additional degrees of parallelism in time.

Acknowledgments The work of Andreas Schmitt is supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt.

This work was partially accomplished during a research stay at NCAR which provided the facilities.

References

1. Abbaoui, K., Cherruault, Y.: Convergence of Adomian's method applied to nonlinear equations. Mathematical and Computer Modelling **20**(9), 69–73 (1994). DOI 10.1016/0895-7177(94)00163-4
2. Abbasbandy, S., Darvishi, M.: A numerical solution of Burgers' equation by modified Adomian method. Applied Mathematics and Computation **163**(3), 1265–1272 (2005). DOI 10.1016/j.amc.2004.04.061
3. Adomian, G.: A new approach to nonlinear partial differential equations. Journal of Mathematical Analysis and Applications **102**(2), 420–434 (1984). DOI 10.1016/0022-247X(84)90182-3
4. Adomian, G.: Nonlinear Stochastic Operator Equations. Academic Press, New York (1986)
5. Adomian, G.: Solution of physical problems by decomposition. Computers & Mathematics with Applications **27**(9), 145–154 (1994). DOI 10.1016/0898-1221(94)90132-5
6. Adomian, G.: Solving Frontier Problems of Physics: The Decomposition Method. Springer Science+Business Media, Dordrecht (1994)

7. Adomian, G.: Explicit solutions of nonlinear partial differential equations. Applied Mathematics and Computation **88**(2), 117–126 (1997). DOI 10.1016/S0096-3003(96)00141-5
8. Akpan, I.: Adomian decomposition approach to the solution of the Burger's equation. American Journal of Computational Mathematics **5**, 329–335 (2015). DOI 10.4236/ajcm.2015.53030
9. Ascher, U.M., Ruuth, S.J., Spiteri, R.J.: Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. **25**(2-3), 151–167 (1997). DOI 10.1016/S0168-9274(97)00056-1
10. Barros, S.R., Peixoto, P.S.: Computational aspects of harmonic wavelet Galerkin methods and an application to a precipitation front propagation model. Computers & Mathematics with Applications **61**(4), 1217–1227 (2011)
11. Bateman, H.: Some recent researches on the motion of fluids. Monthly Weather Review **43**(4), 163–170 (1915). DOI 10.1175/1520-0493(1915)43⟨163:SRROTM⟩2.0.CO;2
12. Burgers, J.: A mathematical model illustrating the theory of turbulence. Advances in Applied Mechanics **1**, 171–199 (1948). DOI 10.1016/S0065-2156(08)70100-5
13. Butcher, J.C.: Implicit Runge-Kutta processes. Mathematics of Computation **18**(85), 50–64 (1964). DOI 10.1090/S0025-5718-1964-0159424-9
14. Butcher, J.C.: Numerical Method for Ordinary Differential Equations, 2 edn. John Wiley & Sons, Ltd (2008)
15. Cheng, M., Scott, K., Sun, Y., Wu, B.: Explicit solution of nonlinear electrochemical models by the decomposition method. Chemical Engineering & Technology **25**(12), 1155–1160 (2002). DOI 10.1002/1521-4125(20021210)25:12⟨1155::AID-CEAT1155⟩3.0.CO;2-A
16. Cherruault, Y., Adomian, G.: Decomposition methods: A new proof of convergence. Mathematical and Computer Modelling **18**(12), 103–106 (1993). DOI 10.1016/0895-7177(93)90233-O
17. Cole, J.D.: On a quasi-linear parabolic equation occuring in aerodynamics. Quarterly of Applied Mathematics **9**, 225–236 (1951). DOI 10.1090/qam/42889
18. Courant, R., Friedrichs, K., Lewy, H.: Über die partiellen Differenzengleichungen der mathematischen Physik. Mathematische Annalen **100**, 32–74 (1928). DOI 10.1007/BF01448839
19. Durran, D.: Numerical Methods for Fluid Dynamics: With Applications to Geophysics. Texts in Applied Mathematics. Springer New York (2010). DOI 10.1007/978-1-4419-6412-0
20. El-Sayed, S.M., Kaya, D.: On the numerical solution of the system of two-dimensional Burgers' equations by the decomposition method. Applied Mathematics and Computation **158**(1), 101–109 (2004). DOI 10.1016/j.amc.2003.08.066
21. El-Tawil, M.A., Bahnasawi, A.A., Abdel-Naby, A.: Solving Riccati differential equation using Adomian's decomposition method. Applied Mathematics and Computation **157**(2), 503–514 (2004). DOI 10.1016/j.amc.2003.08.049
22. Gander, M.J.: 50 years of time parallel time integration. In: Multiple Shooting and Time Domain Decomposition. Springer (2015). DOI 10.1007/978-3-319-23321-5_3
23. Gottlieb, D., Orszag, S.: Numerical Analysis of Spectral Methods: Theory and Applications. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (1977)
24. Guellal, S., Grimalt, P., Cherruault, Y.: Numerical study of Lorenz's equation by the Adomian method. Computers & Mathematics with Applications **33**(3), 25–29 (1997). DOI 10.1016/S0898-1221(96)00234-9

25. Hopf, E.: The partial differential equation $u_t+uu_x = \mu_{xx}$. Communications on Pure and Applied Mathematics **3**(3), 201–230 (1950). DOI 10.1002/cpa.3160030302
26. Jiao, Y., Yamamoto, Y., Dang, C., Hao, Y.: An aftertreatment technique for improving the accuracy of Adomian's decomposition method. Computers & Mathematics with Applications **43**(6), 783–798 (2002). DOI 10.1016/S0898-1221(01) 00321-2
27. Kaya, D., Yokus, A.: A numerical comparison of partial solutions in the decomposition method for linear and nonlinear partial differential equations. Mathematics and Computers in Simulation **60**(6), 507–512 (2002). DOI 10.1016/S0378-4754(01) 00438-4
28. Kaya, D., Yokus, A.: A decomposition method for finding solitary and periodic solutions for a coupled higher-dimensional Burgers equations. Applied Mathematics and Computation **164**(3), 857–864 (2005). DOI 10.1016/j.amc.2004.06.012
29. LeVeque, R.: Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104) (2007)
30. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., et al.: Numerical Recipes, vol. 3. cambridge University Press, cambridge (1989)
31. Shawagfeh, N., Kaya, D.: Comparing numerical methods for the solutions of systems of ordinary differential equations. Applied Mathematics Letters **17**(3), 323–328 (2004). DOI 10.1016/S0893-9659(04)90070-5
32. de Sousa Basto, M.J.F.: Adomian decomposition method, nonlinear equations and spectral solutions of Burgers equation. Ph.D. thesis, Faculdade de Engenharia da Universidade do Porto (2006)
33. Vadasz, P., Olek, S.: Convergence and accuracy of Adomian's decomposition method for the solution of Lorenz equations. International Journal of Heat and Mass Transfer **43**(10), 1715–1734 (2000). DOI 10.1016/S0017-9310(99)00260-4
34. Wesseling, P.: Principles of Computational Fluid Dynamics. Springer Series in Computational Mathematics. Springer Berlin Heidelberg (2009)
35. Zhu, H., Shu, H., Ding, M.: Numerical solutions of two-dimensional Burgers' equations by discrete Adomian decomposition method. Computers & Mathematics with Applications **60**(3), 840–848 (2010). DOI 10.1016/j.camwa.2010.05.031