ESAIM: M2AN 56 (2022) 743–766
https://doi.org/10.1051/m2an/2022018

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

HIGH ORDER LINEARLY IMPLICIT METHODS FOR EVOLUTION EQUATIONS

Guillaume Dujardin[1] and Ingrid Lacroix-Violet[2,*]

Abstract. This paper introduces a new class of numerical methods for the time integration of evolution equations set as Cauchy problems of ODEs or PDEs. The systematic design of these methods mixes the Runge–Kutta collocation formalism with collocation techniques, in such a way that the methods are linearly implicit and have high order. The fact that these methods are implicit allows to avoid CFL conditions when the large systems to integrate come from the space discretization of evolution PDEs. Moreover, these methods are expected to be efficient since they only require to solve one linear system of equations at each time step, and efficient techniques from the literature can be used to do so. After the introduction of the methods, we set suitable definitions of consistency and stability for these methods. This allows for a proof that arbitrarily high order linearly implicit methods exist and converge when applied to ODEs. Eventually, we perform numerical experiments on ODEs and PDEs that illustrate our theoretical results for ODEs, and compare our methods with standard methods for several evolution PDEs.

Mathematics Subject Classification. 65M12, 65M70, 65L20, 65L06, 81Q05, 35Q41, 35K05.

Received October 28, 2020. Accepted February 11, 2022.

1. Introduction

The goal of this paper is to introduce a new class of methods for the time integration of evolution problems, set as (systems of) deterministic ODEs or PDEs. This class consists in methods of arbitrarily high order that require only the solution of one linear problem at each time step: no nonlinear system is to be solved. As is usual in the litterature, we call these methods linearly implicit. They rely on the combination of a classical collocation Runge–Kutta method with a specific treatment for the nonlinearity.

In particular, we show that, using the methods developed in this paper, one can solve numerically virtually any ODE, up to any order, by solving only linear systems at each time step. Moreover, we believe that this new class of methods can help dramatically reducing the computational time in several cases of time integration of evolution PDEs. Indeed, after space discretization of an evolution PDE, if one uses, say, an implicit (for stability reasons) Runge–Kutta method, then one needs to solve a nonlinear problem in high dimension at each time step (and one may use a fixed-point method or a Newton method to do so, for example). With

Keywords and phrases. Cauchy problems, evolution equations, time integration, numerical methods, high order, linearly implicit methods.

[1] Univ. Lille, Inria, CNRS, UMR 8524 – Laboratoire Paul Painlevé, F-59000 Lille, France.

[2] Université de Lorraine, CNRS, IECL, F-54000 Nancy, France.

*Corresponding author: ingrid.lacroix@univ-lorraine.fr

the methods introduced in this paper, the integration over any time step can be carried out using only the solutions of linear systems in high dimension, and one can rely on very efficient techniques, either direct (LU factorization, Choleski factorization, etc.) or iterative (Jacobi method, Gauss–Seidel method, conjugate gradient, Krylov subspace method, etc. [32]), depending on the structure of the problem at hand, to do so.

Of course, high order one step methods in time exist in the literature since the pioneer work of Runge [31] and Kutta [26]. The interested reader may refer to [19] for the integration of nonstiff problems and [18] for the integration of stiff problems, and to [10] for historical notes and references. Some methods are explicit, and lead, for PDE problems, to restrictive CFL conditions in general. Some methods are fully implicit and require the solution of nonlinear systems that are high dimensional in the PDE approximation context. Other high order methods have been developed for PDEs. For example, high order exponential integrators have been used for parabolic problems [22,23] and for NLS equations [6,16]. Beyond the analysis presented in this paper in an ODE context, one of the goals of this paper is to convince the reader that, in a PDE context, linearly implicit methods such as that developed below can outperform classical methods from the literature with the same order. This means that they require less CPU time to compute an approximation of the solution with a given (small) error.

The methods introduced in this paper are *not* linear multistep methods (see [14] or Chapter III of [19]). Indeed, for a general vector field, linear multistep methods are either explicit or fully implicit, while the methods introduced in this paper are only linearly implicit. Let us mention, however, that linearly implicit linear multistep methods have been developed and analysed, for example in [1] for nonlinear parabolic equations (see also [2]). In this paper, we shall introduce suitable concepts of consistency, stability and convergence for our methods, and we sometimes borrow the vocabulary to that of linear multistep methods, but the definitions are indeed different. Note that the methods introduced in this paper are *not* one-step methods either. Therefore, one cannot use composition techniques (see [29,36,38]) directly to build up high order methods from lower order ones: deriving a linearly implicit high order method (that is convergent in a reasonable sense) is a challenge *per se*, that we tackle in this paper. Another important class of high order methods, introduced by J. Butcher, not to be confused with the one introduced in this paper, is that of DIMSIMs (Diagonally Implicit MultiStep Integration Methods) [8], where the word "implicit" does not refer at all to "linearly implicit" but rather to "fully implicit" (meaning: nonlinearly, even if diagonally). These methods have been later generalised in a class called General Linear Methods (GLM) [9], that does not contain the methods introduced in this paper either.

The methods introduced in this paper are *not* classical linearly implicit methods either. Indeed, such methods have a long history, which dates back at least to the work of Rosenbrock [30]. They have been developped and analysed on several evolution equations in several contexts by numerous authors. For example, the work of Rosenbrock was revisited in [24] where Rosenbrock-Wanner (ROW) methods are introduced. The concept of B-convergence has been developed in [17], analysed in [34,35] for linearly implicit one step methods (see also [3]). These methods have been applied to multibody systems in [37], to nonlinear parabolic equations in [28], to advection-reaction-diffusion equations in [11] and more recently to surface evolution in [25]. Indeed, such methods always involve derivatives of the vector field (or part of the vector field), or (sometimes crude) approximations to this derivative. In contrast, the methods developed in this paper do not: They rely on an accurate interpolation procedure in time for the nonlinear terms.

The methods introduced in this paper use additional variables to take care of the nonlinear terms with high order accuracy, while making it possible to solve only linear systems at each time step. Numerical methods using additional variables are not new. Indeed, such methods have been developped in different contexts in order to achieve qualitative properties of the methods. For example, the relaxation method introduced by C. Besse [5] for the nonlinear Schrödinger (NLS) equation is a second order scheme [7] which preserves a discrete energy. That relaxation method [5] uses one additional variable to approximate part of the nonlinearity in the NLS equation and is linearly implicit. In this sense, the methods developped in this paper may be seen as generalizations of this relaxation method. However, our goal is now to develop high order methods. To do so, we use a higher order collocation approximation of part of the nonlinear terms in the equation. Another class of numerical methods using additional variables is that of scalar auxiliary variable methods (SAV) [33] and multiple scalar auxiliary variable (MSAV) [12]. This class was introduced to produce unconditionnaly stable

schemes for dissipative problems with gradient flow structure. The auxiliary variable in this context is used to ensure discrete energy decay. The order of the methods (1 or 2 in the references above) is not the main issue.

Let us mention two additional goals that the authors aim at tackling with the methods introduced in this paper. First, the authors would like to be able to develop a stability (and convergence) analysis for stiff problems, *i.e.* an analysis with constants that depend only on the class of the linear part of the vector field (later referred to as L, see (2.1)) and not on that linear part itself. This would allow for the numerical treatment of evolution PDE problems, as well as their space discretizations. This will be achieved in a forthcoming work, even if numerical examples of PDE problems are presented in Section 3. Second, the authors would like to build up high order linearly implicit methods with suitable qualitative properties (*e.g.* energy decay for dissipative problems, or energy preservation for hamiltonian problems). For example, the relaxation method [5], which belongs to the class of linearly implicit methods described in this paper, preserves an energy when applied to the nonlinear Schrödinger equation. For this reason, this paper only deals with constant time step methods.

The outline of this paper is as follows. In Section 2, we introduce the methods for a general semilinear evolution problem (ODE or PDE) and we introduce specific notions of stability, consistency and convergence for our class of methods. Morover, we show, in a constructive way, that stable methods of arbitrarily high order exist in Theorem 2.5 and Corollary 2.6. The main theoretical result of this paper is that one can build up arbitrarily high order convergent linearly implicit methods for ODEs (Thm. 2.9). We conclude Section 2 with examples of methods of order 1, 2, 4 and 6. In Section 3, we provide numerical examples of solutions of ODEs and PDEs. These numerical experiments illustrate the convergence result of Theorem 2.9 for evolution ODEs. Moreover, they indicate that the result of Theorem 2.9 is still valid in several PDE contexts. We consider for example a NLS equation in 1d and 2d and nonlinear heat equation in 1d. The main result of the numerical experiments of Section 3 is that, for ODEs, the linearly implicit methods do not dramatically outperform classical methods from the literature with the same order, no matter whether they are implicit or explicit (see Sect. 3.1). However, for the approximation of evolution PDEs in 1d (see Sect. 3.2.1), with moderate space discretization, the linearly implicit methods show performances comparable to that of explicit methods. Moreover, for the approximation of evolution PDEs in 2d (see Sect. 3.2.2) with precise space discretization (leading to high number of unknowns), the linearly implicit methods developed in this paper manage to outperform standard methods from the literature with the same order.

2. Linearly implicit methods of arbitrarily high order

2.1. Introduction of the methods

We consider a semilinear autonomous evolution equation of the form

$$\partial_t u = Lu + N(u)u, \tag{2.1}$$

where L is a linear differential operator and N is a nonlinear function of u. One can think for examples of the NLS equation, the nonlinear heat equation, or a simple ODE (see Sect. 3 for actual examples). We start at time $t = 0$ with an initial datum u^0 (with 0 in superscript) in some functional space so that the Cauchy problem is well-posed on some interval $[0, T^\star)$ with $T^\star > 0$. We choose $h > 0$ and set $t_n = nh$ for $n \in \mathbb{N}$ as long as $t_n < T^\star$.

Let us now start with the presentation of the new class of methods. Assume a collocation Runge–Kutta method with $s \geqslant 1$ stages is given with coefficients $0 \leqslant c_1 < \cdots < c_s \leqslant 1$, $(a_{i,j})_{1\leqslant i,j\leqslant s}$ and $(b_i)_{1\leqslant i\leqslant s}$. We denote by c the vector $(c_i)_{1\leqslant i\leqslant s}$ and by $\mathbb{1}$ the vector of size s with all entries equal to one.

We denote by u the exact solution of (2.1) and we set $\gamma(t) = N(u(t,\cdot))$. We assume we are given s approximations

$$\gamma_{n-1+c_i} \sim \gamma(t_{n-1} + c_i h) \qquad 1 \leqslant i \leqslant s,$$

and another approximation $u_n \sim u(t_n, \cdot)$. For possible ways of computing the s first approximations, we refer to Remark 2.11. For the numerical initial datum u_0 (with subscript 0), we consider an approximation of the exact

initial datum u^0 (with superscript 0) of equation (2.1) or its exact value. For $(\theta_1, \dots, \theta_s) \in \mathbb{R}^s$ and $D \in \mathcal{M}_s(\mathbb{R})$ to be chosen later, we define explicitly $(\gamma_{n+c_1}, \dots, \gamma_{n+c_s})$ with the relation

$$\begin{bmatrix} \gamma_{n+c_1} \\ \vdots \\ \gamma_{n+c_s} \end{bmatrix} = D \begin{bmatrix} \gamma_{n-1+c_1} \\ \vdots \\ \gamma_{n-1+c_s} \end{bmatrix} + \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_s \end{bmatrix} N(u_n). \tag{2.2}$$

Then, we define, linearly implicitly $(u_{n,1}, \dots, u_{n,s})$ as the solution of the Runge–Kutta like system

$$u_{n,i} = u_n + h \sum_{j=1}^{s} a_{i,j}(L + \gamma_{n+c_j})u_{n,j} \qquad 1 \leqslant i \leqslant s. \tag{2.3}$$

Last, we set explicitly

$$u_{n+1} = u_n + h \sum_{i=1}^{s} b_i(L + \gamma_{n+c_i})u_{n,i}. \tag{2.4}$$

The steps (2.2)–(2.4) define a linearly implicit method

$$(u_{n+1}, \gamma_{n+c_1}, \dots, \gamma_{n+c_s}) = \Phi_h(u_n, \gamma_{n-1+c_1}, \dots, \gamma_{n-1+c_s}).$$

Remark 2.1. The relaxation method introduced for the NLS equation

$$i\partial_t u + \Delta u = \lambda |u|^2 u,$$

in [5] writes

$$\begin{cases} \dfrac{\phi_{n+1/2} + \phi_{n-1/2}}{2} = |u_n|^2, \\ i\dfrac{u_{n+1} - u_n}{h} = \left(-\Delta + \lambda\phi_{n+1/2}\right)\dfrac{u_n + u_{n+1}}{2} \end{cases}. \tag{2.5}$$

This corresponds to taking $L = i\Delta$ and $N(u) = -i\lambda|u|^2$ in (2.1) and $s = 1$, $a_{1,1} = \frac{1}{2}$, $b_1 = 1$, $c_1 = \frac{1}{2}$, $\gamma_{n+1/2} = -i\lambda\phi_{n+1/2}$, $D = [-1]$ and $\theta_1 = 2$ in the numerical method (2.2)–(2.4).

In order to achieve order 2 with the relaxation method, C. Besse introduced a single auxiliary unknown ϕ on a staggered grid, corresponding to the relation $\phi = |u|^2$. Since we want to achieve higher orders, we decide to introduce several auxiliary unknowns on a staggered grid with s points, corresponding to the relation $\gamma = N(u)$.

Note that the convergence of order 2 of the relaxation method for the NLS equation is a difficult result and is *not* a consequence of the results of this paper, which only deals with ODEs in the theoretical part (Sect. 2) and allows for PDEs for illustration purposes (Sect. 3). Indeed, proving the convergence of a numerical time integration method applied to a PDE requires a functional analysis framework adapted to the PDE at hand and cannot in general be done once and for all. For the relaxation method applied to the NLS equation, the convergence of order 2 is proved in [7].

2.2. Consistency and stability of the step (2.2)

Let us denote by $\rho(D)$ the spectral radius of the matrix D, *i.e.* the biggest modulus of its complex eigenvalues. In view of relation (2.2), we decide to set the following definitions for the stability and consistency of the step (2.2).

Definition 2.2. The step (2.2) is said to be *stable* if

$$\sup_{n \in \mathbb{N}} \|D^n\| < +\infty,$$

for some norm on $\mathcal{M}_s(\mathbb{R})$. The step (2.2) is said to be *strongly stable* if $\rho(D) < 1$.

Remark 2.3. In the definition of the stability above, the boundedness of the sequence $(D^n)_{n\geqslant 0}$ is independant of the norm chosen on $\mathcal{M}_s(\mathbb{R})$. Moreover, it is equivalent to the fact that $\rho(D) < 1$ or $\rho(D) = 1$ with simple Jordan blocks for D for all eigenvalues of modulus 1. In particular, if the step (2.2) is strongly stable, then it is stable. The converse is not true in general. For example, the classical relaxation method of Remark 2.1 is stable but not strongly stable.

In order to define the consistency of step (2.2), we introduce the $s \times s$ square matrices $V_c, V_{c-\mathbb{1}}$ and Θ defined by

- for all $i \geqslant 1$ and $j \geqslant 1$, $\quad \left(V_c^h\right)_{ij} = (c_i h)^{j-1}$,
- for all $i \geqslant 1$ and $j \geqslant 1$, $\quad \left(V_{c-\mathbb{1}}^h\right)_{ij} = ((c_i - 1)h)^{j-1}$,
- for all $i \geqslant 1$ and $j \geqslant 2$, $\quad (\Theta)_{i1} = \theta_i,\ (\Theta)_{ij} = 0$.

Definition 2.4. We say that the step (2.2) is consistent of order s if for all $h > 0$,

$$V_c^h = DV_{c-\mathbb{1}}^h + \Theta. \tag{2.6}$$

This relation holds for all $h > 0$ if and only if it holds for $h = 1$, as we show below. Indeed, introducing the diagonal matrix $G(h)$ with coefficients $1, h, h^2, \ldots, h^{s-1}$, one has $V_c^h = V_c^1 G(h)$ and $V_{c-\mathbb{1}}^h = V_{c-\mathbb{1}}^1 G(h)$. Since the c_i are distinct, the Vandermonde matrices V_c^h and $V_{c-\mathbb{1}}^h$ are invertible. By the relation (2.6) and using the matrix $G(h)$, we have

$$D = \left(V_c^h - \Theta\right)\left(V_{c-\mathbb{1}}^h\right)^{-1} = \left(V_c^1 G(h) - \Theta\right)G(h)^{-1}\left(V_{c-\mathbb{1}}^1\right)^{-1} = V_c^1\left(V_{c-\mathbb{1}}^1\right)^{-1} - \Theta G(h)^{-1}\left(V_{c-\mathbb{1}}^1\right)^{-1}. \tag{2.7}$$

Since Θ is zero except maybe on its first column, we have $\Theta(G(h))^{-1} = \Theta$.

The definition of the step (2.2) of the method (2.2)–(2.4) depends on the s^2 coefficients of the matrix D and the s coefficients $\theta_1, \ldots, \theta_s$. Requiring that the step (2.2) is of order s provides us with s^2 linear equations between these unknowns (see relation (2.7)). In Theorem 2.5, we prove that we can add s equations involving these unknows by imposing the spectrum of the matrix D and that the system that we obtain has indeed a unique solution. This will allow in particular to prove the existence of stable and strongly stable steps (2.2) with order s (see Cor. 2.6).

Theorem 2.5. *Assume $c_1, \ldots, c_s$ are fixed and distinct as above. For all disctinct $\lambda_1, \ldots, \lambda_s \in \mathbb{C}\backslash\{1\}$, there exists a unique $((\theta_1, \ldots, \theta_s), D) \in \mathbb{C}^s \times \mathcal{M}_s(\mathbb{C})$ such that the step (2.2) is of order s and the spectrum of the matrix D is exactly $\{\lambda_1, \ldots, \lambda_s\}$. If moreover the set $\{\lambda_1, \ldots, \lambda_s\}$ is stable under complex conjugation, then $(\theta_i)_{1\leqslant i\leqslant s} \in \mathbb{R}^s$ and $D \in \mathcal{M}_s(\mathbb{R})$.*

Proof. We set $M = \left(V_{c-\mathbb{1}}^1\right)^{-1} V_c^1$. Note that M is in fact independant of the choice of the $(c_i)_{1\leqslant i\leqslant s}$. Indeed, it is the matrix of the linear mapping $P(X) \mapsto P(X+1)$ in the canonical basis of $\mathbb{R}_{s-1}[X]$. This means that the coefficients $(M_{ij})_{1\leqslant i,j\leqslant s}$ of M are given by $M_{ij} = 0$ if $j < i$ and $M_{ij} = \binom{j-1}{i-1}$ otherwise. In particular, it is upper triangular and its diagonal elements are equal to 1. Assuming step (2.2) is consistent of order s, with (2.7), we obtain

$$D = V_{c-\mathbb{1}}^1\left[\left(V_{c-\mathbb{1}}^1\right)^{-1} V_c^1 - \left(V_{c-\mathbb{1}}^1\right)^{-1}\Theta\right]\left(V_{c-\mathbb{1}}^1\right)^{-1}, \tag{2.8}$$

so that the matrix D is similar to $M - Y$, where Y is the matrix $\left(V_{c-\mathbb{1}}^1\right)^{-1}\Theta$. Note that all the coefficients of Y are equal to 0, except maybe on the first column. We shall denote by $y_1, \ldots, y_s$ the coefficients in the first column of Y and by Y_1 the first column of Y. Given distinct $\lambda_1, \ldots, \lambda_s \in \mathbb{C}\backslash\{1\}$, the existence and uniqueness of D and Θ such that step (2.2) has order s and the spectrum of D is exactly $\{\lambda_1, \ldots, \lambda_s\}$ is equivalent to the existence and uniqueness of $y_1, \ldots, y_s \in \mathbb{C}$ such that $M - Y$ has spectrum $\{\lambda_1, \ldots, \lambda_s\}$.

Let us fix $k \in \{1, \ldots, s\}$. The existence of an eigenvector for $M - Y$ for the eigenvalue λ_k is exactly the existence of a nontrivial vector $Z_k \in \mathbb{C}^s$ such that $(M - Y)Z_k = \lambda_k Z_k$. Let us denote by I the identity matrix

of size s and U the upper triangular matrix such that $M = I - U$. Observe that, in view of the definition of M, the entries above the diagonal of U are negative. The relation $(M - Y)Z_k = \lambda_k Z_k$ is equivalent to $(1-\lambda_k)Z_k = (Y+U)Z_k$. Since $YZ_k = z_1^{(k)} Y_1$ where $z_1^{(k)}$ is the first component of Z_k, we infer that the relation $(M - Y)Z_k = \lambda_k Z_k$ is also equivalent to

$$(1-\lambda_k)Z_k = z_1^{(k)} \begin{bmatrix} y_1 \\ \vdots \\ y_s \end{bmatrix} + UZ_k. \tag{2.9}$$

If $z_1^{(k)} = 0$, then, because the matrix U is strictly upper triangular and $\lambda_k \neq 1$, we have $Z_k = 0$ and hence Z_k is not an eigenvector. Therefore, if Z_k is an eigenvector, $z_1^{(k)} \neq 0$ and we can impose without loss of generality that $z_1^{(k)} = 1$. Together with (2.9), this implies, by recursion, that

$$Z_k = \left(\sum_{p=0}^{s-1} \frac{1}{(1-\lambda_k)^{p+1}} U^p \right) \begin{bmatrix} y_1 \\ \vdots \\ y_s \end{bmatrix}. \tag{2.10}$$

The equation on the first line in (2.10) is of the form

$$1 = P_1\left(\frac{1}{1-\lambda_k}\right) y_1 + \cdots + P_s\left(\frac{1}{1-\lambda_k}\right) y_s, \tag{2.11}$$

where for all $i \in \{1, \dots, s\}$, P_i is a polynomial of degree exactly i (remind that the matrix U is strictly upper triangular with negative entries above the diagonal). Moreover, if the relation (2.11) is verified for some $(y_1, \dots, y_s)$, then there exists a solution Z_k of (2.9) with $z_1^{(k)} = 1$: one just has to compute the components of Z_k in (2.10) one after the other to obtain an eigenvector of $M - Y$ for the eigenvalue λ_k. As a summary, we have proved that, for all $k \in \{1, \dots, s\}$, λ_k is an eigenvalue of $M - Y$ if and only if (2.11) holds. Therefore, the fact that the spectrum of $M - Y$ is $\{\lambda_1, \cdots, \lambda_s\}$ is equivalent to the linear system

$$\begin{bmatrix} P_1\left(\frac{1}{1-\lambda_1}\right) & \dots & P_s\left(\frac{1}{1-\lambda_1}\right) \\ \vdots & & \vdots \\ P_1\left(\frac{1}{1-\lambda_s}\right) & \dots & P_s\left(\frac{1}{1-\lambda_s}\right) \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_s \end{bmatrix} = \mathbb{1}. \tag{2.12}$$

Since for all $i \in \{1, \dots, s\}$, the polynomial P_i has degree i and the $(\lambda_j)_{1 \leqslant j \leqslant s} \in (\mathbb{C} \backslash \{1\})^s$ are distinct, the system above is invertible, so that it has a unique solution. Since (2.12) has a unique solution in $\mathbb{C}^s$ and the polynomials $(P_i)_{1 \leqslant i \leqslant s}$ have real coefficients, it is easy to check that, if the set $\{\lambda_1, \dots, \lambda_s\}$ is moreover stable under complex conjugation, then $y_1, \dots, y_s$ are real numbers, and so are $\theta_1, \dots, \theta_s$ and the matrix D has real coefficients using (2.8). This proves the theorem. □

Corollary 2.6. *Assume $c_1, \dots, c_s$ are fixed and distinct as above. Then*

- *There exists $D \in \mathcal{M}_s(\mathbb{R})$ and $\theta_1, \dots, \theta_s \in \mathbb{R}$ such that the step (2.2) is stable and has order s.*
- *There exists $D \in \mathcal{M}_s(\mathbb{R})$ and $\theta_1, \dots, \theta_s \in \mathbb{R}$ such that the step (2.2) is strongly stable and has order s.*

Proof. Choose $\lambda_1, \dots, \lambda_s \in \mathbb{C} \backslash \{1\}$ distinct such that for all i, $|\lambda_i| \leqslant 1$ to obtain a stable method (or for all i, $|\lambda_i| < 1$ to obtain a strongly stable method) in such a way that the set $\{\lambda_1, \dots, \lambda_s\}$ is stable under complex conjugation and apply Theorem 2.5. □

Remark 2.7. In order to actually build the matrix D and the coefficients $\theta_1, \dots, \theta_s$ that define step (2.2) so that this step is stable (respectively strongly stable) and has order s, it is sufficient to fix distinct $c_1, \dots, c_s$ as above, and choose distinct $\lambda_1, \dots, \lambda_s \in \mathbb{C}\backslash\{1\}$ with modulus less (respectively strictly less) than 1, and in such a way that the set $\{\lambda_1, \dots, \lambda_s\}$ is stable under complex conjugation. Then, one forms system (2.12), the rows of which are the first rows of the right-hand side of (2.10) for different values of λ_k, and one solves it for $y_1, \dots, y_s$. One easily computes Θ from Y using the fact that $\Theta = V^1_{c-\mathbb{1}} Y$. In the end, the matrix D is computed using (2.7). Examples are provided in Section 2.4.

2.3. Convergence of the method (2.2)–(2.4)

In this section, we prove that the methods presented above, provided that they involve a step (2.2) with strong stability and order s, and a Runge–Kutta collocation method of order at least s, are indeed convergent in finite time, with order s, when applied to an ODE with sufficiently smooth vector field. We assume the unknown u of equation (2.1) is scalar and that $L = 0$. In fact, up to a change of unknown, any ODE with an equilibrium can be cast into this form:

$$u'(t) = N(u(t))u(t). \tag{2.13}$$

Our methods and results extend to systems of ODEs of the form (2.13) where the unknown u is vector-valued and N is a given smooth matrix-valued function. Similarly, our methods and results extend to the case of complex-valued functions. But for the sake of simplicity we focus on the real valued scalar case.

We assume N is defined and smooth on some open subset Ω of $\mathbb{R}$. We fix $u^0 \in \Omega$. There exists a unique maximal solution to the Cauchy problem (2.13) for $u(0) = u^0$. This solution is defined on an open interval of the form $(T_\star, T^\star)$ with $-\infty \leqslant T_\star < 0 < T^\star \leqslant +\infty$. We fix $T \in (0, T^\star)$. Since the maximal solution is smooth, we have $\sup_{t\in[0,T]} |N(u(t))| < +\infty$. Since it is defined on the compact interval $[0, T]$, one can choose $r > 0$ such that

$$V = \{u(t) + v \mid t \in [0, T],\ v \in \mathbb{R},\ |v| \leqslant r\} \subset \Omega. \tag{2.14}$$

We set $M = \sup_{t\in[0,T]} |N(u(t))|$ and we choose $m > 0$ such that $M + m > \sup_{u\in V} |N(u)|$.

We discretize the time as in Section 2.1, with h small enough to ensure that $T_\star < t_{-1}$. We start by focusing on the consistency of the method. Namely, we set for all $n \in \mathbb{N}$ such that $t_n \leqslant T$,

$$R^1_n = \begin{bmatrix} N(u(t_n + c_1 h)) \\ \vdots \\ N(u(t_n + c_s h)) \end{bmatrix} - D \begin{bmatrix} N(u(t_{n-1} + c_1 h)) \\ \vdots \\ N(u(t_{n-1} + c_s h)) \end{bmatrix} - N(u(t_n)) \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_s \end{bmatrix}. \tag{2.15}$$

Similarly, we define R^2_n as the vector of $\mathbb{R}^s$ with entry number i equal to

$$\left(R^2_n\right)_i = u(t_n + c_i h) - u(t_n) - h \sum_{j=1}^{s} a_{ij} N(u(t_n + c_j h))u(t_n + c_j h), \tag{2.16}$$

and

$$R^3_n = u(t_{n+1}) - u(t_n) - h \sum_{i=1}^{s} b_i N(u(t_n + c_i h))u(t_n + c_i h). \tag{2.17}$$

Lemma 2.8. *Assume that the function N is sufficiently smooth, $u^0 \in \Omega$ and $T \in (0, T^\star)$. Suppose moreover that the numerical coefficients $(c_i)_{1\leqslant i\leqslant s}$, $(a_{i,j})_{1\leqslant i,j\leqslant s}$ and $(b_i)_{1\leqslant i\leqslant s}$ define a Runge–Kutta collocation method of order s and that the step (2.2) is of order s. For any norm on $\mathbb{R}^s$, there exists a constant $C > 0$ such that for a sufficiently small $h > 0$,*

$$\max_{n\geqslant 0,\ t_{n+1}\leqslant T} \left\|R^1_n\right\| \leqslant Ch^s, \tag{2.18}$$

$$\max_{n\geqslant 0,\ t_{n+1}\leqslant T} \left\|R^2_n\right\| \leqslant Ch^{s+1}, \tag{2.19}$$

$$\max_{n\geqslant 0,\ t_{n+1}\leqslant T}\left|R_n^3\right| \leqslant Ch^{s+1}. \tag{2.20}$$

Proof. Let us start with the estimate on R_n^1. First of all we use a Taylor expansion and write for all $1 \leqslant i \leqslant s$

$$N(u(t_n + c_i h)) = \sum_{k=0}^{s-1} \frac{(c_i h)^k}{k!}(N\circ u)^{(k)}(t_n) + \int_0^{c_i h} \frac{(c_i h - \sigma)^{s-1}}{(s-1)!}(N\circ u)^{(s)}(t_n+\sigma)\mathrm{d}\sigma.$$

Let us denote by $X(t_n)$ the vector of $\mathbb{R}^s$ with $(N\circ u)^{(k-1)}(t_n)/(k-1)!$ as component number k. The relation above allows to write

$$\begin{bmatrix} N(u(t_n+c_1h)) \\ \vdots \\ N(u(t_n+c_sh)) \end{bmatrix} = V_c X(t_n) + \begin{bmatrix} \displaystyle\int_0^{c_1h} \frac{(c_1h-\sigma)^{s-1}}{(s-1)!}(N\circ u)^{(s)}(t_n+\sigma)\mathrm{d}\sigma \\ \vdots \\ \displaystyle\int_0^{c_sh} \frac{(c_sh-\sigma)^{s-1}}{(s-1)!}(N\circ u)^{(s)}(t_n+\sigma)\mathrm{d}\sigma \end{bmatrix} := V_c X(t_n) + r_{1,1}^n. \tag{2.21}$$

Similarly, we have

$$\begin{aligned}\begin{bmatrix} N(u(t_{n-1}+c_1h)) \\ \vdots \\ N(u(t_{n-1}+c_sh)) \end{bmatrix} &= V_{c-\mathbb{1}} X(t_n) + \begin{bmatrix} \displaystyle\int_0^{(c_1-1)h} \frac{((c_1-1)h-\sigma)^{s-1}}{(s-1)!}(N\circ u)^{(s)}(t_n+\sigma)\mathrm{d}\sigma \\ \vdots \\ \displaystyle\int_0^{(c_s-1)h} \frac{((c_s-1)h-\sigma)^{s-1}}{(s-1)!}(N\circ u)^{(s)}(t_n+\sigma)\mathrm{d}\sigma \end{bmatrix} \\ &:= V_{c-\mathbb{1}} X(t_n) + r_{1,2}^n. \end{aligned} \tag{2.22}$$

Moreover we have

$$N(u(t_n))\begin{bmatrix}\theta_1 \\ \vdots \\ \theta_s\end{bmatrix} = \Theta X(t_n). \tag{2.23}$$

Mutiplying (2.22) by D and substracting the result and (2.23) to (2.21), we infer that

$$R_n^1 = V_c X(t_n) - DV_{c-\mathbb{1}}X(t_n) - \Theta X(t_n) + r_{1,1}^n - Dr_{1,2}^n.$$

Since the step (2.2) is of order s, we have using (2.6)

$$R_n^1 = r_{1,1}^n - Dr_{1,2}^n.$$

Moreover with (2.7) we have

$$D = V_c^1\big(V_{c-\mathbb{1}}^1\big)^{-1} - \Theta\big(V_{c-\mathbb{1}}^1\big)^{-1},$$

so that D does not depend on h. Let $\|\cdot\|$ be a norm on $\mathbb{R}^s$. The vectors $r_{1,1}^n$ and $r_{1,2}^n$ satisfy

$$\max_{n\geqslant 0,\ t_{n+1}\leqslant T}\big(\|r_{1,1}^n\| + \|r_{1,2}^n\|\big) \leqslant Ch^s,$$

for some $C > 0$ and all sufficiently small $h > 0$. This proves (2.18). Since the Runge–Kutta method with coefficients $a_{i,j}$ and b_i is a collocation method of order at least s at points c_i, the bounds (2.19) and (2.20) are classical (see *e.g.* [20], Sect. II.1.2). □

Theorem 2.9. *Assume that the function N is sufficiently smooth, $u^0 \in \Omega$ and $T \in (0, T^\star)$. Suppose moreover that the numerical coefficients $(c_i)_{1\leqslant i\leqslant s}$, $(a_{i,j})_{1\leqslant i,j\leqslant s}$ and $(b_i)_{1\leqslant i\leqslant s}$ define a Runge–Kutta collocation method of order s and that the step (2.2) is strongly stable and of order s. Provided that r is fixed as in (2.14) and M and m accordingly, there exists constants $C > 0$ and $h_0 > 0$, such that, for all $h \in (0, h_0)$, if the initial data $u_0 \in \Omega$ (respectively $(\gamma_{-1+c_1}, \cdots, \gamma_{-1+c_s}) \in N(V)^s$) is sufficiently close to its exact analogues $u^0 \in \Omega$ (respectively $(N(u(t_{-1}+c_1h)), \cdots, N(u(t_{-1}+c_sh))) \in N(V)^s$) in the sense of relations (2.38) and (2.39), then for all $n \in \mathbb{N}$ such that $t_{n-1} \leqslant T$, the step (2.3) has a unique solution in $\mathbb{R}^s$ and for all $n \in \mathbb{N}$ such that $t_n \leqslant T$,*

$$|\gamma_{n-1+c_i}| \leqslant M + m, \qquad \forall\ i \in [\![1, s]\!] \tag{2.24}$$

$$\max_{k\in[\![0,n]\!]} |u(t_k) - u_k| \leqslant e^{Cnh}\left[|u^0 - u_0| + C\left(\max_{i\in[\![1,s]\!]} |\gamma_{-1+c_i} - N(u(t_{-1}+c_ih))| + h^s\right)\right]. \tag{2.25}$$

Let us first introduce all the notations we use in the proof. We denote by Γ_n the vector of $\mathbb{R}^s$ with component i equal to γ_{n+c_i}. Let us define the convergence errors $P_n \in \mathbb{R}^s$ with component number i equal to $(P_n)_i = N(u(t_n + c_ih)) - \gamma_{n+c_i}$, $Q_n \in \mathbb{R}^s$ with component number i equal to $Q_{n,i} = u(t_n + c_ih) - u_{n,i}$ (provided $u_{n,i}$ is well defined), and $e_n \in \mathbb{R}$ with $e_n = u(t_n) - u_n$. We set $z_n = \max_{0\leqslant k\leqslant n} |e_k|$. We denote by $|\cdot|_\infty$ the norm on $\mathbb{R}^s$ defined as the maximum of the absolute values of the components of the vectors. Moreover, we denote by $\|\cdot\|_\infty$ the norm on $\mathcal{M}_s(\mathbb{R})$ induced by $|\cdot|_\infty$. In the following proof, the letter C denotes a positive real number which does not depend on h (but depends on M and r in particular) and whose value may vary from one line to the other.

Proof. Since step (2.2) is strongly stable we have $\rho(D) < 1$. Therefore, there exists a norm $|\cdot|_D$ on $\mathbb{R}^s$ such that the norm $\|\cdot\|_D$ induced by this norm on $\mathcal{M}_s(\mathbb{R})$ satisfies $\|D\|_D < 1$. In the following we set $\delta = \|D\|_D$. Since $\mathbb{R}^s$ is of dimension s, there exists a $\kappa \in (0, 1]$ such that for all x in $\mathbb{R}^s$, $\kappa|x|_D \leqslant |x|_\infty \leqslant \dfrac{1}{\kappa}|x|_D$.

We divide the proof in two parts. First we assume an *a priori* bound for the numerical solution. Namely we assume that for all n such that $t_n \leqslant T$:

- (H1) $|\Gamma_n|_\infty \leqslant M + m$,
- (H2) the step (2.3) has a unique solution $(u_{n,i})_{1\leqslant i\leqslant s}$ in $\mathbb{R}^s$,
- (H3) $u_n \in V$.

We show that, in this case, we have an explicit bound for the convergence errors P_n and z_n (see Eqs. (2.35) and (2.37)).

Second, we assume that h_0 and the initial errors P_{-1} and e_0 are small enough and we show that the bounds of the first part of the proof are indeed satisfied.

First part. In addition to the bounds above, we assume that $h \in (0, 1)$ and n satisfy $t_{n+1} \leqslant T$. Substracting (2.2) from (2.15) we obtain

$$P_n = DP_{n-1} + (N(u(t_n)) - N(u_n))\begin{bmatrix}\theta_1\\ \vdots\\ \theta_s\end{bmatrix} + R^1_n. \tag{2.26}$$

We infer that

$$|P_n|_D \leqslant \|D\|_D|P_{n-1}|_D + C|e_n| + |R^1_n|_D, \tag{2.27}$$

where the constant C is the product of the Lipschitz constant of N over the compact V times $|(\theta_1, \cdots, \theta_s)^t|_D$ (recall that N is a smooth function over the open set Ω, hence it is Lipschitz-continuous on the compact $V \subset \Omega$).

Substracting (2.3) from (2.16) we obtain

$$Q_{n,i} = e_n + h\sum_{j=1}^{s} a_{i,j}\big(N(u(t_n + c_jh))u(t_n + c_jh) - \gamma_{n+c_j}u_{n,j}\big) + \big(R^2_n\big)_i$$

$$= e_n + h\sum_{j=1}^{s} a_{i,j}\big(N(u(t_n + c_j h)) - \gamma_{n+c_j}\big)u(t_n + c_j h) + h\sum_{j=1}^{s} a_{i,j}\gamma_{n+c_j}(u(t_n + c_j h) - u_{n,j}) + \big(R_n^2\big)_i$$
$$= e_n + h\sum_{j=1}^{s} a_{i,j}P_{n,j}u(t_n + c_j h) + h\sum_{j=1}^{s} a_{i,j}\gamma_{n+c_j}Q_{n,j} + \big(R_n^2\big)_i.$$

We infer that

$$|Q_n|_\infty \leqslant |e_n| + Ch|P_n|_\infty + Ch|\Gamma_n|_\infty|Q_n|_\infty + \big|R_n^2\big|_\infty,$$

which gives with the first point of the assumptions above

$$|Q_n|_\infty \leqslant |e_n| + Ch|P_n|_\infty + Ch(M+m)|Q_n|_\infty + \big|R_n^2\big|_\infty.$$

Provided that $Ch(M+m) \leqslant 1/2$, we have

$$|Q_n|_\infty \leqslant 2|e_n| + Ch|P_n|_\infty + 2\big|R_n^2\big|_\infty. \tag{2.28}$$

Substracting (2.4) from (2.17) we obtain

$$e_{n+1} = e_n + h\sum_{i=1}^{s} b_i(N(u(t_n + c_i h)) - \gamma_{n+c_i})u(t_n + c_i h) + h\sum_{i=1}^{s} b_i\gamma_{n+c_i}(u(t_n + c_i h) - u_{n,i}) + R_n^3$$
$$= e_n + h\sum_{i=1}^{s} b_i P_{n,i}u(t_n + c_i h) + h\sum_{i=1}^{s} b_i\gamma_{n+c_i}Q_{n,i} + R_n^3.$$

We infer that

$$|e_{n+1}| \leqslant |e_n| + Ch|P_n|_\infty + Ch|\Gamma_n|_\infty|Q_n|_\infty + |R_n^3|,$$

which gives with the first point of the assumptions above

$$|e_{n+1}| \leqslant |e_n| + Ch|P_n|_\infty + Ch(M+r)|Q_n|_\infty + |R_n^3|.$$

Using (2.28), we have

$$|e_{n+1}| \leqslant (1+Ch)|e_n| + Ch|P_n|_\infty + Ch\big|R_n^2\big|_\infty + |R_n^3|. \tag{2.29}$$

From (2.27) we have by induction

$$|P_n|_D \leqslant \delta^{n+1}|P_{-1}|_D + C\sum_{k=0}^{n} \delta^{n-k}\big(|e_k| + |R_k^1|_D\big). \tag{2.30}$$

Using the norm equivalence and (2.30) in (2.29) we obtain

$$|e_{n+1}| \leqslant (1+Ch)|e_n| + \frac{Ch}{\kappa}\left[\delta^{n+1}|P_{-1}|_D + C\sum_{k=0}^{n} \delta^{n-k}\big(|e_k| + |R_k^1|_D\big)\right] + Ch\big|R_n^2\big|_\infty + |R_n^3|, \tag{2.31}$$

which gives with Lemma 2.8

$$|e_{n+1}| \leqslant (1+Ch)|e_n| + \frac{Ch}{\kappa}\left[\delta^{n+1}|P_{-1}|_D + C\sum_{k=0}^{n} \delta^{n-k}\big(|e_k| + |R_k^1|_D\big)\right] + Ch^{s+2} + Ch^{s+1}. \tag{2.32}$$

Using the maximal error defined previously and the fact that $\delta < 1$ since the step (2.2) is strongly stable, we have

$$|e_{n+1}| \leqslant (1+Ch)z_n + Ch\left[\delta^{n+1}|P_{-1}|_D + (z_n + h^s)\sum_{k=0}^{n} \delta^{n-k}\right] + Ch^{s+1}$$

$$\leqslant (1+Ch)z_n + Ch\left[\delta^{n+1}|P_{-1}|_D + (z_n+h^s)\frac{1}{1-\delta}\right] + Ch^{s+1},$$

and then

$$|e_{n+1}| \leqslant (1+Ch)z_n + Ch\delta^{n+1}|P_{-1}|_D + Ch^{s+1}. \tag{2.33}$$

Using that $z_{n+1} = \max\{z_n, |e_{n+1}|\}$ and $\delta \in (0,1)$, we infer

$$z_{n+1} \leqslant (1+Ch)z_n + Ch|P_{-1}|_D + Ch^{s+1}. \tag{2.34}$$

By induction it follows that for all n in $\mathbb{N}$ such that $t_n \leqslant T$,

$$\begin{aligned} z_n &\leqslant (1+Ch)^n z_0 + Ch(|P_{-1}|_D + h^s)\sum_{k=0}^{n-1}(1+Ch)^k \\ &\leqslant e^{Cnh}z_0 + Ch(|P_{-1}|_D + h^s)\frac{(1+Ch)^n}{1+Ch-1} \\ &\leqslant e^{Cnh}(z_0 + C(|P_{-1}|_D + h^s)) \\ &\leqslant e^{Cnh}(z_0 + C(|P_{-1}|_\infty + h^s)). \end{aligned} \tag{2.35}$$

Using (2.30) and the same estimations as above, we have moreover

$$|P_n|_D \leqslant |P_{-1}|_D + C(z_n + h^s). \tag{2.36}$$

We infer

$$|P_n|_\infty \leqslant Ce^{Cnh}(z_0 + |P_{-1}|_\infty + h^s). \tag{2.37}$$

Second part. From now on, we denote by C the maximum of the constants appearing in the right hand sides of (2.35) and (2.37). Choose $h_0 \in (0,1)$ sufficiently small to have $h_0 < \min\{-T_\star, T^\star\}$ and $Ce^{CT}h_0^s < r$ and $Ce^{CT}h_0^s < m$ and $h_0\|A\|_\infty(M+m) < 1$. Assume $u_0, \gamma_{-1+c_1}, \ldots, \gamma_{-1+c_s} \in \mathbb{R}$ and $h \in (0, h_0)$ satisfy

$$e^{CT}\left(|u^0 - u_0| + C\left(\max_{i\in[\![1,s]\!]}|\gamma_{-1+c_i} - N(u(t_{-1}+c_ih))| + h_0^s\right)\right) < r, \tag{2.38}$$

and

$$Ce^{CT}\left(|u^0 - u_0| + \max_{i\in[\![1,s]\!]}|\gamma_{-1+c_i} - N(u(t_{-1}+c_ih))| + h_0^s\right) < m. \tag{2.39}$$

First, with (2.37) and (2.39), we have

$$|P_0|_\infty = \max_{i\in[\![1,s]\!]}|\gamma_{0+c_i} - N(u(t_0+c_ih))| < m.$$

Therefore by triangle inequality we have

$$|\Gamma_0|_\infty \leqslant |P_0|_\infty + |(N(u(t_0+c_ih)))_{1\leqslant i\leqslant s}|_\infty < M+m.$$

And then, the hypothesis (H1) of the first part is satisfied for $n=0$.

Moreover, with (2.38), we have $|u^0 - u_0| \leqslant r$ so that $u_0 \in V$ and the hypothesis (H3) of the first part is satisfied with $n=0$. We infer that the system (2.3) (with $L=0$) has a unique solution in $\mathbb{R}^s$ since we assumed $h \leqslant h_0 < 1/(\|A\|_\infty(M+m))$. This implies that the hypothesis (H2) of the first part is satisfied for $n=0$. Then we can apply the analysis of the first part to obtain (2.35) and (2.37) with $n=1$. Using (2.38) and (2.39), we infer that hypotheses (H1)–(H3) are satisfied with $n=1$ and the result follows by induction on n.

□

Remark 2.10. The proof may look like following the usual strategy for the proof of convergence of a numerical method. However, note that the definition of strong stability (Def. 2.2) plays a central role in the proof, and this is not the case in classical theory. Moreover, estimations such as (2.30) in (2.29) to obtain (2.31) are *not* classical.

Remark 2.11. Given $u_0 \in \Omega$ close to $u^0 \in \Omega$, before starting the time-stepping method (2.3) and (2.4), one needs to first compute the s approximations $(\gamma_{-1+c_i})_{1\leqslant i\leqslant s}$ of $(N(u((-1+c_i)h)))_{1\leqslant i\leqslant s}$. These quantities enter the error estimate (2.25). This computation can be done efficiently by determinating s approximations of $(u((-1+c_i)h))_{1\leqslant i\leqslant s}$ using a sufficiently high order method for (2.1) backwards in time, provided the equation makes sense. Alternatively, for example for the nonlinear heat equation, for which running (2.1) backwards in time makes no sense, one can use a sufficiently high order method from u_0 to compute $s+1$ approximations of $(u(c_ih))_{1\leqslant i\leqslant s}$ and $u(h)$ over one time step, and then start the linearly implicit method (2.3) and (2.4) from time h until time T.

2.4. Examples of linearly implicit methods

In this section we present possible choices of methods of order 1, 2, 4 and 6. The general building procedure is the following: We choose $s \in \mathbb{N}^\star$, we fix $0 \leqslant c_1 < c_2 < \cdots < c_s \leqslant 1$ and we compute $a_{i,j}$ and b_i for $1 \leqslant i, j \leqslant s$ using the formulas

$$a_{i,j} = \int_0^{c_i} \mathcal{L}_j(\tau)\mathrm{d}\tau \qquad \text{and} \qquad b_i = \int_0^1 \mathcal{L}_i(\tau)\mathrm{d}\tau,$$

where $\mathcal{L}_i(\tau) = \prod_{\substack{k=1\\k\neq i}}^{s} \dfrac{(\tau - c_k)}{(c_i - c_k)}$ is the ith Lagrange polynomial at points $c_1, \ldots, c_s$. This way, the coefficients $a_{i,j}$, b_i and c_i are those of a Runge–Kutta collocation method. Next we choose $\lambda_1, \ldots, \lambda_s \in \mathbb{C}\backslash\{1\}$ with moduli strictly less than 1, all distincts and in such a way that the set $\{\lambda_1, \ldots, \lambda_s\}$ is invariant under complex conjugation. We compute the polynomials $P_1, \ldots, P_s$ appearing in (2.11) defined using (2.10) in the proof of Theorem 2.5. We solve (2.12) for $y_1, \ldots, y_s$ and compute $\theta_1, \ldots, \theta_s$ using $\Theta = \left(V_{c-\mathbb{1}}^1\right)Y$. Finally, we compute the matrix D using (2.7). This way, we define a step (2.2) that is strongly stable and of order s (see Defs. 2.2 and 2.4). Using Theorem 2.9, the numerical method (2.2)–(2.4) is convergent of order s.

A linearly implicit method of order 1: we choose $s = 1$ and $c_1 = 1$ so as to rely on the implicit Euler method. Then $a_{1,1} = 1$ and $b_1 = 1$. Choosing $\lambda_1 = 1/2$, we have $y_1 = 1/2$ and $\theta_1 = 1/2$.

Two linearly implicit methods of order 2:

1 – With Gauss points: for $s = 2$ and the Gauss points $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}, c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$. Then the Runge–Kutta collocation method has Butcher tableau

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & 1/4 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

2 – With uniform points: for $s = 2$ and the uniform points $c_1 = 0, c_2 = 1$. Then the Runge–Kutta collocation method has Butcher tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

For the two cases, we choose $\lambda_1 = 1/2, \lambda_2 = -1/2$. This leads to $y_1 = 2, y_2 = 3/4$ and $\theta_1 = y_1 + (c_1 - 1)y_2, \theta_2 = y_1 + (c_2 - 1)y_2$.

A linearly implicit method of order 4: we choose $s = 4$ and $c_1 = 0, c_2 = 1/3, c_3 = 2/3, c_4 = 1$. Then the Runge–Kutta collocation method has Butcher tableau

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/8 & 19/72 & -5/72 & 1/72 \\ 2/3 & 1/9 & 4/9 & 1/9 & 0 \\ 1 & 1/8 & 3/8 & 3/8 & 1/8 \\ \hline & 1/8 & 3/8 & 3/8 & 1/8 \end{array}.$$

Choosing $\lambda_1 = 0, \lambda_2 = 1/4, \lambda_3 = 1/2, \lambda_4 = 3/4$ for we have

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 5/2 \\ 117/64 \\ 11/32 \\ 1/64 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1235/864 \\ 833/432 \\ 5/2 \end{pmatrix}.$$

A linearly implicit method of order 6: we choose $s = 6$ and $(c_i)_{1 \leqslant i \leqslant 6}$ a uniform subdivision of $[0, 1]$. Then the Runge–Kutta collocation method has Butcher tableau

$$\begin{array}{c|cccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/5 & 19/288 & 1427/7200 & -133/1200 & 241/3600 & -173/7200 & 3/800 \\ 2/5 & 14/225 & 43/150 & 7/225 & 7/225 & -1/75 & 1/450 \\ 3/5 & 51/800 & 219/800 & 57/400 & 57/400 & -21/800 & 3/800 \\ 4/5 & 14/225 & 64/225 & 8/75 & 64/225 & 14/225 & 0 \\ 1 & 19/288 & 25/96 & 25/144 & 25/144 & 25/96 & 19/288 \\ \hline & 19/288 & 25/96 & 25/144 & 25/144 & 25/96 & 19/288 \end{array}.$$

Choosing $\lambda_k = \dfrac{e^{i(k-1)\frac{\pi}{3}}}{2}$ for $k = 1, \ldots, 6$, we have

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 6 \\ 2783/320 \\ 1239/256 \\ 659/512 \\ 43/256 \\ 21/2560 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{pmatrix} = \begin{pmatrix} 65/64 \\ 193389/125000 \\ 1133667/500000 \\ 1608733/500000 \\ 1111047/250000 \\ 6 \end{pmatrix}.$$

Remark 2.12. In the linearly implicit methods given as examples above, the choice of $(\lambda_i)_{1 \leqslant i \leqslant s}$ is somehow arbitrary, and the only condition we impose is that they ensure that step (2.2) is strongly stable (see Def. 2.2). This implies that these methods are convergent for ODEs (see Thm. 2.9). In order to ensure additional features of a linearly implicit method, this choice has to be made carefully (see *e.g.* Sect. 3.3 a linearly implicit method that preserves non-negativity and energy-dissipation for a nonlinear heat equation). For a general class of evolution PDE, the choice of the collocation method as well as that of $(\lambda_i)_{1 \leqslant i \leqslant s}$ has to be made carefully, in particular to ensure a taylored stability property of the linearly implicit method. This question will be investigated in a forthcoming paper.

3. Numerical experiments

In this section, we illustrate the properties of the methods described in Section 2.1 and analysed in Section 2.3. We first present numerical examples on ODEs, with a scalar case in Section 3.1. In particular, we illustrate the results above, such as Theorem 2.9, for several methods introduced above, and we compare the results we obtain with that obtained using other classical numerical methods. Then, we present numerical experiments for PDEs that fit the framework used in Section 2.1 but do not fit *stricto sensu* the framework of the analysis carried out in Section 2.3. This allows for comparison with classical methods for the same problems anyway. We first focus on a nonlinear Schrödinger equation in Section 3.2 and then move to a nonlinear heat equation in Section 3.3. The methods described and analysed in this paper would also be relevant for several other examples of semilinear evolution equation of the form (2.1).

When comparing the efficiency of numerical methods in this section, we consider as a measure of performance the (lowest possible) CPU time required to achieve a given precision on the numerical result. This CPU time has indeed disadvantages since it depends on the algorithms used to solve the problems, the software used to implement the algorithms and the machine on which the software is run. However, we believe one cannot talk about efficiency whithout taking into account some form of CPU time. And, for reproducibility issues, we detail below as much as possible which discretizations and algorithms are used to implement the numerical methods that we consider. Moreover, we try to be as fair as possible when implementing methods from the litterature to compare them with the linearly implicit methods introduced in this paper.

As we shall see in this section, the efficiency of the linearly implicit methods introduced in this paper is similar to that of classical one-step methods with constant step size from the literature (see Sect. 3.1).

In contrast, the linealy implicit methods sometimes outperform standard methods when applied to several evolution PDE problems, that we consider, once discretized, as high dimensional systems of ODEs (see Sects. 3.2 and 3.3). In the following, the computations are carried out using MATLAB and the linear systems are solved using the backslash MATLAB command. In particular, we do not build a taylored method to solve the linear systems numerically and the gain in computational time one can obtain using linearly implicit methods can surely be improved using taylored methods depending on the matrix structures. This choice is not optimal in terms of efficiency, but is fairly similarly done for all the methods below.

3.1. Application to a scalar nonlinear ODE

We consider the scalar ODE

$$u'(t) = -u(t) - u^2(t). \tag{3.1}$$

This corresponds to taking L as minus the identity operator and $N(u) = -u$ in (2.1). The exact maximal solution starting from $u_0 > 0$ at $t = 0$ is given for $t \geqslant 0$ by

$$u(t) = \frac{1}{\left(\frac{1}{u_0} + 1\right)\mathrm{e}^t - 1}.$$

We start with methods of order 1. We use the linearly implicit method of order 1 introduced in Section 2.4. We compare the results we obtain on the problem above with the Euler implicit and explicit schemes as well as the Lie splitting method. We choose $u^0 = u_0 = 1/3$, $\gamma_{-1+c_1} = \gamma_0 = N(u^0)$ and the final time $T = 2$. The results are displayed in Figure 1. The global error is defined as z_N (with the notations of the proof of Thm. 2.9) at final time T with N such that $Nh = T$, where h is the time step. Numerical experiments indicate that the four schemes are of order 1. For the linearly implicit scheme, this is a consequence of Theorem 2.9. Moreover the CPU time required to reach a given numerical error is much lower for the Lie splitting than for the linearly implicit method and for the linearly implicit method than for the explicit Euler scheme and the implicit Euler scheme.

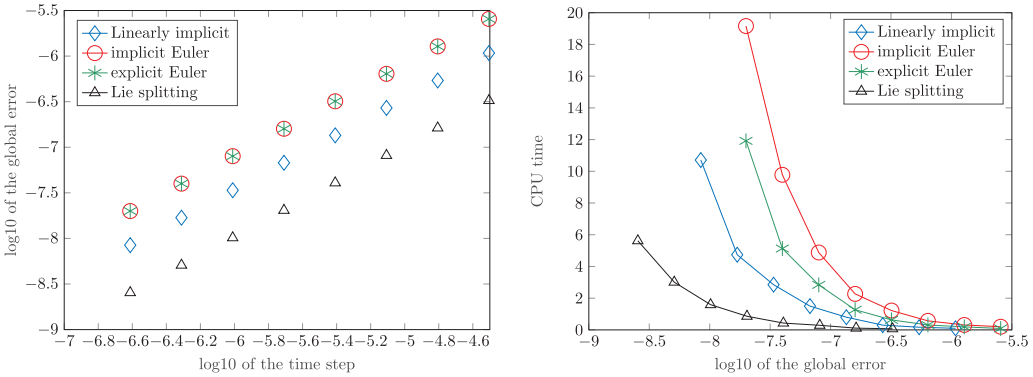

FIGURE 1. Comparison of methods of order 1 applied to (3.1): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

We then consider methods of order 2. We compare the linearly implicit method of order 2 defined in Section 2.4 for Gauss points with other methods of the literature: the midpoint method with Butcher tableau

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array},$$

the RK2 method with Butcher tableau

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array},$$

and the Strang splitting method. We choose $u^0 = u_0 = 0.9$, $\gamma_{-1+c_1} = N(u((-1+c_1)h))$, $\gamma_{-1+c_2} = N(u((-1+c_2)h))$ and the final time $T = 2$.

The results are displayed in Figure 2. Once again the four methods are of order 2. This is a consequence of Theorem 2.9 for the linearly implicit method. The CPU time required for a given numerical error is much lower for the Strang splitting scheme than for the other three methods which perform similarly.

Similar results are obtained (but not displayed here) for methods of order four and six which illustrate Theorem 2.9 for the corresponding linearly implicit methods introduced in Section 2.4. Moreover the CPU time required to reach a given numerical error is always higher for the linearly implicit schemes than for other classical methods of the same order for the ODE (3.1).

3.2. Application to the nonlinear Schrödinger equation

3.2.1. One dimensional nonlinear Schrödinger equation: The soliton case

In this section, we consider the nonlinear one dimensional Schrödinger equation:

$$i\partial_t u = -\partial_x^2 u - q|u|^2 u, \tag{3.2}$$

which corresponds to the evolution problem (2.1) with $L = i\partial_x^2$ and $N(u) = iq|u|^2$. We consider the initial condition

$$u_0(x) = \sqrt{\frac{2a}{q}}\operatorname{sech}\big(\sqrt{a}x\big),$$

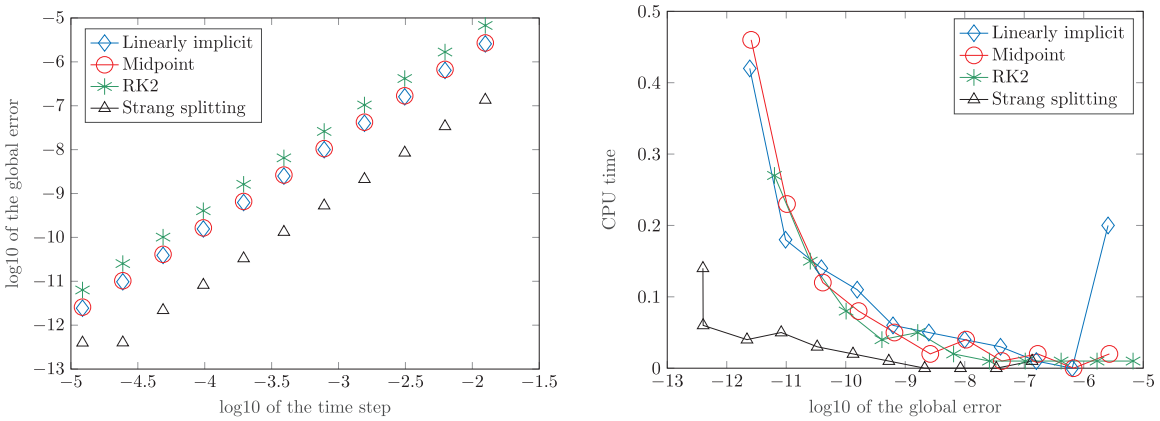

FIGURE 2. Comparison of methods of order 2 applied to (3.1): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

where $q > 0$ and $a = q^2/16$, so that the corresponding exact solution of (3.2) is the zero speed soliton and reads

$$u(t,x) = \sqrt{\frac{2a}{q}}\operatorname{sech}\big(\sqrt{a}x\big)\exp(iat). \tag{3.3}$$

We use $q = 4$ and $a = 1$ for the numerical simulations. The final time is set to $T = 5$. For the space discretization, we consider the interval $[-50, 50]$ with homogeneous Dirichlet boundary conditions since the exact solution (3.3) decays very fast when $|x|$ tends to $+\infty$. We use 2^{14} equispaced points in space for methods of order 1 in time and 2^{18} equispaced points in space for methods of order 2 in time. For splitting methods, one has to integrate numerically equation (3.2) with $q = 0$. This is done *via* the approximation

$$\exp(ihB) = \left(I + i\frac{h}{2}B\right)\left(I - i\frac{h}{2}B\right)^{-1} + \mathcal{O}(h^3), \tag{3.4}$$

where I denotes the identity matrix, B the Laplacian with Dirichlet boundary conditions matrix, and $h > 0$ the small time step. In particular, the splitting methods we use below are also linearly implicit (the nonlinear part of the equation is integrated exactly). The numerical error we consider for all numerical methods is the discrete L^2-norm of the difference between the numerical solution and the projection of the exact solution (3.3) on the space grid at final time T.

First we compare methods of order one. We consider the linearly implicit method of order 1 introduced in Section 2.4, the implicit Euler scheme and the Lie splitting scheme. For the linearly implicit method, we initialize the scheme with $u^0 = u_0 = u(0,\cdot)$, $\gamma_{-1+c_1} = N(u((-1+c_1)h,\cdot))$. The results are given in Figure 3. The figure on the left hand side shows that the three methods are of order 1. This illustrates the fact that the conclusion of Theorem 2.9 for the linearly implicit method holds numerically in this PDE context. For such a simulation, we can see, on the figure on the right hand side, that now the CPU time required to reach a given error is smaller for the linearly implicit method than for the fully implicit Euler method. However the Lie splitting method is the least time consuming method since it is explicit (in fact our implementation of the Lie splitting method makes it linearly implicit, see (3.4)) and has a good error constant.

We then consider methods of order 2. For this experiment we use the linearly implicit method of order 2 introduced in Section 2.4 for the uniform points, the Crank-Nicolson scheme [13, 15] (for which we solve the nonlinear system using a fixed point algorithm) and the Strang splitting method [27]. For the implementation

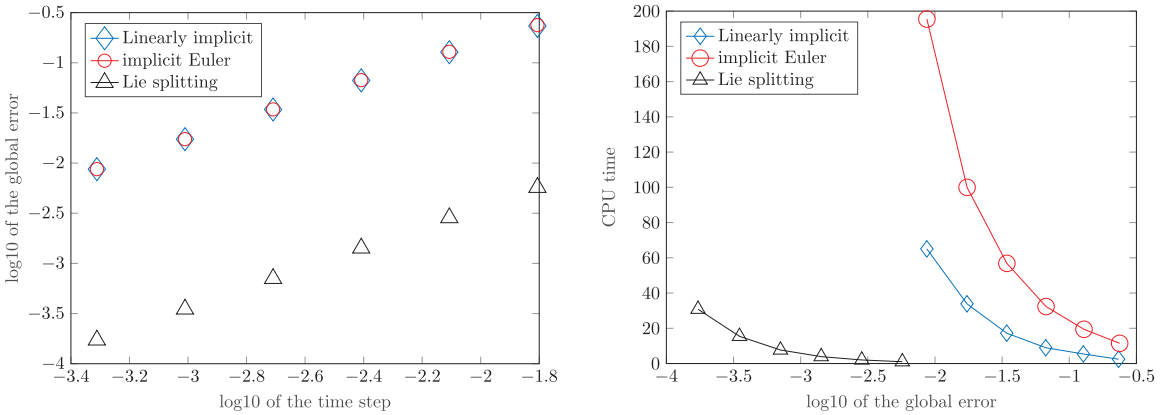

FIGURE 3. Comparison of methods of order 1 applied to (3.2): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

of the Strang splitting method, we use the classical conjugation with the Lie splitting method which we recall briefly below and relies on the identity

$$\left(\exp\left(i\frac{h}{2}B\right)\circ\Phi_h\circ\exp\left(i\frac{h}{2}B\right)\right)^k=\exp\left(i\frac{h}{2}B\right)\circ\left(\Phi_h\circ\exp(ihB)\right)^k\circ\exp\left(-i\frac{h}{2}B\right),\tag{3.5}$$

where Φ_h denotes the numerical flow of the nonlinear part of (3.2) defined componentwise using the function $v\mapsto\exp(ihq|v|^2)v$, and k is any nonnegative integer. The numerical flow of the Lie splitting method is $\Phi_h^{\mathrm{Lie}}=\Phi_h\circ\exp(ihB)$ and that of the Strang splitting method is $\Phi_h^{\mathrm{Strang}}=\exp(ihB/2)\circ\Phi_h\circ\exp(ihB/2)$. Therefore, relation (3.5) also reads

$$\left(\Phi_h^{\mathrm{Strang}}\right)^k=\exp\left(i\frac{h}{2}B\right)\circ\left(\Phi_h^{\mathrm{Lie}}\right)^k\circ\exp\left(-i\frac{h}{2}B\right),\tag{3.6}$$

for all nonnegative integer k. The implementation of the Strang splitting method we use for numerical simulations uses both the right hand side of the equation (3.6) and the approximation formula (3.4). For the linearly implicit method, we initialize the scheme with $u^0=u_0=u(0,\cdot)$, $\gamma_{-1+c_1}=N(u((-1+c_1)h,\cdot))$, $\gamma_{-1+c_2}=N(u((-1+c_2)h,\cdot))$. The results are displayed in Figure 4. The numerical order of each method is the one expected *i.e.* 2. This illustrates once again the numerical relevance of Theorem 2.9 beyond the ODE context. As we can see on the figure on the right hand side, the CPU time required to reach a given error for the Crank-Nicolson method is higher than the one for the linearly implicit method of order 2 which is also a little higher than the one for the Strang splitting method.

Remark 3.1. For this example, if instead of using the linearly implicit method of order 2 with uniform points, we use the linearly implicit method of order 2 with Gauss points, we numerically obtain the superconvergence of the method and a numerical order equal to 4. This is due to the fact that in this particular case the modulus of the solution is constant in time as it can be seen on (3.3).

3.2.2. Two dimensional nonlinear Schrödinger equation

In this section, we consider the following 2D nonlinear Schrödinger equation:

$$i\partial_t u=-\Delta u-|u|^2u,\tag{3.7}$$

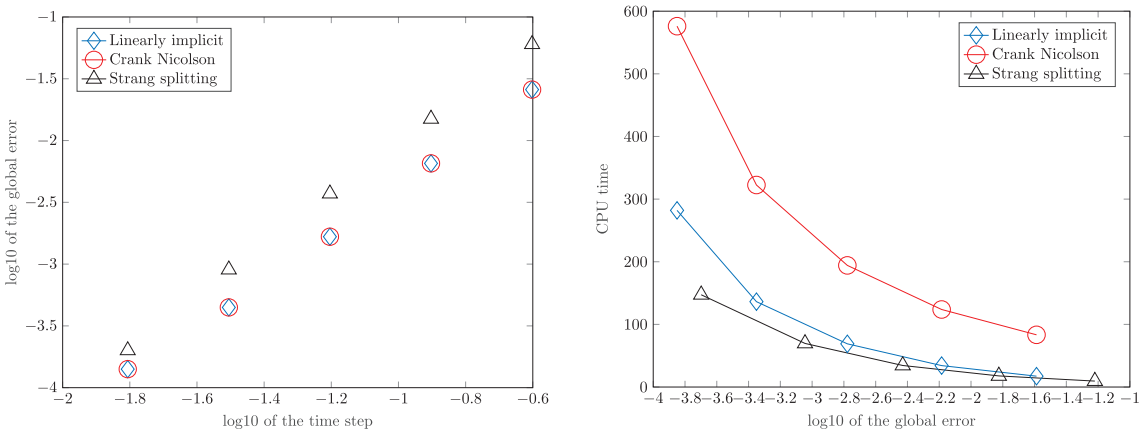

FIGURE 4. Comparison of methods of order 2 applied to (3.2): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

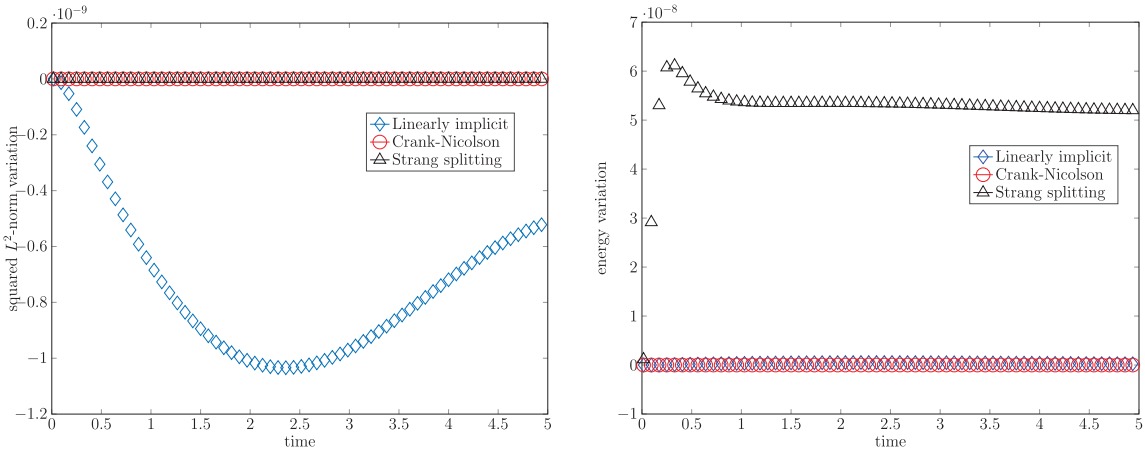

FIGURE 5. Comparison of methods of order 2 for mass and energy conservations: *On the left hand side*: variation of the mass with respect to time; *on the right hand side*: variation of the energy with respect to time.

with homogeneous Dirichlet boundary conditions on the domain represented in gray in the Figure 6 with $l_x = l_y = 1$, $p_x = 2$ and $p_y = 3$. The initial datum we chose reads

$$u_0(x, y) = \sin(2\pi x)\sin(2\pi y)\exp(2i\pi x), \tag{3.8}$$

when (x, y) belongs to the domain. In this particular case, the spectrum of the Laplace operator is not accessible and one cannot use efficiently spectral methods such as exponential Runge–Kutta methods or Lawson methods [6].

We use a finite differences discretization in space with, for $J \in \mathbb{N}^*$, $p_x J + 1$ points in the x-direction and $p_y J + 1$ points in the y-direction in such a way that the step is the same in the two directions. This way, the numerical unknown u_n at time t_n is a vector of $\mathcal{N} = ((p_y - 1)J - 1) \times (p_x J - 1) + J \times ((p_x - 1)J - 1)$ complex numbers. No matter the way we label the unkowns, the matrix B of the Laplace operator with homogeneous

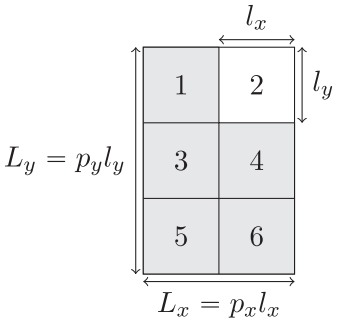

FIGURE 6. 2D domain used in the simulation of the nonlinear Schrödinger equation (3.7).

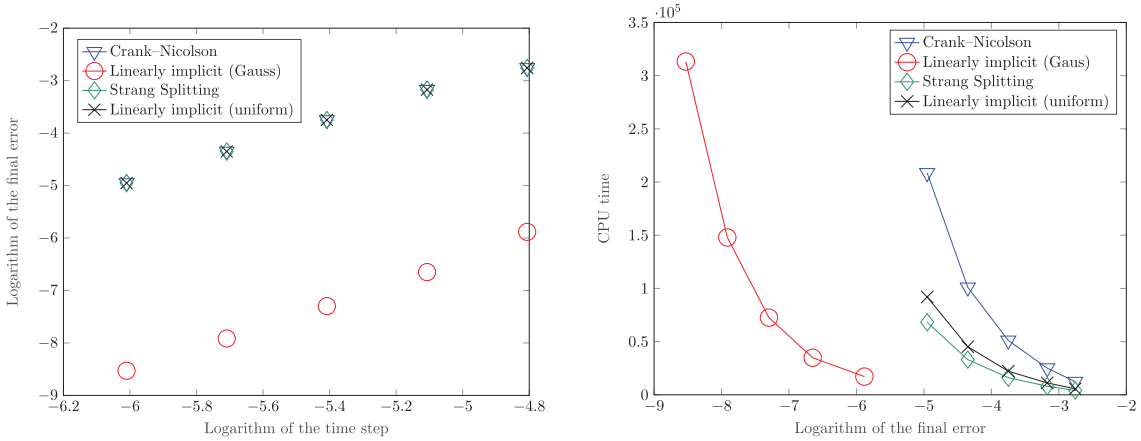

FIGURE 7. Comparison of methods of order 2 applied to (3.7): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

Dirichlet boundary conditions is a sparse matrix of size $\mathcal{N} \times \mathcal{N}$. For the numerical simulations we use $J = 50$ which gives $\mathcal{N} = 12\,251$ unknowns. Moreover, we consider $T = 0.5$ as a final time. The methods we consider are the two linearly implicit methods of order 2 introduced in Section 2.4 (one with Gauss points, the other one with uniform points), the Crank-Nicolson scheme and the Strang splitting method. As one has no direct access to the exact solution of (3.7) with initial condition (3.8), we precompute as a reference solution the numerical solution provided by a Runge–Kutta method at Gauss points with 5 stages (which therefore has order 10) with a time step of 10^{-3}. We initialize the linearly implicit methods with γ_{-1+c_1} and γ_{-1+c_2} computed using one step of a backward Crank-Nicolson scheme. Our numerical results are displayed in Figure 7. As expected, the order of each method above is 2. For the two linearly implicit methods, this again illustrates that the results of Theorem 2.9 extend numerically to this PDE case. Note that, the constant of order is really better for the linearly implicit method with Gauss points than all the other ones. Moreover the CPU time required to achieve a given precision is also smaller for the linearly implicit method with Gauss points. This is the first example where a linearly implicit method developped in this paper clearly outperforms implicit as well as explicit standard methods from the literature.

Remark 3.2 (Preservation of mass in NLS equations by linearly implicit methods)**.** If we consider a linearly implicit method defined by a Runge–Kutta collocation method of order s satisfying the Cooper condition

$$\forall (i,j) \in \{1,\cdots,s\}^2, \qquad b_i b_j = b_i a_{i,j} + b_j a_{j,i}, \tag{3.9}$$

that is to say a Runge–Kutta collocation method at Gauss' points, then this linearly implicit method preserves the mass (*i.e.* the squared L^2-norm) for the NLS equation, regardless of the physical dimension of the problem. This property does not depend on the choice of the matrix D and vector $(\theta_1,\cdots,\theta_s)$ in step (2.2) as long as they are real-valued. This preservation property relies on the fact that, provided the initial $(\gamma_{-1+c_i})_{1\leqslant i\leqslant s}$ are purely imaginary, so will be all the $(\gamma_{n+c_i})_{1\leqslant i\leqslant s}$. This property will be detailed further in a forthcoming paper dealing with the time integration of PDEs using linearly implicit methods. For example, the linearly implicit methods of order 1, 4 and 6 from Section 2.4 do not satisfy the Cooper condition (3.9), hence they do not preserve the mass for the NLS equation. In contrast, the first method of order 2 of Section 2.4, which uses Gauss' points, preserves the mass of the solution of the NLS equation (and the second method of order 2 does not).

3.3. Application to the nonlinear heat equation

In the previous sections, we have proved and illustrated that the linearly implicit methods developed in this paper have good quantitative properties. The goal of this section is to illustrate that they can indeed also have good qualitative properties. Indeed, on some nonlinear heat equation with gradient-flow structure, we give an example below of a linearly implicit fully discrete scheme which preserves the nonnegativity of the solution (just as the exact flow does) as well as the decay of a discrete energy which is consistent with the continuous energy of the problem.

Let us consider the one dimensional nonlinear heat equation given by

$$\partial_t u = \partial_x^2 u + u^3, \tag{3.10}$$

with homogeneous Dirichlet boundary conditions on $\Omega = (-50,50)$. This corresponds to equation (2.1) with $L = \partial_x^2$ and $N(u) = u^2$. Equation (3.10) is the L^2-gradient flow equation for the energy:

$$E(u) = \frac{1}{2}\int_\Omega (\partial_x u)^2 \mathrm{d}x - \frac{1}{4}\int_\Omega u^4 \mathrm{d}x, \tag{3.11}$$

defined for $u \in H_0^1(\Omega)$. It is well-known in the literature (see *e.g.* [21]) that

Theorem 3.3. *For all $u_0 \in H_0^1(\Omega)$, $u_0 \neq 0$, the equation (3.10) has a unique maximal solution u in $\mathcal{C}^0([0,T_*),H_0^1(\Omega)) \cap \mathcal{C}^1((0,T_*),L^2(\Omega))$ with $u(0) = u_0$ for some $T_* > 0$. Moreover this solution u satisfies*

$$\forall\ t \in (0,T_*), \qquad \frac{\mathrm{d}E(u(t))}{\mathrm{d}t} \leqslant 0. \tag{3.12}$$

Finally if $u_0 \geqslant 0$ on Ω then for all $t \in [0,T_*)$, $u(t) \geqslant 0$ on Ω.

In order to give an example with good qualitative properties, we consider a modified version of the fully discrete one stage method presented in Section 2.4 with $s = 1$, $c_1 = 1/2$ so that $a_{1,1} = 1/2$ and $b_1 = 1/2$, and with $\lambda_1 = 1/2$ so that $y_1 = 1/2$ and $\theta_1 = 1/2$.

Let us denote by $\mathcal{N}$ the number of unknowns, so that $\delta x = 100/(\mathcal{N}+1)$. We denote by $\langle\cdot,\cdot\rangle$ the scalar product on $\mathbb{R}^\mathcal{N}$ defined for $v,w \in \mathbb{R}^\mathcal{N}$ by $\langle v,w\rangle = \delta x \sum_{k=1}^\mathcal{N} v(k)w(k)$ and by $\|\cdot\|_2$ the associated norm. Moreover, for all $v \in \mathbb{R}^\mathcal{N}$, we denote by $v^{\circ 2}$ the vector of $\mathbb{R}^\mathcal{N}$ with component k equal to $v^{\circ 2}(k) = v(k)^2$.

Then, the stage (2.2) reads here

$$\gamma_{n+1/2} = \frac{1}{2}\gamma_{n-1/2} + \frac{1}{2}u_n^{\circ 2}, \tag{3.13}$$

and the stages (2.3), and (2.4) can be summarized by

$$\frac{u_{n+1}-u_n}{h} = \big(B + \mathrm{diag}(\gamma_{n+1/2})\big)\frac{u_{n+1}+u_n}{2}, \tag{3.14}$$

where B denotes the matrix of the Laplacian operator with homogeneous Dirichlet boundary conditions on Ω on the equispaced grid, with space step size δx, as defined after (3.4). We still denote by u_0 the evaluation of the initial datum u_0 on the equispaced grid. In addition, we choose for $\gamma_{-1/2}$ the evaluation of $N(u_0)$ on the same grid.

The fully discrete energy associated to the numerical scheme is defined for $u, \gamma \in \mathbb{R}^{\mathcal{N}}$ by

$$E_{rlx}(u,\gamma) = -\frac{1}{2}\langle u, Bu\rangle - \frac{1}{2}\big\langle \gamma, u^{\circ 2}\big\rangle + \frac{1}{4}\langle \gamma, \gamma\rangle. \tag{3.15}$$

Note that this formula is consistent with the continuous energy E defined in (3.11).

Theorem 3.4. *Let us assume $u_0 \in H_0^1(\Omega)$. Still denote by u_0 the projection of u_0 onto the equispaced grid with $\mathcal{N}$ interior points. Choose $T \in (0, T_*)$.*

(1) *Let us assume that there exists $h_0, \delta x_0 > 0$ such that for all $h \in (0, h_0)$ and all $\delta x \in (0, \delta x_0)$ with $h < \delta x^2$, the sequence $(\gamma_{n+1/2})_{n\geqslant 0}$ is bounded in $\mathbb{R}^{\mathcal{N}}$ with the maximum norm as long as $(n+1/2)h \leqslant T$. Then, for all $h \in (0, h_0)$, $\delta x \in (0, \delta x_0)$ and n such that $(n+1/2)h \leqslant T$ and $h/\delta x^2 < 1$, $\gamma_{n+1/2}$ is a nonnegative real-valued vector. Moreover assuming $u_0 \geqslant 0$, there exists a constant $h_1 \in (0, h_0)$ such that for all $h \leqslant h_1$ and $\delta x \in (0, \delta x_0)$ with $h/\delta x^2 < 1$, the sequence $(u_n)_{n\geqslant 0}$ is a sequence of nonnegative vectors as long as $nh \leqslant T$.*

(2) *For all $h \in (0, h_0)$ and for all $n \in \mathbb{N}$ such that $(n+1)h \leqslant T$, the sequence $(u_n, \gamma_{n-1/2})_{n\geqslant 0}$ satisfies*

$$E_{rlx}(u_{n+1}, \gamma_{n+1/2}) \leqslant E_{rlx}(u_n, \gamma_{n-1/2}). \tag{3.16}$$

Proof. We will use M, P and Z matrices as defined for example in the Chapter 10 of [4]. Let us give the main ideas of the proof of Proposition 3.4.

(1) The sign of $\gamma_{n+1/2}$ is a direct consequence of (3.13) and the choice of the initial condition $\gamma_{-1/2} = N(u_0)$: it is a convex combination of vectors with same signs. Moreover assuming $u_0 \geqslant 0$, the nonnegativity of u_n can be obtained by induction using the following arguments. The equation (3.14) can be written

$$\Big(1 - \frac{h}{2}B - \frac{h}{2}\mathrm{diag}\big(\gamma_{n+1/2}\big)\Big)u_{n+1} = \Big(1 + \frac{h}{2}B + \frac{h}{2}\mathrm{diag}\big(\gamma_{n+1/2}\big)\Big)u_n. \tag{3.17}$$

Since $h/\delta x^2 < 1$ and $u_n \geqslant 0$, one has $\Big(1 + \frac{h}{2}B\Big)u_n \geqslant 0$. Since $u_n \geqslant 0$ and $\gamma_{n+1/2} \geqslant 0$, one has that $\mathrm{diag}\big(\gamma_{n+1/2}\big)u_n \geqslant 0$, so that the right-hand side of (3.17) is nonnegative componentwise. Moreover, the operator in the left-hand side of (3.17) has nonnegative inverse since it is an M-matrix for $h_1 \in (0, h_0)$ small enough (depending on the bound on the maximum norm of the sequence $(\gamma_{n+1/2})_{n\geqslant 0}$). Indeed, one can check that it is a Z-matrix since its off-diagonal coefficients are nonpositive, and it is also a P-matrix (for $h \in (0, h_1)$).

(2) Taking the scalar product of (3.14) with $u_{n+1} - u_n$ we obtain

$$\frac{1}{h}\|u_{n+1}-u_n\|_2^2 = \frac{1}{2}\langle u_{n+1}, Bu_{n+1}\rangle - \frac{1}{2}\langle u_n, Bu_n\rangle + \frac{1}{2}\big\langle \gamma_{n+1/2}, u_{n+1}^{\circ 2} - u_n^{\circ 2}\big\rangle,$$

which gives

$$\frac{1}{h}\|u_{n+1}-u_n\|_2^2 = -E_{rlx}(u_{n+1}, \gamma_{n+1/2}) + E_{rlx}(u_n, \gamma_{n-1/2})$$

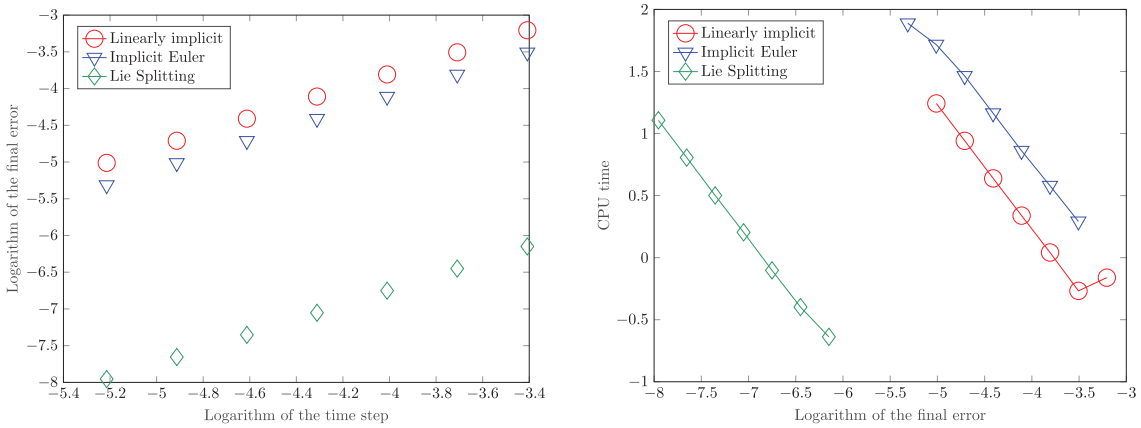

FIGURE 8. Comparison of methods of order 1 applied to (3.10): *On the left hand side*: maximal numerical error as a function of the time step (logarithmic scales); *on the right hand side*: CPU time (in seconds) as a function of the maximal numerical error.

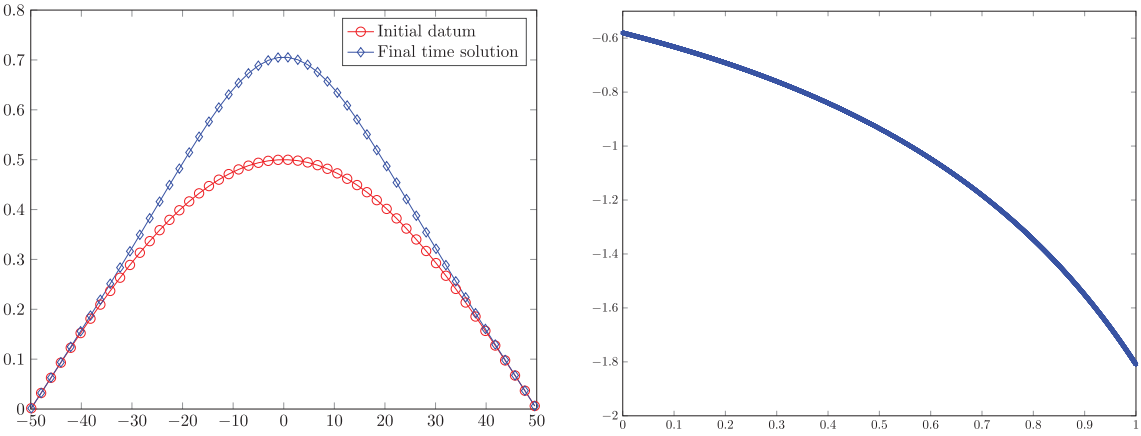

FIGURE 9. Initial datum and solution at the final time $T = 1$ with respect to space (*left hand side*) and evolution of E_{rlx} with respect to time (*right hand side*).

$$+\left\langle \gamma_{n+1/2}, \frac{1}{4}\gamma_{n+1/2} - \frac{1}{2}u_n^{\circ 2} \right\rangle + \left\langle \gamma_{n-1/2}, -\frac{1}{4}\gamma_{n-1/2} + \frac{1}{2}u_n^{\circ 2} \right\rangle.$$

Then, using (3.13), a straightforward computation leads to

$$\frac{1}{h}\|u_{n+1} - u_n\|_2^2 + \frac{3}{4}\|\gamma_{n+1/2} - \gamma_{n-1/2}\|_2^2 = -E_{rlx}\big(u_{n+1}, \gamma_{n+1/2}\big) + E_{rlx}(u_n, \gamma_{n-1/2}), \tag{3.18}$$

which implies the result (3.16). □

We display in Figure 8 the comparison of the method above with the Lie splitting method, with the linear part approximated by a formula similar to (3.4), and with the implicit Euler method. We compute the L^2 numerical errors using a reference solution obtained by a standard method of order 10 with a very small time step. Unsurprisingly, the three methods are of order 1 numerically. Moreover, the linearly implicit method is faster than the implicit Euler method for a given error, but slower than the Lie splitting method. Note that all

the methods preserve the nonnegativity of the solution (as long as one has a bound on $\|u_n\|_2$ and h is sufficiently small with respect to this bound, and under an additional CFL condition for the Lie splitting method).

We display in Figure 9 the plots of the initial datum and the final time solution obtained at $T = 1$ with the same linearly implicit method of order 1 (for $h = 1/(5 \times 2^{11})$) (left hand side) and the plot of the evolution of E_{rlx} (right hand side). This illustrates the results of Proposition 3.4: the numerical solution starting from a nonnegative initial datum stays nonnegative, and the discrete energy does not increase with time (see (3.16)).

4. Conclusion and perspectives

This paper introduces a new class of methods for the time integration of evolution problems set as systems of ODEs (or PDEs after space discretization). This class contains methods that are only linearly implicit, no matter the evolution equation. Moreover, the paper describes a specific way to design linearly implicit methods of any arbitrarily high order. Using suitable definitions of consistency and stability, we prove that such methods are actually of high order for ODEs, and the proof extends to finite systems of ODEs. We illustrate numerically that some of these methods are of the expected order for two examples of PDEs (nonlinear Schrödinger equation in 1d and 2d and a nonlinear heat equation in 1d), and discuss some of their qualitative properties. Our numerical results show that the linearly implicit methods introduced in this paper behave rather poorly in terms of efficiency for simple small systems of ODEs. In contrast, they illustrate numerically that a linearly implicit method of order 2 outperforms standard methods of order 2 from the litterature for a NLS equation on a domain where no spectral method can be applied.

Perspectives of this work include a rigorous analysis of these linearly implicit methods in PDE contexts (*i.e.* before discretization in space). In this direction, a recent result [7] proves that the relaxation method (2.5), which belongs to the class of methods presented here (see Rem. 2.1), is of order 2 when applied to the NLS equation. Another question is that of the possibility to design, in a systematic way, linearly implicit methods of high order with some qualitative properties adapted to the PDE problem (*e.g.* preservation of mass or energy for NLS equation, energy decrease for parabolic problems).

Acknowledgements. This work was partially supported by the Labex CEMPI (ANR-11-LABX-0007-01).

References

[1] G. Akrivis and M. Crouzeix, Linearly implicit methods for nonlinear parabolic equations. *Math. Comput.* **73** (2004) 613–635.
[2] G. Akrivis and C. Lubich, Fully implicit, linearly implicit and implicit-explicit backward difference formulae for quasi-linear parabolic equations. *Numer. Math.* **131** (2015) 713–735.
[3] G. Bader and P. Deuflhard, A semi-implicit mid-point rule for stiff systems of ordinary differential equations. *Numer. Math.* **41** (1983) 373–398.
[4] A. Berman and R.J. Plemmons, Nonnegative matrices in the mathematical sciences. *Classics in Applied Mathematics*. Society for Industrial Mathematics (1987).
[5] C. Besse, A relaxation scheme for the nonlinear Schrödinger equation. *SIAM J. Numer. Anal.* **42** (2004) 934–952.
[6] C. Besse, G. Dujardin and I. Lacroix-Violet, High order exponential integrators for nonlinear Schrödinger equations with application to rotating Bose–Einstein condensates. *SIAM J. Numer. Anal.* **55** (2017) 1387–1411.
[7] C. Besse, S. Descombes, G. Dujardin and I. Lacroix-Violet, Energy-preserving methods for nonlinear Schrödinger equations. *IMA J. Numer. Anal.* **41** (2020) 618–653.
[8] J. Butcher, Diagonally-implicit multi-stage integration methods. *Appl. Numer. Math.* **11** (1993) 347–363.
[9] J. Butcher, General linear methods. Selected Topics in Numerical Methods. *Comput. Math. App.* **31** (1996) 105–112.
[10] J. Butcher and G. Wanner, Runge–Kutta methods: some historical notes. Special Issue Celebrating the Centenary of Runge-Kutta Methods. *Appl. Numer. Math.* **22** (1996) 113–151.
[11] M. Calvo, J. de Frutos and J. Novo, Linearly implicit Runge–Kutta methods for advection-reaction-diffusion equations. *Appl. Numer. Math.* **37** (2001) 535–549.
[12] Q. Cheng and J. Shen, Multiple Scalar Auxiliary Variable (MSAV) approach and its application to the phase-field vesicle membrane model. *SIAM J. Sci. Comput.* **40** (2018) A3982–A4006.
[13] J. Crank and P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Proc. Cambridge Philos. Soc.* **43** (1947) 50–67.
[14] G. Dahlquist, A special stability problem for linear multistep methods. *BIT Numer. Math.* **3** (1963) 27–43.

[15] M. Delfour, M. Fortin and G. Payre, Finite-difference solutions of a nonlinear Schrödinger equation. *J. Comput. Phys.* **44** (1981) 277–288.
[16] G. Dujardin, Exponential Runge–Kutta methods for the Schrödinger equation. *Appl. Numer. Math.* **59** (2009) 1839–1857.
[17] R. Frank, J. Schneid and C.W. Ueberhuber, The concept of B-convergence. *SIAM J. Numer. Anal.* **18** (1981) 753–780.
[18] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, Vol. 14. *Springer Verlag Series in Comput. Math.* Springer Berlin Heidelberg (1996).
[19] E. Hairer, S. Norsett and G. Wanner, Solving Ordinary Differential Equations I: Nonstiff Problems, Vol. 8. Springer (1993).
[20] E. Hairer, C. Lubich and G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, 2nd edition. *Springer Series in Computational Mathematics*. Vol. 31. Springer, Berlin Heidelberg (2002).
[21] K. Hayakawa, On nonexistence of global solutions of some semilinear parabolic differential equations. *Proc. Jpn. Acad.* **49** (1973) 503–505.
[22] M. Hochbruck and A. Ostermann, Exponential Runge–Kutta methods for parabolic problems. Tenth Seminar on Numerical Solution of Differential and Differential-Algebraic Equations (NUMDIFF-10). *Appl. Numer. Math.* **53** (2005) 323–339.
[23] M. Hochbruck and A. Ostermann, Exponential integrators. *Acta Numer.* **19** (2010) 209–286.
[24] P. Kaps and G. Wanner, A study of Rosenbrock-type methods of high order. *Numer. Math.* **38** (1981) 279–298.
[25] B. Kovàcs and C. Lubich, Linearly implicit full discretization of surface evolution. *Numer. Math.* **140** (2018) 121–152.
[26] W. Kutta, Beitrag zur naherungsweisen integration totaler differentialgleichungen. *Z. Math. Phys.* **46** (1901) 435–453.
[27] C. Lubich, On splitting methods for Schrödinger–Poisson and cubic nonlinear Schrödinger equations. *Math. Comp.* **77** (2008) 2141–2153.
[28] C. Lubich and A. Ostermann, Linearly implicit time discretization of non-linear parabolic equations. *IMA J. Numer. Anal.* **15** (1995) 555–583.
[29] R.I. McLachlan, Families of high-order composition methods. *Numer. Algorithms* **31** (2002) 233–246.
[30] H.H. Rosenbrock, Some general implicit processes for the numerical solution of differential equations. *Comput. J.* **5** (1963) 329–330.
[31] C. Runge, Ueber die numerische auflösung von differentialgleichungen. *Math. Annal.* **46** (1895) 167–178.
[32] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd edition. Society for Industrial and Applied Mathematics (2003).
[33] J. Shen and J. Xu, Convergence and error analysis for the Scalar Auxiliary Variable (SAV) schemes to gradient flows. *SIAM J. Numer. Anal.* **56** (2018) 2895–2912.
[34] K. Strehmel and R. Weiner, B-convergence results for linearly implicit one step methods. *BIT Numer. Math.* **27** (1987) 264–281.
[35] K. Strehmel, R. Weiner and I. Dannehl, A study of B-convergence of linearly implicit Runge–Kutta methods. *Computing* **40** (1988) 241–253.
[36] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Phys. Lett. A* **146** (1990) 319–323.
[37] J. Wensch, K. Strehmel and R. Weiner, A class of linearly-implicit Runge–Kutta methods for multibody systems. Special Issue Celebrating the Centenary of Runge–Kutta Methods. *Appl. Numer. Math.* **22** (1996) 381–398.
[38] H. Yoshida, Construction of higher order symplectic integrators. *Phys. Lett. A* **150** (1990) 262–268.