RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

Efficient Solution of Parabolic Equations by Krylov Approximation Methods

E. Gallopoulos and Y. Saad

RIACS Technical Report 90.14
March 1990

EFFICIENT SOLUTION OF PARABOLIC EQUATIONS BY KRYLOV APPROXIMATION METHODS

E. Gallopoulos and Y. Saad

The Research Institute of Advanced Computer Science is operated by
Universities Space Research Association, The American City Building,
Suite 311, Columbia, MD 244, (301)730-2656

Work reported herein was supported by the NAS Systems Division of NASA and DARPA, via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA). Work was performed at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, CA 94035-1000.

Efficient Solution of Parabolic Equations by Krylov Approximation Methods

E. Gallopoulos* and Y. Saad†

Abstract
In this paper we take a new look at numerical techniques for solving parabolic equations by the method of lines. The main motivation for the proposed approach is the possibility of exploiting a high degree of parallelism in a simple manner. The basic idea of the method is to approximate the action of the evolution operator on a given state vector by means of a projection process onto a Krylov subspace. Thus, the resulting approximation consists of applying an evolution operator of very small dimension to a known vector which is, in turn, computed accurately by exploiting well-known rational approximations to the exponential. Because the rational approximation is only applied to a small matrix, the only operations required with the original large matrix are matrix-by-vector multiplications, and as a result the algorithm can easily be parallelized and vectorized. Some relevant approximation and stability issues are discussed. We present some numerical experiments with the method and compare its performance with a few explicit and implicit algorithms.

Key Words. Parabolic problems, method of lines, explicit methods, Krylov subspace, parallelism, matrix exponential, polynomial approximation, rational approximation, stability.

AMS(MOS) subject classifications. 65M20, 65F10, 65W05.

* Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801. Research supported by the National Science Foundation under Grant NSF CCR-8717942. Additional support provided by AT&T Grant No. AT&T-AFFL-67-Sameh, the Air Force Office of Scientific Research Grant No. AFOSR-90-0044, and the Department of Energy Grant No. DOE DE-FG02-85ER25001.

† University of Minnesota, Computer Science Department, 200 Union St. SE, Minneapolis, MN 55455. Work supported by the NAS Systems Division and/or DARPA via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA).

1 Introduction

In recent years there has been a resurgence of interest in explicit methods for solving parabolic partial differential equations, motivated mainly by the desire to exploit the parallel and vector processing capabilities of new supercomputer architectures. The main attraction of explicit methods is their simplicity, since the basic operations involved in them are matrix-by-vector multiplications, which, in general, are rather easy to parallelize and vectorize. On the other hand, the stringent constraint on the size of the time steps required to ensure stability reduces efficiency to such an extent that the use of implicit methods becomes almost mandatory when integrating on long time intervals. Implicit methods do not suffer from stability related restrictions but have another disadvantage: they require the solution of linear systems that are often large and sparse. For this reason implicit methods tend to be far more difficult to implement on parallel machines than their explicit counterparts, which only require matrix-by-vector multiplications. Thus, the trade-off between the two approaches seems to be a large number of matrix-by-vector multiplications on the one hand, versus linear systems to solve on the other. For two-dimensional and, more important, for three-dimensional problems, methods of an explicit type might be attractive if implemented with care.

We next observe that in spite of the above conventional wisdom, the distinction between the explicit and implicit approaches is not always clear. Consider the simple system of ordinary differential equations $y' = Ay + f$. We first point out that if our only desire is to use exclusively matrix-by-vector products as operations with the matrix A, for example, for the purpose of exploiting modern architectures, then certainly standard explicit methods do not constitute the only possibility. For example, one may use an implicit scheme and solve the linear systems approximately by some iterative method, such as the conjugate gradient method, with no preconditioning or with diagonal preconditioning. When the accuracy required for each linear system is very low, then the method will be akin to an explicit scheme, although one of rather unusual type since its coefficients will vary at every step. As the accuracy required for the approximations increases, the method will start moving towards the family of purely implicit methods. Therefore, if we were to call "explicit" any scheme that requires only matrix-vector products, then the borderline between the two approaches is not so well-defined.

We would like to take advantage of this observation to develop schemes that are intermediate between explicit and implicit. In this paper we will use the term *polynomial approximation method* for any scheme for which the only operations required with the matrix A are matrix-by-vector multiplications. The number of such operations may vary from one step to another and may be large.

To derive such intermediate methods we explore systematically the ways in which an approximation to the local behavior of the ordinary differential equations can be obtained by using polynomials in the operator A. Going back to the comparison sketched above, we note that the process involved in one single step of an implicit method is often simply an attempt to generate some approximation to the operation $\exp(\delta t\ A)v$ via a rational approximation to the evolution operator [49]. If a CG-like method is used to solve the linear systems arising in the implicit procedure, the result will be a polynomial scheme. Thus, there are two phases of approximation: the first is obtaining a rational or polynomial approximation to the exponential, and the second is solving the linear systems by some iterative method. We would like to reduce these two phases to only one by attempting to directly approximate $\exp(\delta t\ A)v$. The basic idea is to project the exponential of the large matrix A into a small Krylov subspace.

To make the discussion more specific and introduce some notation, we consider the following linear parabolic partial differential equation:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= -Lu(x,t) + r(x), \quad x \in \Omega, \qquad (1) \\ u(0,x) &= u_0, \quad x \in \Omega, \end{aligned}$$

$$u(t,x) = \sigma(x), \; x \in \partial\Omega, t \geq 0,$$

where $-L$ is a second order partial differential operator of the elliptic type, acting on functions defined on the open, bounded and connected set Ω. Using a method of lines (MOL) approach, equation (1) is first discretized with respect to space variables, resulting in the system of ordinary differential equations

$$\begin{aligned} \frac{dw(t)}{dt} &= -Aw(t) + r, \\ w(0) &= w_0. \end{aligned} \tag{2}$$

For the remainder of our discussion, we will assume A to be time-independent. In this situation the solution is explicitly given by

$$w(t) = A^{-1}r + e^{-tA}(w_0 - A^{-1}r) \tag{3}$$

If we let $\hat{w}(t) \equiv w(t) - A^{-1}r$, and accordingly, $\hat{w}_0 \equiv w_0 - A^{-1}r$, then (3) is equivalent to the following expression:

$$\hat{w}(t) = e^{-tA}\hat{w}_0.$$

Note that when $r = 0$, $w(t)$ is the same as $\hat{w}(t)$. An ideal one-step method would consist of a scheme of the form

$$\hat{w}(t+\delta) = e^{-\delta A}\hat{w}(t) \tag{4}$$

in which δ constitutes the time step.

The basic operation in the above formula is the computation of the exponential of a given matrix times a vector. If we were able to perform this basic operation with high accuracy, we would have what is sometimes called a nonlinear one-step method [24], because it involves a nonlinear operation with the matrix A. We should stress that there is no need to actually evaluate the matrix exponential $\exp(-\delta A)$, but only its product with a given vector. This brings to mind an analogous situation for linear systems in which it is preferable to solve $Ax = b$ than to compute A^{-1} and then multiply the solution by b.

We point out that we follow an approach common in the literature [2, 39], putting the emphasis on the semi-discrete problem (2). As a result, our discussion of stability is purely from an Ordinary Differential Equation point of view and is not concerned with the effect of space discretization errors and convergence. We establish conditions under which our methods, applied to the stiff system of ODEs (2), satisfy certain criteria of stability which, in turn, is an important step toward any investigations of convergence. (See also [4, 37].)

The Krylov subspace method presented here was introduced in [12] for general nonsymmetric matrices. However, similar ideas have been used previously in various ways in different applications for symmetric or skew-symmetric matrices. For example, we would like to mention the use of this basic idea in Park and Light [31] following the work by Nauts and Wyatt [27]. The idea of exploiting the Lanczos algorithm to evaluate terms of the exponential of Hamiltonian operators seems to have been first used in chemical physics by Nauts and Wyatt in the context of the Recursive-Residue-Generation method [26]. More recently, Friesner et al. [8] have demonstrated that these techniques can be extended to solving nonlinear stiff differential equations. The approach developed in this paper is related to the work of Nour-Omid [29], in which systems of ODEs are solved by first projecting into Krylov subspaces and then solving reduced tridiagonal systems of ODEs; the approach of Tal-Ezer and Kosloff [43]; and also the work of Tal-Ezer

[42] and Schaefer [38] on polynomial methods based on Chebyshev expansions. The idea of evaluating arbitrary functions of a Hermitian matrix with the use of the Lanczos algorithm has also been mentioned by van der Vorst [48]. The use of preconditioning for extending the stability interval of explicit methods, thus bringing them closer to fully implicit methods, has been discussed in [47, 34]. Although our method works from a subspace, it does not suffer from some of the aspects of partitioning methods (see, for example, [52]). Partitioning methods rely on explicitly separating and treating differently the stiff and nonstiff parts. However, it is usually impractical to confine stiffness to a subsystem [3]. The Krylov method, on the other hand, relies on the nice convergence property of Krylov approximations to essentially reach a similar goal in an implicit manner [36]. The outermost eigenvalues, including the largest ones, will be well approximated by the Krylov subspace, so that the Krylov approximation to the matrix exponential will be accurate in those eigenvalues, thus accommodating stiffness. We also note that there have been several recent efforts to design agorithms for the solution of time-dependent problems, some of which may be particularly suited to parallel processing; see [45, 17, 19, 20, 40] and [46] for a review.

The structure of our paper is as follows. In Section 2 we formulate the Krylov subspace approximation algorithm and prove some a priori error bounds. In Section 3 we present a method for the accurate approximation of the exponential of the Hessenberg matrix produced in the course of the Arnoldi or Lanczos algorithm. In Section 4 we consider problems with time-dependent forcing and introduce two approaches to handle the integration of the non-homogeneous term. We then proceed in Section 5 with a stability analysis of each approach in the context of the quadrature techniques used, leading to Theorem 5.1. In Section 6 we present numerical experiments for problems of varying difficulty, and finally, in Section 7, our concluding remarks.

2 Polynomial approximation and the use of Krylov subspaces

In this section we consider using polynomial approximation to (4), that is, we seek an approximation of the form:

$$e^{-A}v \approx p_{m-1}(A)v, \tag{5}$$

where p_{m-1} is a polynomial of degree $m-1$. There are several ways in which polynomial approximations can be found. The simplest technique is to attempt to minimize some norm of the error $e^{-z} - p_{m-1}(z)$ on a continuum in the complex plane that encloses the spectrum of A. For example, Chebyshev approximation can be used, but one disadvantage is that it requires some approximation to the spectrum of A. In this paper we consider only approaches that do not require any information on the spectrum of A. This will be considered in Section 2.1. A theoretical analysis will then follow in Section 2.2.

2.1 The Krylov subspace approximation

The approximation (5) to $e^{-A}v$ is to be taken from the Krylov subspace

$$K_m \equiv span\{v, Av, \ldots, A^{m-1}v\}.$$

In order to manipulate vectors in K_m, it is convenient to generate an orthonormal basis $V_m = [v_1, v_2, v_3, \ldots, v_m]$. We will take as initial vector $v_1 = v/\|v\|_2$ and generate the basis V_m with the well-known Arnoldi algorithm, described below.

Algorithm: Arnoldi

```
1. Initialize:
    Compute $v_1 := v/\|v\|_2$.
2. Iterate: Do $j = 1,2,...,m$
    1.  Compute $w := Av_j$
    2.  Do $i = 1,2,\ldots,j$
        (a)  Compute $h_{i,j} := (w, v_i)$
        (b)  Compute $w := w - h_{i,j}v_i$
    3.  Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
```

By construction the above algorithm produces an orthonormal basis $V_m = [v_1, v_2, \ldots, v_m]$, of the Krylov subspace K_m. If we denote the $m \times m$ upper Hessenberg matrix consisting of the coefficients h_{ij} computed from the algorithm by H_m, we have the relation

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T. \tag{6}$$

For the remainder of this discussion, for any given k, e_k will denote the k^{th} unit vector belonging to R^m. From the orthogonality of the columns of V_m we get that $H_m = V_m^T A V_m$. Therefore H_m represents the projection of the linear transformation A to the subspace K_m, with respect to the basis V_m.

Since V_m is orthonormal, the vector $x_{opt} = V_m V_m^T e^{-A} v$ is the projection of $e^{-A}v$ on K_m, that is, it is the closest approximation to $\exp(-A)v$ from K_m. Since for $\beta \equiv \|v\|_2$, we can write $v = \beta v_1$ and $v_1 = V_m e_1$, it follows that:

$$\begin{aligned} V_m V_m^T e^{-A} v &= \beta V_m V_m^T e^{-A} v_1, \\ &= \beta V_m V_m^T e^{-A} V_m e_1. \end{aligned}$$

We can thus write the optimal solution as $x_{opt} \equiv V_m y_{opt}$ where $y_{opt} \equiv \beta V_m^T e^{-A} V_m e_1$. Unfortunately, y_{opt} is not practically computable, since it still involves e^{-A}. We can approximate $V_m^T e^{-A} V_m$ by e^{-H_m}, leading to the approximation $y_{opt} \approx \beta e^{-H_m} e_1$ and

$$e^{-A} v \approx \beta V_m e^{-H_m} e_1. \tag{7}$$

From the practical point of view there remains the issue of efficiently computing the vector $e^{-H_m} e_1$ which we address in Section 3.

The approximation (7) is central to our method, and its effectiveness is discussed throughout the remainder of the paper. The next section is devoted to providing the theoretical justification.

We also note that when A is symmetric, Arnoldi's algorithm simplifies into the Lanczos process, which entails a three-term recurrence. This is a result of the fact that the matrix $H_m = V_m^T A V_m$ must be symmetric and therefore tridiagonal symmetric, and so all $h_{i,j} = 0$ for $i = 1,2,..,j-2$. However, the resulting vectors, which are in theory orthogonal to each other, tend to lose their orthogonality rapidly.

2.2 A priori error bounds and general theory

The next question that arises concerns the quality of the Krylov subspace approximation defined in Section 2.1. A first observation is that the above approximation is exact for $m = n$, because in this situation $v_{m+1} = 0$ and (6) becomes $AV_m = V_m H_m$, where V_m is an $n \times n$ orthogonal matrix. In fact, similarly to the conjugate gradient method and the Arnoldi process, the approximation will be exact for m whenever m is larger or equal to the degree of the minimal polynomial of v_1 with respect to A. As for these algorithms, we need to investigate what happens when m is much smaller than this degree.

In the sequel we will need to use the concept of the *logarithmic norm* of a matrix. Let B be a given matrix. The logarithmic norm $\mu(.)$ is defined by:

$$\mu(B) \equiv \lim_{h \to 0+} \frac{\|I + hB\| - 1}{h} .$$

Note that μ is associated with a particular norm. Unless it is otherwise specified, we assume that the reference norm is the usual 2-norm. Then the logarithmic norm $\mu(B)$ is equal to the maximum eigenvalue of the symmetric part of B, that is,

$$\mu(B) = \lambda_{\max}(\frac{B + B^T}{2}).$$

The function μ satisfies many norm-like properties, but it can also take negative values. We refer to [5, 4] for a description of its properties. It can be shown in particular that

$$\|e^{Bt}\| \leq e^{\mu(B)t}. \tag{8}$$

We assume throughout that A is a real matrix. We now state the main theorem of this section.

Theorem 2.1 *Let A be any matrix and let $\rho \equiv \|A\|_2$, $\beta = \|v\|_2$ and $\eta \equiv \mu(-A)$. Then the error of the approximation (7) is such that*

$$\|e^{-A}v - \beta V_m e^{-H_m} e_1\|_2 \leq 2\beta\rho^m \phi(\eta) \leq 2\beta \frac{\rho^m}{m!} \max(1, e^\eta) \tag{9}$$

where

$$\phi(\eta) \equiv \frac{1}{\eta^m}\left(e^\eta - \sum_{k=0}^{m-1} \frac{\eta^k}{k!}\right) .$$

The proof of the theorem is established in Appendix B.

To see what one can gain in using the logarithmic norm instead of a standard spectral norm, compare Theorem 2.1 with the bound proposed earlier in [12]:

$$\|e^{-A}v - \beta V_m e^{-H_m} e_1\|_2 \leq 2\beta \frac{\rho^m e^\rho}{m!}. \tag{10}$$

For the sake of illustration let

$$B = \begin{pmatrix} 0.5000 & -0.0938 & 0.0000 \\ -0.4063 & 0.5000 & -0.0938 \\ 0.0000 & -0.4063 & 0.5000 \end{pmatrix}$$

and let $A = I \otimes B + B \otimes I$, where $\otimes$ is the symbol for the Kronecker product, and I the identity matrix. Such an A arises from the discretization of $u_{xx} + u_{yy} + \xi(u_x + u_y)$, when $\xi = 10.0$. In that case $\mu(-A) = -0.2929$, whereas $\|A\|_2 = 1.7235$. When $m = 7$, the bound for the remainder[1] obtained from Theorem 2.1 is 0.009, whereas the estimate from (10) is 0.0502, and the actual remainder norm is $\|r_7(-A)\|_2 = 0.0066$. Hence the use of the logarithmic norm results in an overshoot factor of only 1.3, in comparison to 7.5 when using the spectral norm. In general, the advantage of using the logarithmic norm

[1] Note that for simplicity we are here concerned only with the remainder of the Taylor series for e^{-A}, and hence only with the $\|r_m(A)\|_2$ part of the bound.

follows from the inequality $||A|| \geq \mu(-A)$ which is among the properties of $\mu(.)$ (cf. [5]). One can construct examples however, for which the bounds from using $\mu(-A)$ are as loose as those obtained from using $||A||$ (cf. [4]). Also, asymptotically the rates of convergence as estimated by the bounds (10) and (9) are both of the form $\rho/(m!)^{1/m}$. The following corollary follows trivially from Theorem 2.1.

Corollary 2.1 *If the eigenvalues of the symmetric part of the matrix A are non-negative, then:*

$$||e^{-A}v - \beta V_m e^{-H_m} e_1||_2 \leq 2\beta \frac{\rho^m}{m!}.$$

Hence the bound of Corollary 2.1 holds for many important classes of matrices including positive definite matrices and normal matrices with eigenvalues in the positive half-plane.

We note that when A is normal, the bound of Corollary 2.1 can be derived without invoking logarithmic norms because we can write $A = Q\Lambda Q^H$ where Λ is the diagonal matrix of eigenvalues of A and Q is unitary. Then

$$||r_m(A)||_2 = || \sum_{k=m}^{\infty} \frac{1}{k!}(-A)^k||_2 = || \sum_{k=m}^{\infty} \frac{1}{k!}(-\Lambda)^k||_2.$$

From the assumption on A, $\Re e\ [\Lambda] > 0$. Applying component-wise a result of E. Landau [22], (cf. [33, p. 35, problem 151]), the remainder can be bounded by its first term:

$$||r_m(A)||_2 \leq \frac{||-\Lambda^m||_2}{m!} = \frac{||A^m||_2}{m!}.$$

The bound of Corollary 2.1 follows after using a similar treatment for H_m and combining the results.

When we know that A is symmetric positive definite, an even better bound can be obtained by applying the previous theory to $A - \zeta I$, where $\zeta \geq \lambda_{\min}(A)$ (the minimum eigenvalue of A). We refer to [11] for the proof.

Theorem 2.2 *Let A be a symmetric positive definite matrix and let $\rho = ||A||_2$ and $\beta = ||v||_2$. Then the error of the approximation (7) is such that*

$$||e^{-A}v - \beta V_m e^{-H_m} e_1||_2 \leq \beta \frac{\rho^m}{2^{m-1} m!}, \tag{11}$$

These theorems show convergence of the approximation (7). They can also serve as a guide to choosing the step size in a time-stepping procedure. Indeed, if we were to replace A by the scaled matrix τA, then the Krylov subspace will remain the same; that is, V_m will not change, and H_m will be scaled to τH_m. As a result, for arbitrary τ one can use the approximation

$$e^{-\tau A}v \approx \beta V_m e^{-\tau H_m} e_1, \tag{12}$$

and the bound (9) becomes

$$||e^{-\tau A}v - \beta V_m e^{-\tau H_m} e_1||_2 \leq 2\beta(\tau\rho)^m \phi(\mu(-\tau A)). \tag{13}$$

The consequence of (13) is that by reducing the step-size one can always make the scheme accurate enough, without changing the dimension m. We note that these bounds are most useful when m is much larger than what is usually used by standard explicit methods. Indeed, in our experiments, we have used large values of m to our advantage. We refer the reader to [35] for additional results on error bounds for this method.

3 Practical computation of $\exp(-H_m)e_1$

We now address the problem of evaluating $y = e^{-H}e_1$, where H is the Hessenberg or tridiagonal matrix produced by Arnoldi's method or the Lanczos method. We drop the subscript m for convenience. Although H is a small matrix, the cost of computing y can easily become non-negligible. For example, when H is tridiagonal symmetric, the simplest technique for computing y is based on the QR algorithm. However, this is rather expensive. We would thus like to use approximations which have high accuracy, possess desirable stability properties, and allow fast evaluation. The method we recommend is to use rational approximation to the exponential, evaluated by partial fraction expansion. This technique has been discussed in the context of implicit methods by the authors [12, 9], and we would like to take advantage of it in the present context. The (serial) complexity of the QR algorithm is $O(m^3)$ for Hessenberg matrices and $O(m^2)$ for tridiagonal matrices, compared with a cost of $O(m^2)$ for Hessenberg and $O(m)$ for tridiagonal matrices when the rational approximation method is used. In addition, parallelism can be exploited in this approach.

The rational approximation to the exponential has the form

$$e^{-z} \approx R_{\nu_1,\nu_2}(z) \equiv \frac{p_{\nu_1}(z)}{q_{\nu_2}(z)}, \tag{14}$$

where p_{ν_1} and q_{ν_2} are polynomials of degrees ν_1 and ν_2 respectively.

An approximation of this type, referred to as a Padé approximation, is determined by matching the Taylor series expansion of the left-hand-side and right-hand-side of (14) at the origin. Since Padé approximations are local, they are very accurate near the origin but may be inaccurate far away from it. Other schemes have been developed [49, 2, 16] to overcome this difficulty. For typical parabolic problems that involve a second order partial differential operator $-L$ that is self-adjoint elliptic, the eigenvalues of L are located in the interval $[0, +\infty)$, and it is therefore natural to seek the Chebyshev (uniform) rational approximation to the function e^{-z} which minimizes the maximum error on the interval $[0, +\infty)$. To unify the Padé and uniform approximation approaches, we restrict ourselves to "diagonal" approximations of the form (ν, ν), that is, in which the numerator has the same degree ν as the denominator. We note however that alternative strategies (e.g., $(\nu - 1, \nu)$) will frequently work better for Padé approximations, without altering the principle of the method. Note that the stability properties of the aforementioned rational approximations are discussed extensively in the literature [51, 6, 18].

A comparison between the Padé approximation and the Chebyshev rational approximation reveals the vast superiority of the latter in the context of the Krylov-based methods presented in this paper, at least for symmetric positive matrices H, and relatively large values of m. To see why this is so, we note that the idea of the method presented in this paper is to allow the use of large time steps by utilizing Krylov subspaces of relatively high order. However, for our method to be successful, the ability to use a large time step δ must also carry over to the computation of $\exp(-H\delta)e_1$. We mentioned earlier that the Padé approximations provide good approximation only near the origin. Using the Chebyshev rational approximation to the function e^{-z} over the interval $[0, +\infty)$ [1, 49], it becomes possible to utilize time steps as large as our Krylov-based method allows.

For example in the diagonal Chebyshev rational approximation, the infinity norm of the error over the interval $[0, +\infty)$ is of the order of 10^{-10} as soon as ν reaches 10. For each additional degree the improvement is of the order of 9.289025... [1]. What this means is that for all practical purposes e^{-z} can be replaced by a rational function of relatively small degree. When H is nonsymmetric and its eigenvalues are complex, then the rational function is no longer guaranteed to be an accurate approximation to the exponential. Although a rigorous analysis is lacking, we experimentally verified that for the examples we treated, the approximation still remained remarkably accurate when the eigenvalues were near the positive real axis. Although little is known concerning rational uniform approximation in general regions of the complex plane, a promising alternative is to use asymptotically optimal methods based on Faber transformations in the complex plane [7]. We also point out that there exist other techniques for approximating matrix exponentials by rational functions of A; see, for example, [28, 16]. The restricted Padé approximations of [28] avoid complex arithmetic at the price of a reduced order of approximation, and reduced levels of parallelism caused by the occurrence of multiple poles.

For compactness of notation in the diagonal approximations we will write simply R_ν from now on for the (ν, ν) rational approximation to e^{-z}. Then, in order to evaluate the corresponding approximation to $e^{-H}e_1$, we need to evaluate the vector $\tilde{y}$, where

$$\tilde{y} = p_\nu(H)q_\nu(H)^{-1}e_1 = q_\nu(H)^{-1}p_\nu(H)e_1. \tag{15}$$

It has been proposed in several contexts that an efficient method for computing some rational matrix functions is to resort to their partial fraction expansions [9, 12, 10, 21, 23, 30, 41]. The approach is possible since it can be proved analytically that the diagonal Padé approximation to e^{-z} has distinct poles [53]. Explicit calculations indicate that this seems to be also true for the uniform approximation. In particular we write

$$R_\nu(z) \;=\; \alpha_0 + \sum_{i=1}^{\nu} \frac{\alpha_i}{z - \lambda_i}$$

where

$$\alpha_0 \equiv \frac{\pi_\nu}{\kappa_\nu}, \quad \text{and} \quad \alpha_i \equiv \frac{p_\nu(\lambda_i)}{q'_\nu(\lambda_i)} \quad i = 1, 2, \ldots, \nu$$

in which π_ν, κ_ν are the leading coefficients of the polynomials p_ν and q_ν respectively.

With this expansion the algorithm for computing (15) becomes:

Algorithm:

```
1. For $i = 1, 2, \ldots, \nu$ solve $(H - \lambda_i I)y_i = e_1$.
2. Compute $\tilde{y} = \alpha_0 e_1 + \sum_{i=1}^{\nu} \alpha_i y_i$.
```

The motivation in [9] for using the above scheme was parallelism. The first step in the above algorithm is entirely parallel since the linear systems $(H - \lambda_i I)y_i = e_1$ can be solved independently from one another. The partial solutions are then combined in the second step. The matrices arising in [9] are large and sparse, unlike those of the present situation. However, parallel implementation of the above algorithm can be beneficial for small Hessenberg matrices as well. For example, in a parallel implementation of the Krylov scheme, the "Amdahl effect" may cause severe reduction in efficiency unless all stages of the computation were sufficiently parallelized.

We should also point out that even on a scalar machine, the above algorithm represents the best way of computing $\tilde{y}$. It requires fewer operations than a straightforward use of the expression (15). It is also far simpler to implement. The poles λ_i and partial fraction coefficients α_i of $R_\nu(z)$ are computed once and for all and coded in a subroutine or tabulated. These are shown in Appendix A for $\nu = 10$ and $\nu = 14$ for the case of Chebyshev rational approximation.

4 The case of a time-dependent forcing term

In the previous sections we made the restrictive assumption that the function r in the right-hand side is constant with respect to time. In this section we address the more general case where r is time dependent. In other words, we now consider the system of ODEs of the form

$$\frac{dw(t)}{dt} \quad = \quad -Aw(t) + r(t). \tag{16}$$

As is well known, the solution of this system is of the form

$$w(t) \quad = \quad e^{-tA}w_0 + \int_0^t e^{(s-t)A}r(s)ds.$$

Proceeding as in Section 1, we now express $w(t+\delta)$ as

$$\begin{aligned} w(t+\delta) &= e^{-\delta A}\left(w(t) + \int_t^{t+\delta} e^{-(t-s)A}r(s)ds\right) \\ &= e^{-\delta A}w(t) + \int_t^{t+\delta} e^{-(\delta+t-s)A}r(s)ds \\ &= e^{-\delta A}w(t) + \int_0^{\delta} e^{-(\delta-\tau)A}r(t+\tau)d\tau. \end{aligned} \tag{17}$$

In one way or another, the use of the above expression as the basis for a time-stepping procedure will require numerical integration. Note, however, that under the assumption that we can evaluate functions of the form $e^{-As}v$ accurately, we have transformed the initial problem into that of evaluating integrals. Simple though this statement may seem, it means that the concerns about stability disappear as soon as we consider that we are using accurate approximations to the exponential. The reason for this is that the variable w does not appear in the integrand. The issue of stability will be examined in detail in Section 5.

The next question we would like to address is how to evaluate the integral in (17); for this we consider two distinct approaches.

4.1 The first approach

To begin with, consider a general quadrature formula of the form,

$$\int_0^{\delta} e^{-(\delta-\tau)A}r(t+\tau)d\tau \approx \sum_{j=1}^{p} \mu_j e^{-(\delta-\tau_j)A}r(t+\tau_j) \tag{18}$$

where the τ_j's are the quadrature nodes in the interval $[0, \delta]$. One of the simplest rules is the trapezoidal rule on the whole interval $[0, \delta]$ which leads to

$$\begin{aligned} w(t+\delta) &= e^{-\delta A}w(t) + \frac{\delta}{2}e^{-\delta A}r(t) + \frac{\delta}{2}r(t+\delta) \\ &= e^{-\delta A}[w(t) + \frac{\delta}{2}r(t)] + \frac{\delta}{2}r(t+\delta). \end{aligned}$$

The above formula is attractive because it requires only one exponential evaluation. On the other hand it may be too inaccurate to be of any practical value since it means that we may have to reduce the step size δ drastically in order to get a good approximation to the integrals. The next alternative is to use a higher order formula, that is, a larger p in (18). For example we tried a Simpson formula instead of trapezoidal rule. The improvements are noticeable, but we have to pay the price of an additional exponential evaluation at the mid-point $t + \delta/2$.

The recommended alternative is based again on a judicious exploitation of Krylov subspaces. In the formula (18) we note that each term $e^{-(\delta-\tau_j)A}r(t+\tau_j)$ need not be evaluated exactly. Observe that in the ideal situation where $r(s)$ is constant, equal to r, in the interval $[t, t+\delta]$, then formula (12) shows that we can evaluate $e^{-(\delta-\tau)A}r$ for all τ from the Krylov subspace generated for $\tau = 0$ (for example), via

$$e^{-(\delta-\tau)A}r \approx \beta V_m e^{-(\delta-\tau)H_m} e_1,$$

where V_m and H_m correspond to the Krylov subspace $K_m(A, r)$. In the more general case where r varies in the interval $[t, t+\delta]$ we can use a projection formula of the form

$$e^{-(\delta-\tau)A}r(t+\tau) \approx V_m e^{-(\delta-\tau)H_m} V_m^T r(t+\tau). \tag{19}$$

The combination of the quadrature formula (18) and formula (19) has been tested and was found to be remarkably accurate. Our experiment in Section 6.4 shows an example of a rapidly varying forcing term $r(t)$, where the method can perform adequately with only one additional exponential evaluation (at the midpoint). We note, however, that for some highly oscillatory forcing terms, a (preferably adaptive) scheme involving additional exponential evaluations or a reduction of the time step δ may be needed.

4.2 The second approach

In the above approach we need to compute two Krylov subspaces: one associated with the current iterate $w(t)$ and the other associated with $r(t)$. We would like to show that we can reduce the computation to only one Krylov subspace. The resulting algorithm has different numerical properties from the one presented in Section 4.1.

The main idea is to use the identity:

$$e^{-\delta A} = I - A\int_0^\delta e^{-sA}ds = I - A\int_0^\delta e^{-(\delta-\tau)A}d\tau,$$

which is obtained readily by integration. This is then substituted in the first term of the right-hand side of the equation

$$w(t+\delta) = e^{-\delta A}w(t) + \int_0^\delta e^{-(\delta-\tau)A}r(t+\tau)d\tau \tag{20}$$

to obtain

$$w(t+\delta) = w(t) + \int_0^\delta e^{-(\delta-\tau)A}[r(t+\tau) - Aw(t)]d\tau. \tag{21}$$

Note the important fact that the term $e^{-\delta A}w(t)$, which was in the previously used formula (20), has been removed at the slight expense of modifying the function $r(t+\tau)$ in the interval $\tau \in (0, \delta)$. The modification consists of subtracting a vector that is *constant* in the interval of integration. In terms of computations, this modification requires one matrix-by-vector multiplication, certainly an inexpensive overhead, compared with that of applying the propagation operator to a vector. There is one fundamental difference between the scheme (20) used in the first approach and the scheme (21) of the second approach: the unknown function w now figures in the integrand. This may mean completely different numerical properties and, as is shown in Section 5, the loss of the unconditional stability.

5 Stability

In this section we investigate the linear stability of the Krylov time-stepping methods when used for the solution of the semi-discrete system (2). As noted in the introduction, this discussion will not take into account the interaction between space and time discretizations.

We first consider the stability properties of the approximate evolution operator in (7). If A is positive real[2] i.e., if its symmetric part $S = \frac{1}{2}(A + A^T)$ is positive definite, then, so is the matrix H_m [32]. Moreover the eigenvalues of A and H_m have positive real parts and the smallest eigenvalue $\lambda_{\min}(S)$ of the symmetric part of A is a lower bound for the eigenvalues of $S_m = \frac{1}{2}(H_m + H_m^T)/2$, because $S_m = V_m^T S V_m$. In terms of logarithmic norms, $\mu(-H_m) \leq \mu(-A) \leq 0$; see also Lemma B.3 in Appendix A. Since V_m has orthonormal columns, the approximate evolution operator satisfies

$$\|V_m e^{-H_m\delta}\|_2 \leq \|e^{-H_m\delta}\|_2.$$

As a result, we can state that in the case where $\exp(-H_m\delta)$ is evaluated exactly then

$$\begin{aligned} \|V_m e^{-H_m\delta}\|_2 &\leq \|e^{-H_m\delta}\|_2 \\ &\leq e^{\mu(-H_m\delta)} \leq e^{\mu(-A\delta)} \\ &\leq 1. \end{aligned}$$

If, on the other hand, $\exp(-H_m\delta)$ is not computed exactly, then stability will depend upon the method of evaluation used. In particular, let $\exp(-H_m\delta)$ be evaluated using a diagonal (ν, ν) Padé approximation R_ν. Diagonal Padé approximations are $A-$acceptable. From above, $\Re x^H H_m x \geq 0$ for any x and $\mu(-H_m) \leq 0$. We then obtain

$$\|V_m R_\nu(H_m\delta)\|_2 \leq \|R_\nu(H_m\delta)\|_2 \leq 1$$

from a result of von Neumann which states that, when the field of values of a matrix B is contained in $\mathcal{H}$, the nonnegative half of the complex plane, and if a rational function f maps $\mathcal{H}$ in the unit disk, then $\|f(B)\|_2 \leq 1$; see also [44] and [13, Theorem 4]. A similar conclusion holds for the subdiagonal $(\nu - 1, \nu)$ approximation. When a diagonal Chebyshev approximation is used then this no longer holds as these approximations may amplify small eigenvalues of H_m near zero. We can only state that for symmetric positive definite matrices and large enough values of ν, and $\delta\lambda_{\min}(H_m)$ bounded away from zero then $\|V_m R_\nu(H_m\delta)\|_2 \leq 1$.

For the remainder of this section it will be assumed that the exponential terms $\exp(-H_m\delta)$ are computed exactly.

5.1 Stability behavior of the first approach

Consider a general solution scheme for (16) of the form

$$w_{n+1} = e^{-A\delta} w_n + s_n \tag{22}$$

where s_n is some approximation to the integral (18). The above scheme is a one-step technique where δ is the time step. If we assume that the exponential term is exactly evaluated, then the above methods are referred to as the nonlinear multistep methods by Lee [24]. It was remarked in [25] that these methods are stable. More generally, let us assume that the error incurred in the evaluation of the term $e^{-A\delta}w_n$ in (22) is $e_{1,n}$, while the error in the evaluation of the integral term s_n is $e_{2,n}$. Then the recurrence (22) is replaced by

$$w_{n+1} = e^{-A\delta} w_n + e_{1,n} + s_n^* + e_{2,n} \tag{23}$$

[2] A matrix B is positive real if $x^T B x > 0$ for any real vector $x \neq 0$ [50].

in which s_n^* is the exact integral (18). In this situation, the total error at each step is of the form

$$e_{n+1} \equiv w(t_{n+1}) - w_{n+1} = e^{-A\delta} e_n + e_{1,n} + e_{2,n}. \tag{24}$$

This shows that if A has no eigenvalues in the negative half-plane of the complex domain, then the above procedure is stable. Note that this is independent of the procedure used to compute the approximation to the matrix exponential by vector product. If one uses the Krylov approximation to the exponential, then we essentially have an explicit procedure that is stable. Although this may seem like a contradiction, notice that we have made very special assumptions. The important point is that we are essentially considering accurate one-step methods. In the extreme case where r is constant, the solution can be evaluated *in just one step* at any point in time provided the exponential is accurately approximated.

5.2 Stability behavior of the second approach

As mentioned earlier the alternative approach described in Section 4.2 is attractive from the point of view of efficiency but may have poor numerical properties. We will outline in this section a stability analysis of this class of methods in an effort to determine how to select the quadrature formulas that are most likely to lead to robust procedures.

We consider the simple time stepping scheme derived by applying a quadrature formula to the equation (21)

$$w_{n+1} = w_n + \delta \sum_{i=1}^{k} \mu_i e^{-(\delta-\tau_i)A} [r(t+\tau_i) - Aw_n] \tag{25}$$

where $\tau_i, i = 1, ..., k$ are the quadrature nodes in the interval $[0, \delta]$ and μ_i their corresponding weights. Once more, we assume that the exponential term in (25) is exactly calculated. The above equation can be recast in the form:

$$w_{n+1} = \left(I - \delta \sum_{i=1}^{k} \mu_i e^{-(\delta-\tau_i)A} A \right) w_n + g_n$$

in which g_n is a term that does not contain the variable w_n. The stability of the above recurrence is easily studied by replacing the matrix A by a generic eigenvalue λ. This leads to the scalar recurrence:

$$w_{n+1} = \left(1 - \delta \sum_{i=1}^{k} \mu_i e^{-(\delta-\tau_i)\lambda} \lambda \right) w_n + g_n.$$

We need to determine under which conditions the modulus of the evolution operator

$$a(\lambda) = 1 - \delta \sum_{i=1}^{k} \mu_i e^{-(\delta-\tau_i)\lambda} \lambda$$

does not exceed one.

Before proceeding with the more complicated general analysis, we first consider in detail two basic quadrature formulas: the trapezoidal rule and the mid-point rule. For the trapezoidal rule we have

$$a(\lambda) = 1 - \frac{\delta}{2}[\lambda e^{-\lambda\delta} + \lambda].$$

We restrict ourselves to the case where λ is real and positive. We need to have

$$-1 \le a(\lambda) = 1 - \frac{\delta\lambda}{2}[e^{-\lambda\delta} + 1] \le 1$$

or, since the second inequality is trivially satisfied,

$$\delta\lambda[e^{-\lambda\delta} + 1] \leq 4. \tag{26}$$

Since $e^{-\lambda\delta} \leq 1$, a sufficient condition for the above inequality to hold is that

$$\delta\lambda \leq 2,$$

which is just as restrictive as an ordinary explicit method. Note that a *necessary condition* for (26) to be true is that $\delta\lambda \leq 4$.

For the mid-point rule, we have

$$a(\lambda) = 1 - \delta\lambda e^{-\lambda\delta/2}.$$

Considering again real and positive λ, we will seek conditions under which we have

$$-1 \leq a(\lambda) = 1 - \delta\lambda e^{-\lambda\delta/2} \leq 1. \tag{27}$$

The second inequality is always satisfied and from the first we get the condition

$$\lambda\delta e^{-\lambda\delta/2} \leq 2.$$

As is easily seen through differentiation, the maximum with respect to $\lambda\delta$ of the left-hand side is reached for $\lambda\delta = 2$, and its value is $2e^{-1}$ which is less than 2. Therefore, inequality (27) is *unconditionally satisfied*. This fundamental difference between the trapezoidal rule and the mid-point rule underscores the change of behavior in the second approach depending on the quadrature rule used. We will extend this analysis shortly.

The above development for the mid-point rule was restricted to λ being on the positive real line. Let us consider this case in more detail for λ complex. Setting $u = \lambda\delta = \alpha - i\beta$, we have

$$\begin{aligned} a(\lambda) &= 1 - ue^{-u/2} \\ &= 1 - (\alpha - i\beta)e^{-(\alpha - i\beta)/2} \\ &= 1 - \alpha e^{-\alpha/2}(\cos(\beta/2) + i\sin(\beta/2)) + i\beta e^{-\alpha/2}(\cos(\beta/2) + i\sin(\beta/2)) \\ &= 1 - e^{-\alpha/2}\left((\alpha c + \beta s) + i(\alpha s - \beta c)\right) \end{aligned}$$

where we have set $c = \cos(\beta/2)$ and $s = \sin(\beta/2)$. The modulus of $a(\lambda)$ is easily found to satisfy

$$\begin{aligned} |a(\lambda)|^2 &= 1 + e^{-\alpha}(\alpha^2 + \beta^2) - 2e^{-\alpha/2}(\alpha c + \beta s) \\ &= 1 + e^{-\alpha}\left(|u|^2 - 2e^{\alpha/2}(\alpha c + \beta s)\right). \end{aligned}$$

This leads to the region of stability, symmetric about the positive real axis, defined by

$$(\alpha^2 + \beta^2) \leq 2e^{\alpha/2}\left(\alpha\cos(\beta/2) + \beta\sin(\beta/2)\right). \tag{28}$$

Figure 1 shows shaded, the part of the complex domain $[0, 50] \times [-25, 25]$ which corresponds to values of u satisfying (28).

If we concentrate on the shaded region enveloping the positive real axis, we note that for large α, the limits of the curve bounding that section of the stability region are $\beta = \pm\pi$. This is because for $\pi < |\beta| \leq 2\pi$, the coefficient $\cos(\beta/2)$ becomes negative, making (28) impossible to satisfy *for large* α, (β being fixed). On the other hand, for fixed β such that $|\beta| \leq \pi$, we have $\cos(\beta/2) \geq 0$ and there will always

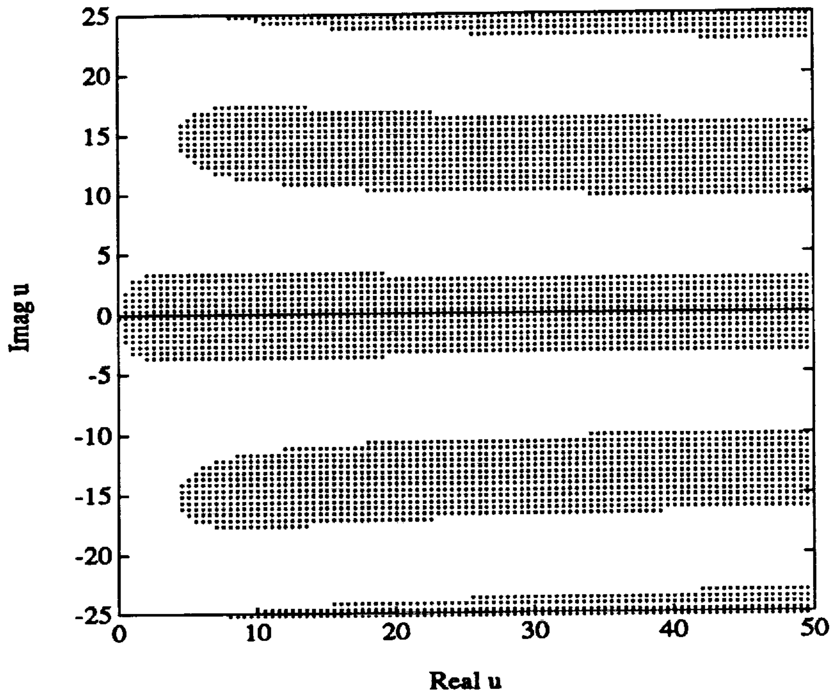

Figure 1: Stability region for the mid-point rule (cf. (28)).

be an α large enough that will satisfy (28). In addition, all the lines $\beta = \pm 4k\pi, k = 0, 1, 2, ..$ will belong partly to the region: specifically all those points in these lines with α larger than the (only) positive root of $x(2e^{x/2} - x) = \beta^2$ are acceptable points. As shown in Figure 1, around each of these lines there is a whole subregion of stability, whereas in between, there are regions around the lines $\beta = \pm 2(2k+1)\pi$ which are unstable.

We now go back to studying the general scheme and extend the above analysis to the general case. Again we restrict ourselves to the case where λ is real positive. This condition will be replaced by a weaker condition later. However we do not necessarily assume that the weights are positive. Then we examine the conditions under which

$$-1 \le a(\lambda) = 1 - \delta \sum_{i=1}^{k} \mu_i e^{-(\delta - \tau_i)\lambda} \lambda \le 1$$

or

$$0 \le \delta \lambda e^{-\delta\lambda} \sum_{i=1}^{k} \mu_i e^{\tau_i \lambda} \le 2. \tag{29}$$

Consider now any quadrature rule that satisfies the following two conditions:

1. Positivity condition:

$$\sum_{i=1}^{k} \mu_i e^{\lambda \tau_i} \ge 0, \text{ for } \lambda \ge 0. \tag{30}$$

2. Undervaluation condition:

$$E_k = \int_0^\delta e^{\lambda s} ds - \sum_{i=1}^{k} \mu_i e^{\tau_i \lambda} \geq 0, \text{ for } \lambda \geq 0. \tag{31}$$

The purpose of the positivity condition is to restrict the quadrature rule so that it yields nonnegative approximations to the integral of the function $e^{\lambda s}$ on any positive interval. It is verified whenever the quadrature weights are nonnegative. In particular

Lemma 5.1 *The positivity condition is satisfied for any Gaussian quadrature formula.*

Proof The fact that the weights are positive for Gaussian quadrature is well known; see, e.g. [15, p. 328]. □

For the undervaluation condition, we can prove the following lemma.

Lemma 5.2

1. *Any composite or simple open Newton-Cotes formula satisfies the undervaluation condition (31).*
2. *Any k-point Gaussian quadrature rule satisfies the undervaluation condition (31).*

Proof This is a consequence of the well-known error formulas for open Newton-Cotes rules [15, p. 313-314], and for Gaussian quadrature rules [15, p. 330], and the fact that all the derivatives of the function $e^{\lambda s}$ are positive in the interval $[0, \delta]$. □

Going back to the condition (29), we first observe under the positivity condition (30) the left-hand inequality is trivially satisfied. Moreover, under the undervaluation condition we have

$$\delta \sum_{i=1}^{k} \mu_i e^{\tau_i \lambda} \leq \int_0^\delta e^{\lambda s} ds = \frac{e^{\lambda\delta} - 1}{\lambda}$$

and as a result

$$0 \leq \delta\lambda e^{-\delta\lambda} \sum_{i=1}^{k} \mu_i e^{\tau_i \lambda} \leq \lambda e^{-\delta\lambda} \left(\frac{e^{\lambda\delta} - 1}{\lambda} \right) = 1 - e^{-\lambda\delta} \leq 2. \tag{32}$$

We have therefore proved the following result.

Theorem 5.1 *Consider the time-stepping procedure (25) based on the second approach (Section 4.2) using a quadrature formula satisfying the positivity condition (30) and the undervaluation condition (31). Then the region of stability of this scheme contains the positive real line.*

Note that schemes with such properties are said to be A_0-stable in the literature [39]. In many of our numerical experiments, we have observed this difference in stability behavior between schemes that satisfy the conditions of the theorem and those that don't. In many instances the composite closed-type Newton-Cotes formulas tended to diverge for a small number of subintervals. On the other hand we never noticed any stability difficulties with the open Newton-Cotes formulas or with the Gauss-Chebyshev quadrature.

As a general recommendation, it is advisable to use open Newton-Cotes formulas instead of closed formulas. Although these formulas satisfy the undervaluation condition according to the previous lemma, it is not known whether all of them satisfy the positivity condition (30). We do know that some of the low order, open Newton-Cotes rules (3-point, 4-point, and 6-point) do satisfy this condition since their weights are positive.

Gaussian rules are extremely attractive not only because of their stability properties but because of their potential to drastically reduce the number of function evaluations needed to produce a certain level of accuracy. There is still much work to be done to determine which of the quadrature formulas will yield the best results.

As is suggested by the analysis of the mid-point rule for complex λ, we expect the full analysis of the stability of the second approach to be very complicated for such cases.

6 Numerical experiments

6.1 A symmetric model problem

Our first test problem is issued from the semi-discretization of the heat equation

$$\begin{aligned} u_t &= u_{xx} + u_{yy} + u_{zz}, \; x, y, z \in (0,1) \\ u &= 0 \text{ on the boundary} \end{aligned}$$

using 17 grid points in each direction, yielding a matrix of size $N = 15^3 = 3375$. The initial conditions are chosen after space discretization, in such a way that the solution is known for all t. More precisely,

$$u(0, x_i, y_j, z_k) = \sum_{i',j',k'=1}^{n} \frac{1}{i'+j'+k'} \sin\frac{ii'\pi}{n+1} \sin\frac{jj'\pi}{n+1} \sin\frac{kk'\pi}{n+1}.$$

The above expression is simply an explicit linear combination of the eigenvectors of the discretized operator. In order to separate the influence of spatial discretization errors and emphasize the time evolution approximation, for the experiments in this section we consider the solution of the semi-discrete problem $u_t = -Au$ to be the exact solution.

The purpose of the first test is to illustrate one of the main motivations for this paper, namely the effectiveness of using large dimensional Krylov subspaces whenever possible. As shown in [12, 9], similar conclusions also hold for methods based on rational approximations to the exponential.

Assume that we want to integrate the above equation between t=0 and t=0.1, and achieve an error-norm at $t = 0.1$ which is less than $\epsilon = 10^{-10}$. Here by error-norm we mean the 2-norm of the absolute error.

We can vary both the degree m and the time-step δ. Normally we would prefer to first choose a degree m and then try to determine the maximum δ allowed to achieve the desirable error level. However, for convenience, we proceed in the opposite way: we first select a step-size δ and then determine the minimum m that is needed to achieve the desirable error level. This experiment was performed on a Cray Y-MP. What is shown in Table 1 are the various time steps chosen (column 1) and the minimum needed values of m (column 2) to achieve an error norm less than $\epsilon = 10^{-10}$ at t=0.1. We show in the third column the total number of matrix-by-vector multiplications required to complete the integration. The times required to complete the integration on a Cray Y-MP are shown in column 4. We also timed separately the evaluations of $e^{-\delta H_m} e_1$ and found these times to be negligible with respect to the rest of the computation. The last column of the table shows the type of rational approximation used when evaluating $e^{-\delta H_m} e_1$, with $C(\nu, \nu)$ representing the diagonal (ν, ν) approximation and $P(\nu, \nu)$ representing the diagonal (ν, ν) Padé approximation.

Another point is that the matrix is symmetric, so we have used a Lanczos algorithm to generate the v_i's instead of the full Arnoldi algorithm. No reorthogonalization of any sort was performed. The matrix consists of 7 diagonals, so the matrix by vector products are performed by diagonals resulting in a very

δ	m	M-vec's	Time (sec)	$\|Error\|_2$	Method
0.5000E-04	6	12006	0.8173E+01	0.1957E-11	P(2,2)
0.1000E-03	7	7007	0.4793E+01	0.3308E-10	P(2,2)
0.5000E-03	10	2010	0.1342E+01	0.1800E-10	P(4,4)
0.1000E-02	12	1200	0.7983E+00	0.2260E-10	P(4,4)
0.5000E-02	20	400	0.2672E+00	0.5271E-10	P(8,8)
0.1000E-01	26	260	0.1740E+00	0.7247E-10	P(8,8)
0.2000E-01	34	170	0.1080E+00	0.3236E-10	C(14,14)
0.3000E-01	39	156	0.9876E-01	0.6362E-10	C(14,14)
0.4000E-01	44	132	0.8030E-01	0.4122E-10	C(14,14)
0.5000E-01	49	98	0.5932E-01	0.5791E-10	C(14,14)
0.1000E+00	71	71	0.4186E-01	0.9993E-10	C(14,14)

Table 1: Performance of the polynomial scheme with varying accuracy on the Cray Y-MP.

effective use of the vector capabilities of the Cray architecture. Based on the time for the last entry of the table, we have estimated that the average Mflops rate reached, excluding the calculation of $e^{-\delta H_m} e_1$, was around 220. This is achieved with little code optimization.

Observe that the total number of matrix by vector products decreases rapidly as m increases. The ratio between the lowest degree $m = 6$ and the highest degree $m = 71$ is 169. The corresponding ratio between the two times is roughly 200. The case $m = 71$ can achieve the desired accuracy in just one step, that is, with $\delta = 0.1$. On the other hand for $m = 6$ a time-step of $\delta = 5 \times 10^{-5}$ must be taken resulting in a total of 2000 steps. We should point out that we are restricting ourselves to a constant time-step, but more efficient variable time stepping procedures are likely to reduce the total number of steps needed. From the result of Theorem 2.1 and Theorem 2.2 these observations come with no surprise. In effect, increasing the dimension of the Krylov subspace will increase the accuracy in such a way that a much larger ρ (i.e., a larger δ) can quickly be afforded.

6.2 A nonsymmetric problem with time-varying forcing term

In this section we consider the more difficult problem

$$\frac{\partial u(x,y,z,t)}{\partial t} = \Delta u(x,y,z,t) + \gamma \frac{\partial u(x,y,z,t)}{\partial x} + r(x,y,z,t), \tag{33}$$

where Δ stands for the three-dimensional Laplacian operator with homogeneous boundary conditions and initial conditions:

$$u(x,y,z,0) = x(x-1)y(y-1)z(z-1).$$

The function r is defined in such a way that the exact solution of the above partial differential equation is given by

$$u(x,y,z,t) = \frac{x(x-1)y(y-1)z(z-1)}{1+t}. \tag{34}$$

This yields

$$\begin{aligned} r(x,y,z,t) &= -\frac{x(x-1)y(y-1)z(z-1)}{(1+t)^2} + \gamma \frac{(2x-1)y(y-1)z(z-1)}{1+t} \\ &\quad -\frac{2[y(y-1)z(z-1) + x(x-1)z(z-1) + x(x-1)y(y-1)]}{1+t}. \end{aligned}$$

δ	m	Npts	Mvec's	Time (sec)	$\|Error\|_2$
0.2000E+00	40	60	205	0.2402E+01	0.6151E-05
0.1000E+00	40	40	410	0.3690E+01	0.7483E-06
0.1000E+00	40	30	410	0.3114E+01	0.3011E-05
0.1000E+00	35	40	360	0.3177E+01	0.7483E-06
0.1000E+00	30	40	310	0.2617E+01	0.7484E-06
0.1000E+00	25	40	260	0.2110E+01	0.8743E-06
0.1000E+00	25	30	260	0.1721E+01	0.1054E-04
0.5000E-01	25	30	520	0.3413E+01	0.7503E-07
0.5000E-01	25	20	520	0.2726E+01	0.3961E-06
0.5000E-01	20	20	420	0.2124E+01	0.6163E-05
0.5000E-01	15	20	320	0.1550E+01	0.5463E-04
0.2500E-01	20	10	840	0.2992E+01	0.9015E-06
0.2500E-01	15	20	640	0.3086E+01	0.5327E-05
0.2500E-01	15	10	640	0.2109E+01	0.9887E-05
0.1000E-01	10	10	1100	0.3561E+01	0.1743E-05
0.1000E-01	7	10	800	0.2693E+01	0.9483E-05

Table 2: Performance of the polynomial scheme with varying accuracy on the Cray-2.

As in the previous example, we took the same number of grid points in each direction, i.e., $n_x = n_y = n_z = 17$, yielding again a matrix of dimension $N = 15^3 = 3375$. This experiment was conducted on a Cray-2. Table 2 is the analogue of Table 1, except that we only report some representative runs with various values of m and δ. The parameter γ is set equal to 10.0. The integration is carried out from $t = 0.0$ to $t = 1.0$. The second approach was used in which the integrals were calculated with 11-points composite (closed) Newton-Cotes formulas. In most cases we had to take more than 11 points, in which case we simply used a composite rule with a total number of points equal to $1 + k \times 10$. The third column reports the total number of subintervals $Npts$ used to advance by one time step of δ. Thus, $Npts$ is a multiple of 10. The time shown in the fifth column is the time in seconds to advance the solution from $t = 0.0$ to $t = 1.0$, on a Cray-2. The sixth column shows the 2-norm norm of the error with respect to the exact solution of the *continuous* system, that is, with respect to (34).

We observe that for larger time steps a larger number of quadrature points must be used to keep a good level of accuracy. We show the results associated with the smallest number of points for which there are no significant qualitative improvements in the error when we increase $Npts$, while keeping m and δ constant. Our tests indicate that the higher the order of the quadrature used, the better. This means that large gains in speed are still likely if we use more optimal, Gaussian quadrature formulas. A noticeable difference with the previous simple example is that while large values of m tend to reduce the total number of matrix by vector multiplications required, the reduction is not as substantial.

6.3 A comparison with other methods

Although an exhaustive comparison with other schemes is beyond the scope of this paper, we would like to give an idea on how the efficiency of the Krylov subspace propagation compares with some immediate contenders. The first of these contenders is simply the forward Euler scheme. This is an explicit scheme, and for not-too-small space mesh sizes, should not be excluded given that the corresponding process is highly vectorizable. However, an approach that may be far more challenging is to use an implicit scheme such as the Crank-Nicolson method:

$$(I + \frac{\delta}{2}A)w_{n+1} = (I - \frac{\delta}{2}A)w_n + \delta r(t_n + \delta/2); \tag{35}$$

combined with an iterative method, for example the conjugate gradient method, for solving the linear systems. The main attraction here is that we can solve the linear systems inaccurately, making the solution process very inexpensive. From this viewpoint, this "inexact Crank-Nicolson" method shares many of the benefits of the Krylov method, as was already mentioned in the introduction. Finally, a well known stiff ODE package such as LSODE [14] is also considered.

For this comparison we took the same problem as before, but we needed to take $\gamma = 0.0$ in order to make the matrix A symmetric. This was necessary in order to be able to utilize the usual conjugate gradient algorithm for the linear systems in the Crank-Nicolson scheme. The r function is defined as before, and the number of grid points in each direction is again $n_x = n_y = n_z = 17$, yielding $N = 15^3 = 3375$.

We should point out that for the Crank-Nicolson method, we do not use preconditioning, and this is by no means a drawback. Because of time stepping, the matrix is usually very well conditioned, and as a result, the algorithm converges in a rather small number of steps. Moreover, because there is no need to solve the systems with high accuracy, the overhead in setting up the preconditioner would be difficult to amortize. Finally, the good preconditioners such as the incomplete factorizations do not generally yield a high a performance on vector machines. In our tests, the CG algorithm is stopped as soon as the residual norm is reduced by a factor which does not exceed a tolerance ϵ. We always take the tolerance ϵ that yields the smallest (or close to the smallest) time for the Crank-Nicolson scheme to complete. In this test we used the Chebyshev rational approximation of order (6,6) throughout, for the computation of $e^{-\delta H_m}e_1$. A final point of detail is that symmetry has been taken advantage of, both in Crank-Nicolson, which is able to use the usual conjugate gradient method, and in the Krylov method, in which we replaced the Arnoldi algorithm with the Lanczos version.

For LSODE we used the method flag MF=24, which means that a stiff method is used, and the Jacobian is user-supplied in banded format. The Cray-optimized Linpack banded solver is called to solve the linear systems. Table 3 shows the results. For LSODE we used a relative tolerance of `rtol` $= 10^{-14}$ and an absolute tolerance of `atol` $= 0.0$.

This comparison reveals that the Krylov scheme is superior when one considers the number of matrix-by-vector products as the primary criterion. There are situations in which these may dominate the cost, in which case the execution time could be proportional to the number of matrix-vector products. When execution time is the primary criterion for comparison, then the Krylov scheme is still faster than Crank Nicolson but not by as large a margin. The Forward Euler scheme was unstable for the time step $dt = 0.001$ and $dt = 0.00075$. We also performed a set of tests with a larger version of problem corresponding to the grid sizes $n_x = n_y = n_z = 22$, leading to a problem of size $N = 8000$. The conclusion is essentially the same in that Crank-Nicolson and the Krylov method are comparable, but the time for the explicit Euler scheme becomes much higher. We should add that we have regarded the problem purely from the angle of systems of ODEs, although we are aware that in practice a balanced accuracy between space and discretization is generally sought. However, this would lead to comparisons that are too complex.

6.4 A case with highly oscillating forcing term

We consider here an example of the same form as in the previous subsection; that is, the general equation is of the form (33), and the initial and boundary conditions are identical. However, we now consider a forcing term for which the exact solution is given by

$$u(x, y, z, t) = x(x-1)y(y-1)z(z-1)\cos(\alpha\pi t). \tag{36}$$

Method used	Method parameters	Matrix-vec. products	Total Cray-2 time (sec.)	Final error
Krylov	$m = 30, npts = 40$	155	0.9374E+00	0.6670E-05
$\delta = 0.2$	$m = 40, npts = 40$	205	0.1225E+01	0.6652E-05
	$m = 35, npts = 40$	180	0.1038E+01	0.6672E-05
	$m = 30, npts = 40$	155	0.9355E+00	0.6670E-05
	$m = 20, npts = 40$	105	0.6615E+00	0.7103E-05
	$m = 20, npts = 30$	105	0.5229E+00	0.1764E-04
Krylov	$m = 25, npts = 30$	182	0.9530E+00	0.9367E-06
$\delta = 0.15$	$m = 20, npts = 30$	147	0.7828E+00	0.7185E-05
	$m = 15, npts = 30$	112	0.6151E+00	0.4244E-04
Krylov	$m = 20, npts = 30$	210	0.1044E+01	0.7956E-06
$\delta = 0.1$	$m = 15, npts = 30$	160	0.9086E+00	0.8574E-05
Crank-	$dt = .01, \epsilon = .001$	1053	0.1192E+01	0.1267E-05
Nicolson	$dt = .005, \epsilon = .001$	1578	0.1767E+01	0.3329E-06
F-Euler	$dt = .0005$	2000	0.2779E+01	0.8678E-06
LSODE	MF=24	1077	0.3766E+02	0.4222E-04

Table 3: Performance comparison of a few methods on Problem of Section 6.3.

In other words, r is defined by

$$\begin{aligned} r(x,y,z,t) &= -\alpha x(x-1)y(y-1)z(z-1)\sin(\alpha\pi t) \\ &\quad -2[y(y-1)z(z-1)+x(x-1)z(z-1)+x(x-1)y(y-1) \\ &\quad +\gamma(2x-1)y(y-1)z(z-1)]\cos(\alpha\pi t). \end{aligned}$$

If the coefficient α is chosen to be large, then the problem can be difficult to solve. We took here $\gamma = 0.0$ and $\alpha = 20$. The discretization mesh is the same as in the previous example.

We compared the same four methods as those of the previous section, the Forward Euler scheme, the Crank-Nicolson/CG scheme, LSODE, and the Krylov method using the second approach. In this example LSODE failed to converge in a reasonable amount if time.

One difference with the previous tests is that here we varied the quadrature formulas used. Thus $npts = 4 \times 8$ indicates that we used a composite rule in which the interval of integration is first divided by 4 and then on each subinterval a nine-point formula is used. Apart from this, all of the details concerning implementation are identical with those of Section 6.3, except that this time we used the Chebyshev rational approximation of order (8,8) instead of (6,6) to compute the vectors $e^{-\delta H_m}e_1$.

The results in Table 4 indicate that for this harder problem, the Krylov scheme performs far better than its competitors. The Crank-Nicolson scheme now requires smaller time steps to achieve acceptable accuracies. The forward Euler scheme would require a much smaller time step that those of the other methods to achieve comparable performance.

7 Summary and Conclusion

The goal of this paper was to show how to systematically develop explicit type schemes, or to use our terminology, polynomial schemes for solving parabolic partial differential equations by the method of lines.

Method	Method	Matrix-vec.	Total Cray-2	Final
used	parameters	products	time (sec.)	error
Krylov	$m = 40, npts = 3 \times 10$	205	0.1202E+01	0.8051E-04
$\delta = 0.2$	$m = 30, npts = 2 \times 10$	155	0.6902E+00	0.2262E-03
	$m = 30, npts = 8 \times 5$	155	0.1196E+01	0.2862E-04
	$m = 25, npts = 20 \times 2$	130	0.1096E+01	0.3188E-04
	$m = 25, npts = 8 \times 5$	130	0.1063E+01	0.2320E-04
Krylov	$m = 20, npts = 2 \times 10$	210	0.1083E+01	0.7585E-05
$\delta = 0.1$	$m = 15, npts = 10 \times 2$	160	0.8995E+00	0.9713E-03
	$m = 15, npts = 2 \times 10$	160	0.8739E+00	0.6988E-04
	$m = 15, npts = 3 \times 8$	160	0.1043E+01	0.1592E-04
	$m = 15, npts = 4 \times 8$	160	0.1298E+01	0.1757E-05
	$m = 10, npts = 4 \times 8$	110	0.1066E+01	0.3504E-04
Crank-	$dt = .001, \epsilon = .001$	4322	0.5723E+01	0.8816E-04
Nicolson	$dt = .5E\text{-}03, \epsilon = .001$	8000	0.1058E+02	0.2203E-04
F-Euler	$dt = .5E\text{-}03$	2000	0.4780E+01	0.2358E-02
	$dt = .1E\text{-}03$	10000	0.2364E+02	0.4712E-03
	$dt = .5E\text{-}05$	20000	0.4861E+02	0.2356E-03
LSODE	MF=24	—	—	—

Table 4: Performance comparison of a few methods for Problem of Section 6.4.

We have proposed one such procedure that has the advantage of being very simple. The method proposed requires no information about the spectrum of the space discretization operator. We have recommended using high dimension Krylov subspaces whenever possible. By using a Krylov subspace of high dimension to approximate the evolution operator, we are able to use larger time-steps. At each step there is an additional cost due to the increased dimension of the Krylov subspace which translates into an increase in the number of matrix by vector multiplications. On the other hand, because of the larger time-step, the total number of steps required is reduced to such an extent that there is an appreciable net gain in performance. We have also proposed two approaches for handling non-constant forcing terms, with the view of extending these methods for general ODEs and nonlinear partial differential equations. The stability analysis of these approaches shows that the first is unconditionally stable and the second is A_0 stable for a large class of integration schemes used. This has been widely confirmed by numerical experiments which indicate that the schemes proposed are competitive with standard methods such as Crank-Nicolson.

Improvements to the approach described in Section 4.2 are possible by developing quadrature formulas that are more elaborate and specialized than the simple Newton-Cotes formulas used in our numerical experiments. We believe that the method proposed here can be extended to the solution of general time dependent nonlinear partial differential equations: the only subtlety is to isolate the action of the evolution operator, which is then well approximated by the schemes proposed here.

A Appendix: Partial fraction coefficients

In Table 5 we list some of the coefficients of the partial fraction expansion for the Chebyshev rational approximation to the exponential. These are the (k, k) approximations for $k = 10$ and $k = 14$. Note that because the roots go in complex conjugate pairs, we only need to show those with nonnegative imaginary

Degree	Coef/Root	Real Part	Imaginary Part
10	α_0	0.136112052334544905E-09	
	α_1	0.963676398167865499E+01	-0.421091944767815675E+02
	α_2	-0.142343302081794718E+02	0.176390663157379776E+02
	α_3	0.513116990967461106E+01	-0.243277141223876469E+01
	α_4	-0.545173960592769901E+00	0.284234540632477550E-01
	α_5	0.115698077160221179E-01	0.137170141788336280E-02
	λ_1	-0.402773246751880265E+01	0.119385606645509767E+01
	λ_2	-0.328375288323169911E+01	0.359438677235566217E+01
	λ_3	-0.171540601576881357E+01	0.603893492548519361E+01
	λ_4	0.894404701609481378E+00	0.858275689861307000E+01
	λ_5	0.516119127202031791E+01	0.113751562519165076E+02
14	α_0	0.183216998528140087E-11	
	α_1	0.557503973136501826E+02	-0.204295038779771857E+03
	α_2	-0.938666838877006739E+02	0.912874896775456363E+02
	α_3	0.469965415550370835E+02	-0.116167609985818103E+02
	α_4	-0.961424200626061065E+01	-0.264195613880262669E+01
	α_5	0.752722063978321642E+00	0.670367365566377770E+00
	α_6	-0.188781253158648576E-01	-0.343696176445802414E-01
	α_7	0.143086431411801849E-03	0.287221133228814096E-03
	λ_1	-0.562314417475317895E+01	0.119406921611247440E+01
	λ_2	-0.508934679728216110E+01	0.358882439228376881E+01
	λ_3	-0.399337136365302569E+01	0.600483209099604664E+01
	λ_4	-0.226978543095856366E+01	0.846173881758693369E+01
	λ_5	0.208756929753827868E+00	0.109912615662209418E+02
	λ_6	0.370327340957595652E+01	0.136563731924991884E+02
	λ_7	0.889777151877331107E+01	0.166309842834712071E+02

Table 5: Coefficients of the partial fraction expansion for degrees 10 and 14

parts. In fact there are exactly $\lceil k/2 \rceil$ such roots for the (k, k) approximation. Moreover in the case of a complex pair the corresponding coefficient α_i in the partial fraction expansion is doubled. The roots are also distinct, and we can thus write for a real x:

$$e^{-x} \approx \alpha_0 + Re\left[\sum_{i=1}^{\nu} \frac{\alpha_i}{x - \lambda_i}\right]. \tag{37}$$

B Appendix: Proof of Theorem 2.1

The following lemma provides the basis for establishing error bounds for the error of the approximation (7).

Lemma B.1 *Let A be any matrix, and p be any polynomial of degree smaller than m, approximating e^{-z} with the remainder $r_m(z) = e^{-z} - p(z)$. Then,*

$$\|e^{-A}v - \beta V_m e^{-H_m} e_1\|_2 \leq \beta(\|r_m(A)\|_2 + \|r_m(H_m)\|_2), \tag{38}$$

where $\beta = \|v\|_2$.

Proof As a result of the relation $e^{-z} = p(z) + r_m(z)$, we have

$$e^{-A}v = \beta[p(A)v_1 + r_m(A)v_1]. \tag{39}$$

Using induction and the relation (6) we can show that $A^j v_1 = V_m H_m^j e_1$, for $j \leq m-1$ and as a consequence we have

$$p(A)v_1 = V_m p(H_m)e_1. \tag{40}$$

As a result of the definition of p and r_m, we can write

$$p(H_m)e_1 = e^{-H_m}e_1 - r_m(H_m)e_1. \tag{41}$$

To complete the proof, we substitute (41) in (40) and the resulting equation in (39) to get,

$$\begin{aligned} e^{-A}v &= \beta V_m e^{-H_m}e_1 \\ &\quad + \beta[r_m(A)v_1 - V_m r_m(H_m)e_1]. \end{aligned}$$

The result follows immediately. □

Thus, the error can be estimated by bounding each of the two remainder terms. We now use the concept of the *logarithmic norm* of a matrix as defined in Section 2.2. We will specifically use the inequality $\|e^{Bt}\| \leq e^{\mu(B)t}$.

We next prove the following lemmas:

Lemma B.2 *Let*

$$s_{m-1}(z) = \sum_{k=0}^{m-1} \frac{(-z)^k}{k!}$$

be the $(m-1)$-th partial Taylor sum of e^{-z} and let $r_m(z)$ be the associated remainder $r_m(z) = e^{-z} - s_{m-1}(z)$. Define

$$\phi(\eta) \equiv \frac{1}{\eta^m}\left(e^\eta - \sum_{k=0}^{m-1}\frac{\eta^k}{k!}\right)$$

Then,

$$\|r_m(A)\| \;\leq\; \|A^m\|\phi(\eta) \;\leq\; \|A^m\|\frac{\max(1, e^\eta)}{m!}, \tag{42}$$

where $\eta \equiv \mu(-A)$.

Proof The remainder after m terms of the Taylor series expansion in integral form applied to $\exp(-A)$ is given by

$$r_m(A) = \frac{(-A)^m}{(m-1)!}\int_0^1 e^{-A(1-\tau)}\tau^{m-1}d\tau \tag{43}$$

and therefore,

$$\|r_m(A)\| \;\leq\; \frac{\|A^m\|}{(m-1)!}\int_0^1 \|e^{-A(1-\tau)}\|\tau^{m-1}d\tau.$$

Denoting $\eta = \mu(-A)$ for convenience, since $0 < \tau < 1$, we have from Eq. (8):

$$\|e^{-A(1-\tau)}\| \leq e^{\mu(-A)(1-\tau)} \equiv e^{\eta(1-\tau)}.$$

from which we get

$$\|r_m(A)\| \leq \frac{\|A^m\|}{(m-1)!} \int_0^1 e^{\eta(1-\tau)} \tau^{m-1} d\tau. \tag{44}$$

The value of the integral in the above expression is determined by noting that the remainder of the $(m-1)$-st Taylor expansion of e^η satisfies

$$e^\eta - \sum_{k=0}^{m-1} \frac{\eta^k}{k!} = \frac{\eta^m}{(m-1)!} \int_0^1 e^{\eta(1-\tau)} \tau^{m-1} d\tau$$

which gives

$$\phi(\eta) = \frac{1}{(m-1)!} \int_0^1 e^{\eta(1-\tau)} \tau^{m-1} d\tau.$$

Incidentally, this expression shows that $\phi(\eta)$ is nonnegative. Substituting this in (44) proves the first inequality in (42).

To prove the second part of the inequality, we observe that

$$\phi(\eta) = \frac{1}{(m-1)!} \int_0^1 e^{\eta(1-\tau)} \tau^{m-1} d\tau \leq \frac{1}{(m-1)!} \int_0^1 \max(1, e^\eta) \tau^{m-1} d\tau = \frac{\max(1, e^\eta)}{m!}. \tag{45}$$

□

We would like to mention that the upper bound for $\phi(\eta)$ used in the above lemma can be somewhat refined. More specifically, it can be shown that:

$$\phi(\eta) \leq \begin{cases} \frac{1}{m!} & \text{if } \eta < 0 \\ \frac{e^\eta}{m!} & \text{if } 0 < \eta \leq \sqrt[m-1]{(m-2)!m} \\ \frac{e^\eta}{(m-1)\eta^{m-1}} & \text{if } \sqrt[m-1]{(m-2)!m} \leq \eta \end{cases}$$

Finally, we will need the following lemma:

Lemma B.3 *If A is any real matrix and H_m is the associated $m \times m$ upper Hessenberg matrix generated by m steps of the Arnoldi algorithm, then:*

$$\mu(-H_m) \leq \mu(-A).$$

Proof By construction V_m consists of m orthonormal vectors and H_m satisfies $H_m = V_m^T A V_m$. Since the maximum eigenvalues of the symmetric parts of A and H_m can be characterized as the maximum values taken by their Rayleigh quotients, it easily follows that

$$\mu(-H_m) = \max_i \lambda_i(-\frac{V_m^T A V_m + V_m^T A^T V_m}{2}) \leq \max_i \lambda_i(-\frac{A + A^T}{2}) = \mu(-A).$$

□

Proof of Theorem 2.1.

First note that as in Lemma B.2 we can show that

$$\|r_m(H_m)\|_2 \leq \rho_m^m \phi(\mu(-H_m))$$

where $\rho_m = \|H_m\|_2 = \|V_m^T A V_m\|_2$. The right-hand side of the above inequality is an increasing function of ρ_m and $\rho_m \leq \rho$. From Lemma B.3, $\mu(-H_m) \leq \mu(-A) = \eta$, and thus:

$$\|r_m(H_m)\|_2 \leq \rho^m \phi(\eta). \tag{46}$$

Using Lemma B.1 the proof follows. □

Acknowledgments

We would like to thank Dr. Randall Bramley for his careful reading and suggestions, Professor Richard Varga for his comments on an early version of this work, Roland Freund for several helpful discussions, and the referees for their many helpful recommendations and for bringing to our attention the work of [7].

References

[1] A. J. CARPENTER, A. RUTTAN, AND R. S. VARGA, *Extended numerical computations on the 1/9 conjecture in rational approximation theory*, in Rational Approximation and Interpolation, P. R. Graves-Morris, E. B. Saff, and R. S. Varga, eds., vol. 1105 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1984, pp. 383–411.

[2] J. C. CAVENDISH, W. E. CULHAM, AND R. S. VARGA, *A comparison of Crank-Nicolson and Chebyshev rational methods for numerically solving linear parabolic equations*, J. Comput. Phys., 10 (1972), pp. 354–368.

[3] A. R. CURTIS, *Jacobian matrix properties and their impact on the choice of software for stiff ODE systems*, IMA J. Numer. Anal., 3 (1983), pp. 397–415.

[4] K. DEKKER AND J. G. VERWER, *Stability of Runge-Kutta methods for stiff nonlinear differential equations*, North-Holland, Amsterdam, 1984.

[5] C. DESOER AND H. HANEDA, *The measure of a matrix as a tool to analyze computer algorithms for circuit analysis*, IEEE Trans. Circuit Theory, 19 (1972), pp. 480–486.

[6] B. L. EHLE, *A-stable methods and Padé approximations to the exponential*, SIAM. J. Numer. Anal., 4 (Nov. 1973), pp. 671–680.

[7] S. W. ELLACOTT, *On the Faber transformation and efficient numerical rational approximation*, SIAM J. Numer. Anal., 20 (Oct. 1983), pp. 989–1000.

[8] R. A. FRIESNER, L. S. TUCKERMAN, B. C. DORNBLASER, AND T. V. RUSSO, *A method for exponential propagation of large systems of stiff nonlinear differential equations*, J. Sci. Comput., 4 (1989), pp. 327–354.

[9] E. GALLOPOULOS AND Y. SAAD, *Efficient parallel solution of parabolic equations: implicit methods on the Cedar multicluster*, in Proc. Fourth SIAM Conf. Parallel Processing for Scientific Computing, J. Dongarra, P. Messina, D. C. Sorensen, and R. G. Voigt, eds., SIAM, 1990, pp. 251–256. Chicago, Dec. 1989.

[10] ——, *Parallel block cyclic reduction algorithm for the fast solution of elliptic equations*, Parallel Comput., 10 (April 1989), pp. 143–160.

[11] ——, *Efficient solution of parabolic equations by polynomial approximation methods*, Tech. Rep. 969, Center for Supercomputing Research and Development, Feb. 1990.

[12] ——, *On the parallel solution of parabolic equations*, in Proc. 1989 ACM Int'l. Conference on Supercomputing, Herakleion, Greece, June 1989, pp. 17–28. Also CSRD Tech. Rep. 854.

[13] E. HAIRER, G. BADER, AND C. LUBICH, *On the stability of semi-implicit methods for ordinary differential equations*, BIT, 22 (1982), pp. 211–232.

[14] A. C. HINDMARSH, *ODEPACK, A systematized collection of ODE solvers*, in Scientific Computing, R. S. Stepleman, et al., ed., North Holland, Amsterdam, 1983, pp. 55–64.

[15] E. ISAACSON AND H. B. KELLER, *Analysis of Numerical Methods*, John Wiley & Sons, New York, 1966.

[16] A. ISERLES, *Rational interpolation to* $\exp(-x)$ *with application to certain stiff systems*, SIAM J. Numer. Anal., 18 (Feb. 1981), pp. 1–12.

[17] A. ISERLES AND S. P. NØRSETT, *On the theory of parallel Runge-Kutta methods*, IMA J. Numer. Anal., 10 (1990), pp. 463–488.

[18] A. ISERLES AND M. J. D. POWELL, *On the A-acceptability of rational approximations that interpolate the exponential function*, IMA J. Numer. Anal., 1 (1981), pp. 241–251.

[19] O. A. KARAKASHIAN AND W. RUST, *On the parallel implementation of implicit Runge-Kutta methods*, SIAM J. Sci. Stat. Comput., 9 (Nov. 1988), pp. 1085–1090.

[20] S. KEELING, *Galerkin/Runge-Kutta discretizations for parabolic equations with time-dependent coefficients*, Math. Comp., 52 (April 1989), pp. 561–586.

[21] H. T. KUNG, *New algorithms and lower bounds for the parallel evaluation of certain rational expressions and recurrences*, J. Assoc. Comput. Mach., 23 (April 1976), pp. 252–261.

[22] E. LANDAU, *Über einen Mellinshen Satz*, Arch. Math. Phys. Ser. 3, 24 (1915), pp. 97–107.

[23] J. D. LAWSON AND D. A. SWAYNE, *High-order near best uniform approximations to the solution of heat conduction problems*, in Proc. IFIP Congress 80 - Information Processing 80, New York, 1980, North Holland, pp. 741–746.

[24] D. LEE, *Nonlinear Multistep Methods for Solving Initial Value Problems in Ordinary Differential Equations*, PhD thesis, Polytechnic Institute of New York, 1974.

[25] D. LEE AND J. S. PAPADAKIS, *Numerical solutions of underwater acoustic wave propagation problems*, Tech. Rep. NUSC TR. 5929, Naval Underwater Systems Center, New London, CT, 1979.

[26] A. NAUTS AND R. E. WYATT, *New approach to many-state quantum dynamics: The recursive-residue-generation method*, Phys. Rev. Lett., 51 (1983), pp. 2238–2241.

[27] ——, *Theory of laser-module interaction: The recursive-residue-generation method*, Physical Rev., 30 (1984), pp. 872–883.

[28] S. P. NØRSETT, *Restricted Padé approximations to the exponential function*, SIAM J. Numer. Anal., 15 (Oct. 1978), pp. 1008–1029.

[29] B. NOUR-OMID, *Applications of the Lanczos algorithm*, Comput. Phys. Comm., 53 (1989), pp. 153–168.

[30] P. PANDEY, C. KENNEY, AND A. J. LAUB, *A parallel algorithm for the matrix sign function*, Int'l. J. High Speed Comput., 2 (June 1990), pp. 181–191.

[31] T. J. PARK AND J. C. LIGHT, *Unitary quantum time evolution by iterative Lanczos reduction*, J. Chem. Phys., 85 (1986), pp. 5870–5876.

[32] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, 1980.

[33] G. PÓLYA AND G. SZEGÖ, *Problems and Theorems in Analysis I*, Springer-Verlag, New York, 1972.

[34] G. Rodrigue and D. Wolitzer, *Preconditioned time-differencing for the parallel solution of the heat equation*, in Proc. Fourth SIAM Conf. Parallel Processing for Scientific Computing, J. Dongarra, P. Messina, D. C. Sorensen, and R. G. Voigt, eds., SIAM, 1990, pp. 268–272. Chicago, Dec. 1989.

[35] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, tech. rep., Research Institute for Advanced Computer Science, 1990.

[36] ——, *On the rates of convergence of the Lanczos and the block-Lanczos methods*, SIAM J. Numer. Anal., 17 (Oct. 1980), pp. 687–706.

[37] J. M. Sanz-Serna and J. G. Verwer, *Stability and convergence at the PDE/stiff ODE interface*, Appl. Numer. Math., 5 (1989), pp. 117–132.

[38] M. J. Schaefer, *A polynomial based iterative method for linear parabolic equations*, Tech. Rep. 661, Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, May 1987.

[39] W. L. Seward, G. Fairweather, and R. L. Johnston, *A survey of high-order methods for the numerical integration of semidiscrete parabolic problems*, IMA J. Numer. Anal., 4 (1984), pp. 375–425.

[40] Q. Sheng, *Solving linear partial differential equations by exponential splitting*, IMA J. Numer. Anal., 9 (1989), pp. 199–212.

[41] R. A. Sweet, *A parallel and vector cyclic reduction algorithm*, SIAM J. Sci. Statist. Comput., 9 (July 1988), pp. 761–765.

[42] H. Tal-Ezer, *Spectral methods in time for parabolic problems*, SIAM J. Numer. Anal., 26 (Feb. 1989), pp. 1–11.

[43] H. Tal-Ezer and R. Kosloff, *An accurate and efficient scheme for propagating the time dependent Schrödinger equation*, J. Chem. Phys., 81 (1984), pp. 3967–3971.

[44] J. v. Neumann, *Eine Spektraletheorie für allgemeine Operatoren eines unitären Raumes*, Math. Nachr., 4 (1950/51), pp. 258–281.

[45] P. J. van der Houwen and B. P. Sommeijer, *Parallel iteration of high-order Runge-Kutta methods with stepsize control*, J. Comput. Appl. Math., 29 (1990), pp. 111–127.

[46] ——, *Parallel ODE solvers*, in 1990 International Conference on Supercomputing, Amsterdam, June 1990, ACM, pp. 71–81.

[47] P. J. van der Houwen, B. P. Sommeijer, and F. W. Wubs, *Analysis of smoothing operators in the solution of partial differential equations by explicit difference schemes*, Appl. Numer. Math., 6 (1989/90), pp. 501–521.

[48] H. van der Vorst, *An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix A*, J. Comput. Appl. Math., 18 (1987), pp. 249–263.

[49] R. S. Varga, *On higher order stable implicit methods for solving parabolic partial differential equations*, J. Math. Phys., 40 (1961), pp. 220–231.

[50] E. L. Wachspress, *Iterative solution of elliptic systems*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.

[51] G. Wanner, *Order stars and stability*, in The State of the Art in Numerical Analysis, A. Iserles and M. J. D. Powell, eds., Clarendon Press, Oxford, 1987, pp. 451–472.

[52] D. S. Watkins and R. W. Hansonsmith, *The numerical solution of separably stiff systems by precise partitioning*, ACM Trans. Math. Softw., 9 (Sept. 1983), pp. 293–301.

[53] V. Zakian, *Properties of I_{MN} and J_{MN} approximants and applications to numerical inversion of Laplace transforms and initial value problems*, J. Math. Anal. Applic., 50 (1975), pp. 191–222.