

Accuracy Analysis of Time Integration Schemes for Stiff Multiscale Problems

Christophe Vandekerckhove
Dirk Roose

Report TW 457, April 2006



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Accuracy Analysis of Time Integration Schemes for Stiff Multiscale Problems

Christophe Vandekerckhove
Dirk Roose

Report TW 457, April 2006

Department of Computer Science, K.U.Leuven

Abstract

In the context of multiscale computations, techniques have recently been developed that enable microscopic simulators to perform macroscopic level tasks (equation-free multiscale computation). The main tool is the so-called coarse-grained time-stepper, which implements an approximation of the unavailable macroscopic time-stepper using only the microscopic simulator. Several schemes were developed to accelerate the coarse-grained time-stepper, exploiting the smoothness in time of the macroscopic dynamics. To date, mainly the stability of these methods was analysed. In this paper, we focus on their accuracy properties in the context of parabolic problems. We study the global error of the different methods, compare with explicit stiff ODE solvers, and use the theoretical results to develop more accurate variants. Our theoretical results are confirmed by various numerical experiments.

Keywords : Accuracy, Stability, Explicit, Parabolic, Equation-Free Multiscale Computing.

AMS(MOS) Classification : 35K45, 65M15, 65Y20.

Accuracy Analysis of Time Integration Schemes for Stiff Multiscale Problems

Christophe Vandekerckhove and Dirk Roose
Department of Computer Science, K.U.Leuven,
B-3001 Heverlee, Belgium

April 2006

Abstract

In the context of multiscale computations, techniques have recently been developed that enable microscopic simulators to perform macroscopic level tasks (equation-free multiscale computation). The main tool is the so-called coarse-grained time-stepper, which implements an approximation of the unavailable macroscopic time-stepper using only the microscopic simulator. Several schemes were developed to accelerate the coarse-grained time-stepper, exploiting the smoothness in time of the macroscopic dynamics. To date, mainly the stability of these methods was analysed. In this paper, we focus on their accuracy properties in the context of parabolic problems. We study the global error of the different methods, compare with explicit stiff ODE solvers, and use the theoretical results to develop more accurate variants. Our theoretical results are confirmed by various numerical experiments.

Keywords: Accuracy, Stability, Explicit, Parabolic, Equation-Free Multiscale Computing

AMS subject classification: 35K45, 65M15, 65Y20

1 Introduction

Many physical phenomena can be modeled at a microscopic level, using simple physical laws acting on large amounts of microscopic units (such as atoms, molecules or bacteria). However, a microscopic simulation code is often computationally very demanding. Moreover, in many cases we are only interested in the coarser, averaged, macroscopic behaviour over the largest length and time scales, rather than in all the microscopic details. Traditional modeling approaches therefore involve the derivation of a macroscopic evolution equation from the microscopic model, which is then studied using standard numerical tools. The macroscopic equation is formulated in terms of macroscopic variables, which often correspond to the lowest order moments of the microscopic variables or their distributions. In some cases, it might happen that although the macroscopic equation conceptually exist, it is not available in closed form. For such problems, Kevrekidis *et al.* developed the equation-free approach

which enables the modeler to perform macroscopic tasks, even when no macroscopic equation is available [8]. The main tool is the so-called coarse-grained time-stepper, which implements an approximation of the macroscopic time-stepper using only the microscopic simulator. This procedure relies on the same separation of time scales (between the macroscopic variables and the higher order moments) that also underpins the existence of the unknown macroscopic equation.

Computations with the coarse-grained time-stepper can be made more efficient in several ways. To compute the macroscopic time evolution of the system, the range of temporal and spatial scales (between the microscopic and the macroscopic variables but potentially also in the space of macroscopic variables itself) can be exploited to increase the efficiency [5, 8]. In this paper we consider methods that rely on the smoothness in time of the macroscopic dynamics to enable the efficient long-term time integration of multiscale problems through the coarse-grained time-stepper. In the related context of solving stiff ordinary differential equations (ODEs), many efficient techniques were developed since the 1950’s, e.g. implicit methods in combination with Newton’s method, or explicit methods with an extended stability region. Unlike these methods, the methods discussed in this paper only use the direct output of a time-stepper. Therefore, they can readily be applied to the coarse-grained time-stepper, for which equations are no longer available. The first method developed in this direction is the so-called Projective Method.

Projective Methods (PMs) were introduced in [5] for the efficient time integration of ODEs with a large gap between their fast and slow time-scales. The basic idea is remarkably simple: any stable explicit integrator (the “inner integrator”) is used to integrate the system forward in time over a number of small time steps (a “sequence of inner steps”), and then a polynomial extrapolation is used to compute an approximation to the solution far ahead. The small steps serve to damp the fast components in the solution and provide new data to the extrapolation routine. It was shown that the PM can be constructed such that it is absolutely stable if the eigenvalues fall in two clusters along the real axis, with a large gap in between. The remarkable result is that one is able to extrapolate (“project”) forward in time over step sizes that are adapted to the slow modes in the solution (as it should be!). A similar method was also developed in [2], but in that paper the small steps only serve to estimate the time derivative and not for damping of the fast modes. Therefore this method should only be used when there are only slow (and no fast) modes present in the system.

If there is no large time scale gap between the eigenvalues of the ODE, only a small speedup can be obtained with the PM. For problems with the eigenvalues in a narrow strip near the real axis (parabolic problems), Telescopic Methods (TMs) were developed [6]. These methods apply the projective integration idea recursively, and are related to the explicit Chebyshev methods for solving stiff ODEs [9, 12], or to the stabilization techniques given in [3].

Inspired by the PM, we developed a related method to accelerate time-steppers with a step size that is much smaller than the dominant slow time scales of the dynamics of the system [11]. The main difference with the PM is

that the extrapolation is not based on data from a single sequence of inner steps, but from several sequences. Hence, the scheme has a multistep character, and we will therefore call it the “Multistep State Extrapolation Method” (MSEM). It was shown in [11] that this scheme can also be used to accelerate parabolic time-steppers efficiently and in a stable manner.

In the context of the equation-free approach, it is expected that many of the fast, microscopic and possible undamped components of the microscopic simulator no longer present in coarse-grained time-stepper. This time-stepper can consequently be accelerated if the remaining eigenvalues match the stability conditions of one of the acceleration methods above. An interesting alternative view is that these methods estimate the right-hand-side (and possibly also higher order time derivatives) of the unknown macroscopic equation, which are then used to integrate forward in time. This is very similar to what a traditional time integration code would do if the exact right-hand-side of the macroscopic equation were available.

To date, most papers concerning the PM, TM and MSEM mainly deal with the stability of these methods. The goal of this paper is to analyse and to compare their accuracy properties in the context of parabolic problems, and to use this knowledge to construct two variants of the MSEM that are more accurate for parabolic problems. The outline of the paper is as follows. In section 2, we briefly review the PM, TM and the MSEM and their basic stability properties. In section 3, we derive expressions for the global error of the PM and MSEM and compare with the accuracy of the TMs and with the class of Chebyshev methods which were developed for ODEs. In the following two sections we study the accuracy and stability of the two variants of the MSEM. The first variant exploits the extra degrees of freedom that arise from using more states than strictly necessary to obtain an extrapolation up to a certain order (section 4), whereas the second variant combines the MSEM with the PM (section 5). Section 6 provides a numerical confirmation of our theoretical results and illustrates the performance of the different methods. Finally, in section 7 we summarize the main conclusions of this paper.

2 Preliminaries

In this section, we briefly review the MSEM, the PM and the TM, and their most important stability properties. We will consider discrete evolution equations of the form

$$y_{n+1} = \Phi_{\Delta t}(y_n), \quad y_n \in \mathbb{R}^M, \quad (1)$$

where $\Phi_{\Delta t}$ is a continuous and differentiable map (mostly called a time-stepper in this paper), y_n the M -dimensional state at time t_n and y_{n+1} the state at time $t_{n+1} = t_n + \Delta t$. We assume that the map (1) generates a sequence of points $\{y_i\}_{\Delta t}$ on a trajectory of a — not necessarily known — time-continuous evolution equation. Furthermore, we assume that the time step Δt is fixed and small compared to the dominant slow time scales of the dynamics of the system. Hence, iterating on (1) to compute the sequence $\{y_i\}_{\Delta t}$ is not very efficient, as the solution could also be described sufficiently accurate by another sequence

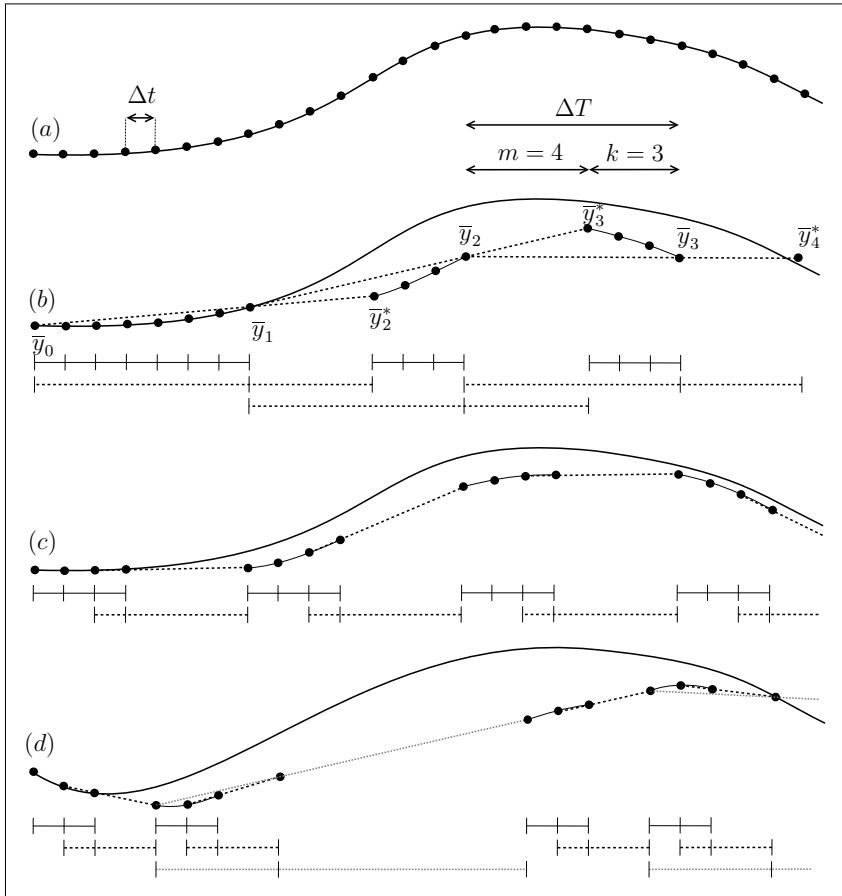


Figure 1: A schematic comparison of unaccelerated time-stepping (a), and accelerating the time-stepper using MSEM (b), PM (c) or TM (d). All methods are based on linear extrapolation. In subfigures (b) and (c) we used $k = 3$ and $m = 4$. In subfigure (d) we chose $k = m = 2$ and we added one extra level of recursion ($l = 2$).

$\{\bar{y}_j\}_{\Delta T}$, where the time step between the successive states is $\Delta T \gg \Delta t$. The MSEM, PM or TM are three different methods to compute such a sequence $\{\bar{y}_j\}_{\Delta T}$.

2.1 The Multistep State Extrapolation Method

Suppose that we have already computed $\{\bar{y}_0, \bar{y}_1, \dots, \bar{y}_n\}$ at times T_0, T_1, \dots, T_n . We can then compute the value \bar{y}_{n+1} at time $T_{n+1} = T_n + \Delta T$ as follows:

1. We use a component-wise polynomial extrapolation based on the interpolating polynomial P_N of degree N through the states $\bar{y}_{n-N}, \bar{y}_{n-N+1}, \dots, \bar{y}_n$ at times $T_{n-N}, T_{n-N+1}, \dots, T_n$, to obtain a new, intermediate point \bar{y}_{n+1}^* at time $T_n + m\Delta t = T_n + \mu\Delta T$.
2. Starting from \bar{y}_{n+1}^* , k additional time-stepper calls are performed using the map (1) to obtain \bar{y}_{n+1} .

This method can be seen as an extension of the method that was proposed in [10] in the context of increasing the real stability boundary of explicit methods. A schematic picture of this scheme is shown in Figure 1(b) for the case where $N = 1$, $m = 4$ and $k = 3$. The larger the part of the step ΔT that is bridged using the polynomial extrapolation, the larger the efficiency will be as long as the stability is maintained. Since the extrapolation is a very crude way to march forward in time, we lose some accuracy. One of the goals of this paper is to quantify this accuracy loss.

The MSEM can be written compactly as

$$\bar{y}_{j+1} = \Phi_{\Delta t}^k(\bar{y}_{j+1}^*) = \Phi_{\Delta t}^k \left(\sum_{s=0}^N l_s \cdot \bar{y}_{j-s} \right),$$

with

$$l_s = \frac{\mu \cdot (\mu + 1) \cdots (\mu + N)}{(s)!(N - s)!(-1)^s(\mu + s)}.$$

If we use a linear or quadratic extrapolation ($N = 1$ or $N = 2$), the scheme becomes

$$\bar{y}_{j+1} = \Phi_{\Delta t}^k((\mu + 1)\bar{y}_j - \mu\bar{y}_{j-1}) \quad (2)$$

or

$$\bar{y}_{j+1} = \Phi_{\Delta t}^k \left(\frac{(\mu + 1)(\mu + 2)}{2}\bar{y}_j - \mu(\mu + 2)\bar{y}_{j-1} + \frac{\mu(\mu + 1)}{2}\bar{y}_{j-2} \right). \quad (3)$$

If the overhead due to the extrapolation is negligible, we obtain a speedup

$$S \approx \frac{m + k}{k} = 1 + \frac{m}{k} = \frac{1}{1 - \mu}.$$

An extensive stability analysis of (2) and (3) was done in [11]. We studied the properties of the stability regions which are defined as the set of complex ρ -values for which the acceleration scheme, when applied to the scalar linear test integrator $y_{n+1} = \rho y_n$, tends to zero. In [6], an acceleration scheme is called $[0, 1]$ -stable if the line segment between zero and one is contained within the stability region. We showed that method (2) is $[0, 1]$ -stable for all values of S . In this paper, we will refer to such methods as being *overall* $[0, 1]$ -stable. A time-stepper with only real eigenvalues on $[0, 1)$ can unconditionally be accelerated by such a method in a stable way. Therefore, the extrapolation step size is only subject to accuracy considerations, as it should be. Unlike the scheme (2), the scheme (3) is not overall $[0, 1]$ -stable since it is only $[0, 1]$ -stable if $m < 5.2k$ (this corresponds to $\mu \in [0, 0.839]$) [11].

2.2 The Projective Method

The basic idea of the PM was briefly explained in the introduction, and is depicted schematically in Figure 1(c). The variable k (of which the value is now at least two) again denotes the number of time steps in each sequence of inner steps (this is slightly different from the notation in [5, 6]). The main difference

compared to the MSEM is that the polynomial to extrapolate forward in time is now based on points of only the last sequence of inner steps. Therefore the PM is expected to be more accurate than the MSEM if the underlying time-stepper is deterministic. In section 3 we will make this statement more precise. The gain in accuracy is however at the price of stability. It was shown in [5] that the stability regions of the PM with linear extrapolation are connected only if $m < 3.6(k-1)$ (this actually is an asymptotic result for $k \rightarrow \infty$, but it also holds quite well for small values of k). Therefore the PM is not overall $[0,1]$ -stable. As $m \rightarrow \infty$, the stability region approaches two disks, one centered at the origin with radius $(1/m)^{1/(k-1)}$, and another centered at $1 - 1/m$ with radius $1/m$. Similar results can be obtained for schemes based on higher order extrapolation methods. Note that the disk centered at the origin reflects the fact that the fast components in the solution are damped by the sequence of inner steps; such a disk is therefore not present in stability regions of the approach described in [2].

2.3 The Telescopic Method

The disadvantage of $[0,1]$ -stable PMs is that their speedup cannot be much larger than four. In [6], the PM is applied in a recursive manner to obtain $[0,1]$ -stable schemes with a larger efficiency. The projective integrator is then viewed as just another time-stepper, which can be used in a further projective integrator, and so on. These recursive methods are called TMs. The idea of the TM is shown schematically in Figure 1(d) for $k = m = 2$, and using one extra level of recursion. In this paper, we will denote the total number of projective levels by l ($l = 1$ corresponds to the PM). It was shown in [6] that by introducing these levels of recursion, one can obtain overall $[0,1]$ -stable methods. It was also shown that the TM with optimal efficiency is the one with $k = m = 2$. In that case, the size of the total telescopic step is $\Delta T = s^2$, with s the number of time-stepper calls in each such step. In this respect, these TMs are comparable to Chebyshev methods for stiff ODEs [12] (see also section 3.4).

3 Study of the Local and Global Error

In section 3.1 and 3.2 we study respectively the local and the global error of the PM and the MSEM. We show that although the local error of both schemes is quite comparable, the multistep character of the MSEM may cause the global error of the MSEM to be much larger than the global error of the PM. In section 3.3 and 3.4 we compare the accuracy of these methods to the accuracy of the TMs and of the Chebyshev methods. For simplicity, we shall restrict ourselves to the case where the time-stepper is the forward Euler method applied to the ODE $y'(t) = f(t, y)$, and we will use a linear extrapolation technique based on two points.

3.1 The Local Error of the PM and the MSEM

The local error $L_n := y(T_n) - \bar{y}_n$ at time step n is the truncation error made by stepping from time T_{n-1} to T_n , assuming that that no previous errors were made. Its dominant term can easily be found by assuming that $y''(t) = a$, with a constant. Under this assumption, the local error is independent of n and it is simply the sum of the local errors of the scheme's substeps. Since the PM can also be viewed as a scheme with $(k - 1)$ forward Euler steps of size Δt and 1 forward Euler step of size $(m + 1)\Delta t$, the local error is

$$L_{\text{PM}} = \frac{a}{2}((m + 1)\Delta t)^2 + (k - 1)\frac{a}{2}\Delta t^2 = \frac{a}{2}(m^2 + 2m + k)\Delta t^2. \quad (4)$$

For the MSEM, which consists of an extrapolation step of size $\mu\Delta T = m\Delta t$ and k forward Euler steps of size Δt , the local error is

$$L_{\text{MSEM}} = (\mu + \mu^2)\frac{a}{2}\Delta T^2 + k\frac{a}{2}\Delta t^2 = \frac{a}{2}(2m^2 + mk + k)\Delta t^2. \quad (5)$$

If $y''(t)$ is not constant over time, the dominant term of the local error will still be L_{PM} or L_{MSEM} , with a now replaced by $d^2y/dt^2 = df/dt = \partial f/\partial t + f\partial f/\partial y$. At this point, we would like to emphasize that the PM and the MSEM are intended to be used to accelerate time-steppers with a fixed time step. Therefore, unlike in traditional numerical analysis, we are mainly interested in how the error depends on m rather than on Δt . Expressions (4) and (5) indicate that for large speedups, the local error of the MSEM is approximately twice as large as the local error of the PM.

3.2 The Global Error of the PM and the MSEM

The global error $e_n := y(T_n) - \bar{y}_n$ at time step n is the total error made relative to the exact solution. It is affected by the local error at step n as well as by the propagation of previous errors. Below, we obtain an estimate of the global error for the PM and the MSEM through an asymptotic expansion [7]. We further restrict ourselves to the PM with two forward Euler steps between each two extrapolation steps, and to the MSEM with only one forward Euler step between each two extrapolation steps. These assumptions simplify the calculations below substantially, but they are by no means critical to the derivation, nor to its conclusions.

The PM can be written as

$$\bar{y}_{n+1} = \bar{y}_n + \Delta t f(T_n, \bar{y}_n) + (m + 1)\Delta t f(T_n + \Delta t, \bar{y}_n + \Delta t f(T_n, \bar{y}_n)),$$

while the exact solution y satisfies

$$y(T_n + (m + 2)\Delta t) = y(T_n) + \Delta t f(T_n, y(T_n)) + (m + 1)\Delta t f(T_n + \Delta t, y(T_n)) + \Delta t f(T_n, y(T_n)) + L_{\text{PM}} + \mathcal{O}(\Delta t^3).$$

Using the mean value theorem and assuming that $\frac{\partial f}{\partial y}$ is constant, we obtain

$$e_{n+1} = e_n + \Delta t \frac{\partial f}{\partial y} e_n + (m + 1)\Delta t \frac{\partial f}{\partial y} \left(e_n + \Delta t \frac{\partial f}{\partial y} e_n \right) + L_{\text{PM}} + \mathcal{O}(\Delta t^3).$$

We now define $\tilde{e}_{n+1} := e_n/\Delta t$ and introduce C_{PM} such that $L_{\text{PM}} = C_{\text{PM}}y''\Delta t^2$, to obtain

$$\tilde{e}_{n+1} = \tilde{e}_n + \Delta t \left(\frac{\partial f}{\partial y} \tilde{e}_n + (m+1) \frac{\partial f}{\partial y} \tilde{e}_n + C_{\text{PM}}y'' \right) + \mathcal{O}(\Delta t^2),$$

or

$$\tilde{e}_{n+1} = \tilde{e}_n + \Delta t(m+2) \left(\frac{\partial f}{\partial y} \tilde{e}_n + \frac{C_{\text{PM}}y''}{m+2} \right) + \mathcal{O}(\Delta t^2),$$

This formula can now be interpreted as the forward Euler scheme applied to

$$\tilde{e}'(t) = \frac{\partial f}{\partial y} \tilde{e}(t) + \frac{C_{\text{PM}}y''}{m+2} \quad (6)$$

Similarly, the MSEM can be written as

$$\bar{y}_{n+1} = (1+\mu)\bar{y}_n - \mu\bar{y}_{n-1} + \Delta t f(T_n + \mu(m+1)\Delta t, (1+\mu)\bar{y}_n - \mu\bar{y}_{n-1}),$$

and the exact solution satisfies

$$\begin{aligned} y(T_n + (m+1)\Delta t) &= (1+\mu)y(T_n) - \mu y(T_n - (m+1)\Delta t) \\ &\quad + \Delta t f(T_n + \mu(m+1)\Delta t, (1+\mu)y(T_n) - \mu y(T_n - (m+1)\Delta t)) \\ &\quad + L_{\text{MSEM}} + \mathcal{O}(\Delta t^3). \end{aligned}$$

By introducing C_{MSEM} such that $L_{\text{MSEM}} = C_{\text{MSEM}}y''\Delta t^2$, we obtain

$$\begin{aligned} \tilde{e}_{n+1} &= (1+\mu)\tilde{e}_n - \mu\tilde{e}_{n-1} \\ &\quad + \Delta t \left((1+\mu) \frac{\partial f}{\partial y} \tilde{e}_n - \mu \frac{\partial f}{\partial y} \tilde{e}_{n-1} + C_{\text{MSEM}}y'' \right) + \mathcal{O}(\Delta t^2), \end{aligned}$$

which can in turn be interpreted as the MSEM itself applied to

$$\tilde{e}'(t) = \frac{\partial f}{\partial y} \tilde{e}(t) + C_{\text{MSEM}}y''. \quad (7)$$

From (6) and (7) it follows that $E_{\text{PM}} := C_{\text{PM}}/(m+2)$ and $E_{\text{MSEM}} := C_{\text{MSEM}}$ are a natural measure for the global error of the PM and the MSEM. Note that these values correspond to the local error of the method, divided by the sum of the coefficients in front of $\Delta t f_i$ in the numerical scheme (assuming that the coefficient in front of \bar{y}_{n+1} is one).

Using (4) and (5), E_{PM} and E_{MSEM} simplify to

$$E_{\text{PM}} = \frac{m^2 + 2m + 2}{2(m+2)} \quad (8)$$

and

$$E_{\text{MSEM}} = \frac{2m^2 + m + 1}{2} \quad (9)$$

Solving (6) or (7) for a particular problem yields an accurate (Δt is very small!) explicit estimate for the global error. E.g., if y'' is constant (implying that $\frac{\partial f}{\partial y} = 0$), and if we set $e(0) = 0$, the global error estimate is

$$e(t) = \tilde{e}(t)\Delta t = E_i y'' t \Delta t.$$

Table 1: The global error at $t = 2$ for the accelerated simulation of a forward Euler time-stepper for $y' = -y$, $y(0) = 2$. The PM, MSEM and TM were used for various values of k and S . The parameter l indicates the speedup of the method: for the PM and the MSEM we used $m = 2^l$, and for the TM we used l levels of recursion. As l increases, the error increases by the factors shown between brackets.

Meth\l	2	3	4	5	6	7
PM (k=2)	1.17e-6	2.22e-6 (×1.89)	4.36e-6 (×1.96)	8.68e-6 (×1.99)	1.73e-5 (×2.00)	3.47e-5 (×2.00)
PM (k=3)	1.04e-6	2.04e-6 (×1.96)	4.15e-6 (×2.03)	8.44e-6 (×2.04)	1.71e-5 (×2.02)	3.44e-5 (×2.01)
MSEM (k=1)	1.00e-5	3.71e-5 (×3.70)	1.43e-4 (×3.86)	5.64e-4 (×3.94)	2.24e-3 (×3.97)	8.98e-3 (×4.01)
MSEM (k=2)	5.68e-6	1.98e-5 (×3.48)	7.39e-5 (×3.74)	2.86e-4 (×3.87)	1.13e-3 (×3.94)	4.49e-3 (×3.98)
TM (k=2)	2.30e-6	8.80e-6 (×3.82)	3.48e-5 (×3.95)	1.39e-4 (×3.99)	5.55e-4 (×4.00)	2.22e-3 (×4.01)
TM (k=3)	4.06e-6	2.35e-5 (×5.80)	1.41e-4 (×5.97)	8.43e-4 (×6.00)	5.10e-3 (×6.04)	3.21e-2 (×6.29)

Similarly, in the case of the linear test equation $y' = \lambda y$ with $y(0) = 1$, the global error estimate satisfies

$$e(t) = \tilde{e}(t)\Delta t = E_i \lambda^2 \exp(\lambda t) t \Delta t.$$

From equations (8) and (9), we see that as $m \rightarrow \infty$, the error increases linearly in m for the PM and quadratically in m for the MSEM (for constant Δt). The extra power of m in the case of the MSEM is due to the fact that also the slope of the line used for the extrapolation becomes less accurate as m increases (which is not the case for the PM). For other values of k , a similar derivation can be done, yielding $e = \mathcal{O}(m) = \mathcal{O}(kS)$ for the PM and $e = \mathcal{O}(m^2/k) = \mathcal{O}(kS^2)$ for the MSEM as $S \rightarrow \infty$. We can easily illustrate these relationships numerically by e.g. accelerating the forward Euler scheme with $\Delta t = 10^{-6}$ applied to $y' = -y$, $y(0) = 2$ on $t \in [0, 2]$. A cubic spline interpolation algorithm was used to determine the solution at $t = 2$, and for the MSEM the starting values were determined using the analytical solution. The global error of the different schemes are shown in Table 1. The expected order behavior is clearly visible as S becomes larger. Furthermore, the dependency of the error as a function of m for smaller values of S (i.e. the factors shown in the Table 1) can perfectly be predicted using equation (8), (9) or their analogues for other values of k .

In the next sections we compare the MSEM to the TM and to the class of Chebyshev methods, which are both — unlike the PM — also applicable to parabolic problems.

3.3 An Asymptotic Error Formula for the TM

The recursive character of the TM makes it difficult to derive precise accuracy estimates, and therefore we only give a hand waving argument which shows that in this case the global error also increases *at least* quadratically in S .

In [6] it was shown that the most efficient TM is the one with $k = m = 2$. For these parameter values, the number of time-stepper calls of one telescopic step is $W = 2^l$ and the total telescopic step size is $\Delta T = 4^l$, with l the number of recursion levels. Hence the speedup is $S = \Delta T/W = 2^l = W$ and $\Delta T = S^2$. The extrapolation step size at recursion level l is of size $\Delta T/2$ and therefore it is not unreasonable to expect that the global error will satisfy $e = \mathcal{O}(S^2)$ as $S \rightarrow \infty$. This is however no longer true for $k > 2$. We illustrate this for $k = 3$, for which it was shown that the largest value of m leading to an overall $[0, 1]$ -stable scheme is also 3 (see [6]). The number of time-stepper calls of one telescopic step is $W = 3^l$ and the total telescopic step size is $\Delta T = 6^l$. Hence we obtain $S = \Delta T/W = 2^l = W^{0.63}$ and $\Delta T = S^{2.58}$. The extrapolation step size at recursion level l is again of size $\Delta T/2$ and therefore we expect that $e = \mathcal{O}(S^{2.58})$ as $S \rightarrow \infty$.

Both asymptotic formulas for $k = 2$ and $k = 3$ are confirmed numerically in Table 1. Doubling S results in an error which is four times larger if $k = 2$, while the error is $2^{2.58} \approx 6.0$ times larger if $k = 3$.

3.4 An Asymptotic Error Formula for Chebyshev Methods

The TM bears strong resemblances with the class of Chebyshev methods for ODEs which address the issue of extending the stability region along the negative real axis [9, 12]. These methods are suitable for (mildly) stiff ODEs, typically coming from a PDE discretisation. One of the simplest first order methods is the so-called factorised method [1, 12]. Although this simple scheme is not recommended for practical use, as it suffers from severe internal instability [12], it does demonstrate the general efficiency properties of this class of methods. A numerical approximation y_{n+1} for the solution of $y'(t) = f(t, y)$ at time $T_{n+1} = T_n + \Delta T$ is computed with this method using the following s stages

$$\begin{aligned} X_0 &= y_n \\ X_j &= X_{j-1} + \tau_j f\left(t_n + \sum_{m=1}^{j-1} \tau_m, X_{j-1}\right), \quad 1 \leq j \leq s \\ y_{n+1} &= X_s. \end{aligned}$$

This scheme is stable if $|\prod_{j=1}^s (1 + \tau_j \lambda_i)| < 1$ for all eigenvalues λ_i of the Jacobian matrix $\partial f/\partial y$. Using the optimality properties of the Chebyshev polynomials of the first kind, it can be shown that for ODEs with only real eigenvalues, $\Delta T = \sum_{j=1}^s \tau_j$ is maximized if

$$\tau_j = \frac{2/\lambda_{\max}}{1 - \cos\left(\frac{2j-1}{s} \frac{\pi}{2}\right)},$$

with $\lambda_{\max} = \max |\lambda_i| \quad (\forall i)$ [1, 12]. In this case, $\Delta T = (2/\lambda_{\max})s^2$, and the speedup is $S = s$. This scheme is of first order in ΔT , hence we obtain $e = \mathcal{O}(s^2) = \mathcal{O}(S^2)$.

Summarizing, we can say that although the causing mechanisms are different, the global error of the MSEM, the optimal TM and the Chebyshev methods all behave as $\mathcal{O}(S^2)$ when $S \rightarrow \infty$. This seems to be the price to pay for accelerating problems with a range of eigenvalues along the real axis when using explicit, first order methods.

4 MSEM Variant 1: Using Extra Points

In the previous section, we studied the accuracy of the simplest instance of the MSEM, i.e. the scheme (2) with $k = 1$. The extrapolation was of first order since a linear extrapolation was used, based on two points. When using a polynomial of degree N through $N + 1$ points, the extrapolation will be of order N . As shown in [11], these methods are however not overall $[0,1]$ -stable, and hence no longer applicable to parabolic time-steppers, except for small speedups. In this section we investigate whether the extra degrees of freedom that arise from using extra points can be used to improve the method's accuracy, without losing the overall $[0,1]$ -stability property.

Section 4.1 deals with schemes based on first order accurate extrapolations, and in section 4.2 we study schemes based on second order accurate extrapolations. Without loss of generality, we choose $k = 1$ in this section (k time steps of size Δt can be interpreted as one integration step of another time integrator with step $k\Delta t$).

4.1 Schemes based on First Order Extrapolation

We initially focus on a variant of the MSEM which uses an extrapolation of first order and which is based on three points to compute \bar{y}_{j+1}^* . Such a scheme is of the form

$$\bar{y}_{j+1} = \Phi_{\Delta t}(\bar{y}_{j+1}^*) = \Phi_{\Delta t}(A(\mu)\bar{y}_j + B(\mu)\bar{y}_{j-1} + C(\mu)\bar{y}_{j-2}), \quad (10)$$

with

$$A(\mu) + B(\mu) + C(\mu) = 1, \quad B(\mu) + 2C(\mu) = -\mu. \quad (11)$$

The operator $\Phi_{\Delta t}$ now represents a first order accurate time integration step in which a local error $\theta/2\Delta t^2 y'' + \mathcal{O}(\Delta t^3)$ is made.

4.1.1 Accuracy and Stability Analysis

The local truncation error of the scheme (10) consists of a contribution from the extrapolation step (L_{extr}) and a contribution from the time-stepper call (L_{ts}). Assuming that no previous errors were made, we can expand both local errors through Taylor expansions as

$$L_{\text{extr}} = (1 - A - B - C)y + (\mu + B + 2C)y' \Delta T + \left(\frac{\mu^2}{2} - \frac{B}{2} - 2C \right) y'' \Delta T^2 + \mathcal{O}(\Delta T^3) \quad (12)$$

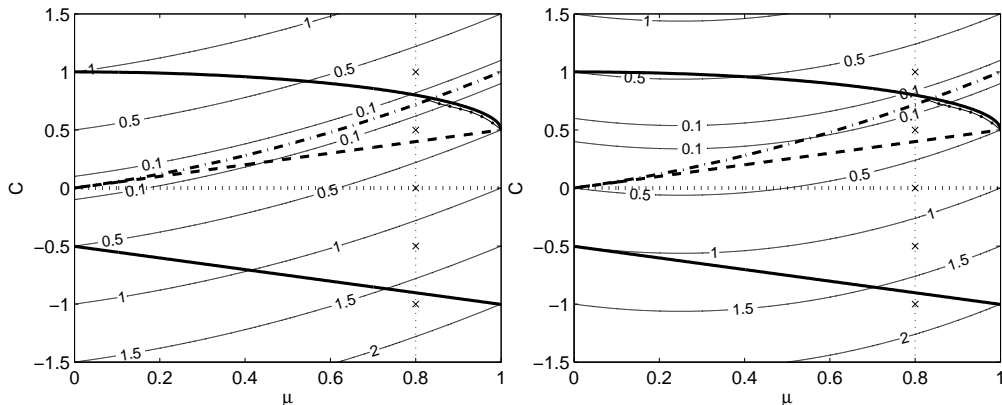


Figure 2: Diagrams showing the boundaries of $[0, 1]$ -stability (thick solid lines) and the contour lines of K if $\theta = 0$ (left) or $\theta = 1$ (right). The thick dashed, dotted and dash-dotted line correspond to different acceleration schemes based on linear extrapolation using three points (see text for details).

$$L_{ts} = \frac{1}{2}(\mu - 1)^2 \theta y'' \Delta T^2 + \mathcal{O}(\Delta T^3).$$

Taking (11) into account, we obtain

$$L = L_{\text{extr}} + L_{ts} = \underbrace{\left(\left(\frac{\mu^2}{2} + \frac{\mu}{2} - C \right) + \left(\frac{1}{2}(\mu - 1)^2 \theta \right) \right)}_K y'' \Delta T^2 + \mathcal{O}(\Delta T^3). \quad (13)$$

As indicated above, we will further denote the factor between the large brackets by K . Assuming that θ is constant, we thus obtain a family of acceleration methods with a single parameter C . We now determine the set of C and μ values for which the scheme is $[0, 1]$ -stable. The condition to be $[0, 1]$ -stable is that the three zeros ξ_i of the characteristic polynomial

$$\xi^3 = \rho(A\xi^2 + B\xi + C) = \rho((1 + \mu + C)\xi^2 - (\mu + 2C)\xi + C) \quad (14)$$

are smaller than one in magnitude for all time-stepper eigenvalues $\rho \in [0, 1)$. Although we could use the Routh-Hurwitz criterion (as $\rho \in \mathbb{R}$), we resort to a numerical approach, since this approach is easily extendable to schemes involving more than three points (for which the analytical formulations may become extremely difficult or even impossible). Our numerical code uses a discretisation of ρ on $[0, 1)$. For any value of μ , we compute the values of C that lie on the boundary of $[0, 1]$ -stable methods (i.e. the value of C for which at least one of the zeroes of (14) becomes equal to one in magnitude for any of the discrete values of ρ , while the magnitude of the other roots is smaller than one). The roots of (14) as well as the values of C at the $[0, 1]$ -stability boundary can easily be found using a root finding algorithm. The accuracy of the result can be made arbitrarily small by refining the discretisation of ρ on $[0, 1)$.

The result is shown in Figure 2. The thick solid lines represent the boundaries that separate $[0, 1]$ -stable schemes from $[0, 1]$ -unstable ones. Only the region

between both lines corresponds to $[0,1]$ -stable methods. In the left subfigure we added the contour lines of K with $\theta = 0$, i.e. using a second (or higher) order time-stepper. In the right subfigure we added the contour lines of K with $\theta = 1$ (e.g. when using the forward Euler scheme). As expected, the contour lines of K for both values of θ differ the most if $\mu \approx 0$, while they almost coincide if $\mu \approx 1$.

The thick dotted line corresponds to method (2). For this method, $\mu = 0$ implies that $L_{\text{extr}} = 0$, reflecting the fact that doing no extrapolation just produces the original trajectory of the time-stepper. Since this time-stepper only computes an approximation to the exact solution, $L \neq 0$ at $\mu = 0$. As $\mu \rightarrow 1$, the value of K corresponding to the method (2) increases towards 1, independent of θ . Figure 2 also confirms that this method is indeed overall $[0,1]$ -stable. The thick dash-dotted line, for which $L_{\text{extr}} \equiv 0$, corresponds to method (3). This method is clearly no longer $[0,1]$ -stable if $\mu > 0.839$, a result which was also analytically derived in [10, 11].

Figure 2 also shows that for large values of the speedup S , the highest accuracy is obtained if $C \approx 0.5$. An overall $[0,1]$ -stable scheme which is “optimal” for large speedups can now easily be constructed. One example is

$$\bar{y}_{j+1} = \Phi_{\Delta t}(\bar{y}_{j+1}^*) = \Phi_{\Delta t}\left(\left(1 + \frac{3}{2}\mu\right)\bar{y}_j - 2\mu\bar{y}_{j-1} + \frac{1}{2}\mu\bar{y}_{j-2}\right). \quad (15)$$

The thick dashed line in Figure 2 corresponds to this method. It is worth noting that this scheme has a nice mathematical interpretation. It corresponds to the MSEM variant that uses a linear extrapolation in time, based on a second order accurate time derivative estimate that was computed by combining the first order accurate time derivative estimates $\bar{y}'_1(T_j) \approx (\bar{y}_j - \bar{y}_{j-1})/\Delta T$ and $\bar{y}'_2(T_j) \approx (\bar{y}_j - \bar{y}_{j-2})/(2\Delta T)$. Many other schemes can be derived in a similar manner, but none of them will be more accurate near $\mu = 1$ without losing the $[0, 1]$ -stability property.

The scheme with “optimal” accuracy for *all* values of $\mu \in [0, 1)$ corresponds to the method (3) if $\mu < 0.83$, and to the scheme (10)–(11) with $C(\mu)$ chosen such that it stays slightly below the upper $[0,1]$ -stability boundary. The function $C(\mu)$ can easily be approximated by the polynomial

$$C(\mu) = -65.02\mu^3 + 172.75\mu^2 + -153.87\mu + 46.64. \quad (16)$$

The thin solid line with point markers in Figure 2 shows this relation $C(\mu)$ for $\mu > 0.83$.

Finally, Figure 3 shows the stability regions of five different methods based on a first order extrapolation using three points. The value of μ was kept fixed to 0.8, but different values of C were used (as indicated also by the crosses in Figure 2). According to Figure 2, only the methods with $C = -1$ and $C = 1$ are not $[0,1]$ -stable.

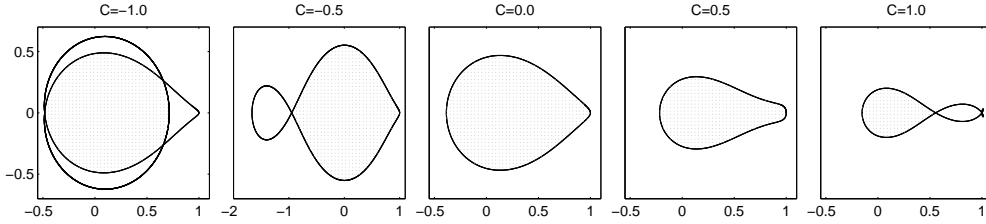


Figure 3: The stability regions of five different methods based on a first order extrapolation using three points. The value of μ is kept fixed to 0.8, but different values of C are used. At the solid line, at least one of the roots of the characteristic polynomial is one in modulus, while the shaded regions correspond to the stability regions. Note that the vertical axis is the same for all subfigures, while the horizontal axes differ.

4.1.2 Reducing the $\mathcal{O}(m^2)$ Behavior of the Global Error

Using the technique described in section 3.2, we can derive from (13) that the global error of the method (10)–(11) satisfies

$$e = \mathcal{O} \left(\left((1 - C)m^2 + \left(\frac{1}{2} - 2C\right)m - C + \frac{1}{2}\theta \right) \Delta t \right).$$

Again, we observe that the global error increases quadratically as $m \rightarrow \infty$. In the previous section we used the extra degree of freedom that arose from using one extra point to improve the method's accuracy without losing overall $[0,1]$ -stability. We can also use these degrees of freedom to reduce the $\mathcal{O}(m^2)$ behavior of the global error. Suppose for a moment that $C(\mu) = c_0 + c_1\mu + c_2\mu^2$. Then

$$e = \mathcal{O} \left(\left((1 - c_0 - c_1 - c_2)m^2 + \left(\frac{1}{2} - 2c_0 - c_1\right)m - c_0 + \frac{1}{2}\theta \right) \Delta t \right). \quad (17)$$

If $c_0 + c_1 + c_2 = 1$, then $e = \mathcal{O}(m\Delta t)$ as $\Delta t \rightarrow 0$ and $m \rightarrow \infty$. If additionally $c_0 = (1 - 2c_1)/4$, then $e = \mathcal{O}(\Delta t)$ under the same conditions. If also $c_0 = 0$, we obtain method (3) and the error is $\theta/2\Delta t + \mathcal{O}(\Delta t^2)$ (i.e. essentially of the same accuracy as the original time-stepper!). We can easily illustrate the above relations numerically, e.g. by accelerating the forward Euler scheme with time step $\Delta t = 10^{-6}$ applied to the problem

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} -500001 & 499999 \\ 500000 & -500000 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

using (10)–(11) with $C(\mu) = \mu/2$, $C(\mu) = (\mu^2 + \mu + 1)/3$ or $C(\mu) = (\mu^2 + \mu)/2$. The results are shown in Table 2, which clearly reflects the expected quadratic, linear and constant behavior of the global error as m grows. Moreover, the dependency of the error as a function of m for smaller values of m (the factors shown in the table) can be computed from equation (17). Only if m becomes too large, higher order terms in the expansion of the global error destroy the expected asymptotic behavior as Δt has a nonzero value.

Table 2: The global error as a function of m , for a problem with a gap in the eigenvalue spectrum (see text). We used three different acceleration methods which gives rise to a quadratic, linear or constant increase of the global error as m is increased.

m	e	factor	e	factor	e	factor
1	3.8e-07		1.3e-07		1.9e-07	
2	9.6e-07	2.5e+00	3.2e-07	2.5e+00	1.9e-07	1.0e+00
4	3.3e-06	3.4e+00	7.0e-07	2.2e+00	1.9e-07	1.0e+00
8	1.2e-05	3.8e+00	1.5e-06	2.1e+00	1.9e-07	1.0e+00
16	4.9e-05	4.0e+00	3.0e-06	2.0e+00	1.9e-07	9.9e-01
32	2.0e-04	4.0e+00	6.1e-06	2.0e+00	1.8e-07	9.4e-01

It should be clear however that *none* of the acceleration schemes above will be overall $[0, 1]$ -stable if the global error increases less than quadratically in m . Indeed, the condition $c_0 + c_1 + c_2 = 1$ implies that $C(1) = 1$, which corresponds to a method which is not $[0, 1]$ -stable (see Figure 2). We do not observe any instabilities in Table 2, since we used the particular time-stepper above, which has a large gap in its eigenvalue spectrum (the ODE eigenvalues are -1 and -10^6). Therefore, even if the acceleration method is no longer $[0, 1]$ -stable, instabilities do not occur because both eigenvalues will still fall within the (disjunct pieces of the) stability region.

We will now generalise the observations above for acceleration schemes that use an arbitrary number of points for the extrapolation. The first part of the proof (Theorem 1) is a generalization of the fact that in the example above, the order reduction implies that we obtain the scheme (3) for which the extrapolation is second order accurate at $\mu = 1$. In the next section we then prove that no overall $[0, 1]$ -stable schemes can be achieved if the extrapolation is of second order.

Theorem 1 Consider first order acceleration methods of the form

$$\bar{y}_{j+1} = \Phi_{\Delta t}(\bar{y}_{j+1}^*) = \Phi_{\Delta t}(A(\mu)\bar{y}_j + B(\mu)\bar{y}_{j-1} + C(\mu)\bar{y}_{j-2} + D(\mu)\bar{y}_{j-3} + \dots) \quad (18)$$

with $A(\mu)$, $B(\mu)$, $C(\mu)$, \dots polynomials in μ of degree at most d and $\Phi_{\Delta t}$ a first (or higher) order time-step in which a local error $\theta/2\Delta t^2 y'' + \mathcal{O}(\Delta t^3)$ is made. For such schemes, the dominant term of the global error grows less than quadratically in m (as $\Delta t \rightarrow 0$ and $m \rightarrow \infty$) if and only if the extrapolation step is of second order at $\mu = 1$.

Proof. The global error,

$$e = \mathcal{O} \left(\left(\frac{\mu^2/2 - \frac{1}{2}B(\mu) - 2C(\mu) - \frac{9}{2}D(\mu) - \dots}{(\mu - 1)^2} + \frac{\theta}{2} \right) \Delta t \right),$$

can be rewritten using $\mu = \frac{m}{m+1}$ as

$$e = \mathcal{O} \left(\frac{\left(\frac{1}{2} - \frac{B(1)}{2} - 2C(1) - \frac{9}{2}D(1) - \dots \right) m^{\max(2,d)} + \mathcal{O}(m^{\max(2,d)-1})}{(m+1)^{\max(2,d)-2}} \Delta t \right). \quad (19)$$

The theorem now directly follows from (19), since the leading coefficient in $m^{\max(2,d)}$ in the numerator is exactly the second order condition for the extrapolation scheme at $\mu = 1$ (see e.g. (12) for the case with only three points). ■

Combining Theorem 1 with Theorem 2 from the following section immediately proves that the global error of overall $[0,1]$ -stable methods of the form (18) using a first order extrapolation will always increase at least quadratically in m .

4.2 Schemes based on Second Order Extrapolation

In this section, we initially consider a variant of the MSEM which uses an extrapolation of second order and which is based on four points to compute \bar{y}_{j+1}^* . Such a scheme is of the form

$$\bar{y}_{j+1} = \Phi_{\Delta t}(\bar{y}_{j+1}^*) = \Phi_{\Delta t}(A(\mu)\bar{y}_j + B(\mu)\bar{y}_{j-1} + C(\mu)\bar{y}_{j-2} + D(\mu)\bar{y}_{j-3}). \quad (20)$$

with

$$A + B + C + D = 1, \quad B + 2C + 3D = -\mu, \quad B + 4C + 9D = \mu^2 \quad (21)$$

The operator $\Phi_{\Delta t}$ now represents a second order accurate time integration step in which a local error $\theta/6\Delta t^3 y''' + \mathcal{O}(\Delta t^4)$ is made. As in the previous subsection, one can derive that the local truncation error of this scheme is

$$L = L_{\text{extr}} + L_{\text{ts}} = \underbrace{\left(\left(\frac{\mu^3}{6} + \frac{\mu^2}{3} + \frac{\mu}{2} + D \right) + \left(\frac{1}{6}(\mu - 1)^3 \theta \right) \right)}_K y''' \Delta T^3 + \mathcal{O}(\Delta T^4).$$

We again denote the factor between brackets by K . If θ is constant, we obtain a family of acceleration methods with a single parameter D .

A diagram similar to Figure 2 is shown in Figure 4. In this case, only the region inside the thick solid line corresponds to $[0,1]$ -stable methods. Therefore, no methods of the form (20)–(21) can be found that are overall $[0,1]$ -stable. In the left subfigure we added the contour lines of the local error constant K with $\theta = 0$, i.e. using a third (or higher) order time-stepper. In the right subfigure we added the contour lines of K with $\theta = 2$ (e.g. when using the midpoint rule). The thick dotted line corresponds to method (3). If $\mu \rightarrow 1$, the local error constant increases towards 1, independent of θ . The diagram in Figure 4 confirms that method (3) is $[0,1]$ -stable if and only if $\mu < 0.839$. Moreover, the method (3) is very close to the method which is $[0,1]$ -stable for the largest set of μ -values. The thick dash-dotted line, for which $L_{\text{extr}} \equiv 0$, corresponds to the method based on a third order extrapolation using four points. This method is clearly no longer $[0,1]$ -stable if $\mu > 0.63$, cf. [10, 11].

The observation that no overall $[0,1]$ -stable scheme exists which uses four points and a second order extrapolation, can be generalized as follows.

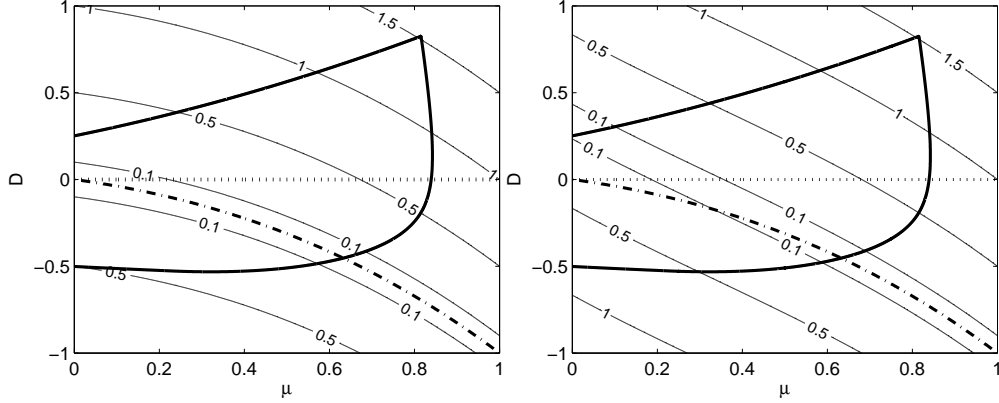


Figure 4: Diagrams showing the boundaries of $[0, 1]$ -stability (thick solid lines) and the contour lines of the local error constant K if $\theta = 0$ (left) or $\theta = 2$ (right). The thick dotted and dash-dotted line correspond to different acceleration schemes based on second order extrapolation using four points (see text for details).

Theorem 2 Consider an acceleration method of the form

$$\bar{y}_{j+1} = \Phi_{\Delta t}(\bar{y}_{j+1}^*) = \Phi_{\Delta t}(A(\mu)\bar{y}_j + B(\mu)\bar{y}_{j-1} + C(\mu)\bar{y}_{j-2} + D(\mu)\bar{y}_{j-3} + \dots)$$

with $A(\mu), B(\mu), C(\mu), \dots$ arbitrary functions of μ and $N \geq 3$. If the extrapolation method to find \bar{y}_{j+1}^* is of second (or higher) order, then the acceleration method is not overall $[0, 1]$ -stable.

Proof. We prove that for $\mu = 1$, there always exists an interval $[1 + \Delta\rho, 1]$ (with $\Delta\rho < 0$) which lies outside the stability region of the acceleration method if that method uses an extrapolation of second (or higher) order.

Define $A := A(1), B := B(1), C := C(1), \dots$. The general form of the characteristic polynomial at $\mu = 1$ is

$$C(\xi, \rho) := \xi^N - \rho(A\xi^{N-1} + B\xi^{N-2} + C\xi^{N-3} + \dots + X\xi^2 + Y\xi + Z) = 0. \quad (22)$$

If the extrapolation is of second order, $\xi = 1$ is a threefold zero of (22) if $\rho = 1$. Indeed,

$$C(1, 1) = 1 - A - B - C - \dots - X - Y - Z = 0,$$

$$(N-1)C(1, 1) - \frac{\partial C}{\partial \xi}(1, 1) = 1 + B + 2C + \dots + (N-3)X + (N-2)Y + (N-1)Z = 0,$$

$$\begin{aligned} (N-1)^2 C(1, 1) + (3-2N) \frac{\partial C}{\partial \xi}(1, 1) + \frac{\partial^2 C}{\partial \xi^2}(1, 1) = \\ -1 + B + 4C + \dots + (N-3)^2 X + (N-2)^2 Y + (N-1)^2 Z = 0. \end{aligned}$$

Therefore the characteristic equation (22) can be written as

$$C(\xi, \rho) := \xi^N - 3\rho\xi^{N-1} + 3\rho\xi^{N-2} - \rho\xi^{N-3} + \rho(\xi-1)^3 P(\xi) = 0,$$

with $N \geq 3$. If $N = 3$, then $P(\xi) = 0$, and if $N > 3$ then $P(\xi)$ is a polynomial of degree (at most) $N - 4$.

Since $C(\xi, \rho)$ is an algebraic curve, we can expand it near $(\xi, \rho) = (1, 1)$ as

$$C(1 + \Delta\xi, 1 + \Delta\rho) \approx -\Delta\rho + (1 + P(1))\Delta\xi^3 + \text{h.o.t.} = 0.$$

Locally, we can express $\Delta\xi$ in terms of $\Delta\rho$:

$$\Delta\xi \approx \sqrt[3]{\frac{1}{1 + P(1)}} \Delta\rho.$$

The three complex roots of this equation correspond to three branches of the curve defined by $C(\xi, \rho) = 0$ in the neighborhood $(1, 1)$. If $\Delta\rho < 0$, we can distinguish between three cases.

1. $P(1) < -1$. There is one positive real root $\Delta\xi$, and two complex roots with $\arg(\Delta\xi) = \pm\frac{2}{3}\pi$. The branch corresponding to the positive real root leaves the unit circle.
2. $P(1) > -1$. There is one negative real root $\Delta\xi$, and two complex roots with $\arg(\Delta\xi) = \pm\frac{1}{3}\pi$. Both branches corresponding to the complex conjugated roots leave the unit circle.
3. $P(1) = -1$. Since

$$C(\xi, 1) := \xi^{N-3}(\xi - 1)^3 + (\xi - 1)^3 P(\xi) = (\xi - 1)^3(\xi^{N-3} + P(\xi)) = 0,$$

$\xi = 1$ is now a fourfold root of the characteristic equation, and the extrapolation method is of third order. The expansion of $C(\xi, \rho)$ about $(\xi, \rho) = (1, 1)$ is

$$C(1 + \Delta\xi, 1 + \Delta\rho) \approx -\Delta\rho + \left(N + \frac{dP}{d\xi}(1)\right) \Delta\rho^4 + \text{h.o.t.} = 0,$$

leading to

$$\Delta\rho \approx \sqrt[4]{\frac{1}{N + \frac{dP}{d\xi}(1)}} \Delta\rho,$$

There are now 4 branches, and independent of the sign of $N + \frac{dP}{d\xi}(1)$, at least one branch will leave the unit circle. The same reasoning also holds for methods using higher orders of extrapolation.

For the three cases above, at least one branch of the curve defined by $C(\xi, \rho) = 0$ near $(\xi, \rho) = (1, 1)$ is leaving the unit circle. Therefore the acceleration methods are never $[0, 1]$ -stable at $\mu = 1$. By continuity, the acceleration methods will also not be $[0, 1]$ -stable for values of μ close to 1. ■

Remark 1 *It is also interesting to note where the reasoning in the proof above fails if the extrapolation is of first order. In that case we can derive that $\Delta\xi \approx \sqrt{1/(1 + P(1))} \Delta\rho$, and there are two branches defined by $C(\xi, \rho) = 0$ near $(\xi, \rho) = (1, 1)$. If $P(1) < -1$ and $\Delta\rho < 0$, the roots of this equation are real and one branch leaves the unit circle. If $P(1) > -1$ and $\Delta\rho < 0$, both roots are imaginary and both branches are tangent to the unit circle. In that case, the curvature of the branches determines whether they stay inside or if they leave the unit circle.*

The main conclusion of this section is that if we are interested in overall $[0,1]$ -stable methods of the form (18), we should only consider methods based on a first order extrapolation scheme. For these methods, the global error will increase at least quadratically in m or S . Therefore, the accuracy of these methods can only be improved by using more previous states to lower the constant factor in front of the m^2 term. In section 4.1.1, we showed that by using three points instead of only two, this factor decreased from 1 to $1/2$. By using more points, we can lower this factor even further. A numerical computation reveals that for $\mu = 1$, the most accurate $[0,1]$ -stable method using four points lowers the factor to about 0.38 (e.g. the scheme (20) with parameters $C = 1.443$ and $D = -0.275$).

5 MSEM Variant 2: Combination with the PM

In section 2 and 3 we showed that the PM is more accurate than the MSEM, but that the MSEM has better stability properties. In this section we investigate if a scheme exists which combines the benefits of both schemes. More specifically, we seek a scheme that combines the MSEM with the PM, which is overall $[0,1]$ -stable and more accurate than the MSEM. If we restrict ourselves to the linear extrapolation case for both the MSEM and the PM, such a combination scheme is for instance

$$\bar{y}_{n+1} = \Phi^k(\bar{y}_n + (\alpha F_{\text{PM}} + (1 - \alpha)F_{\text{MSEM}})m\Delta t), \quad (23)$$

with $k \geq 2, \alpha \in [0, 1]$ and

$$F_{\text{PM}} = \frac{\bar{y}_n - \Phi_{\Delta t}^{-1}(\bar{y}_n)}{\Delta t}, \quad F_{\text{MSEM}} = \frac{\bar{y}_n - \bar{y}_{n-1}}{(m+k)\Delta t}. \quad (24)$$

Let us again assume that $\Phi_{\Delta t}$ represents a first order accurate time integration step in which a local error $\theta/2\Delta t^2 y'' + \mathcal{O}(\Delta t^3)$ is made. Through a Taylor expansion about \bar{y}_n , we find that the local error of the combination scheme is

$$L := y(T_{j+1}) - \bar{y}_{j+1} = \frac{(2 - \alpha)m^2 + (k - \alpha(k - 1))m + \theta k}{2}.$$

For $\alpha = 0$ and $\theta = 1$, we retrieve equation (5). For $\alpha = 1$ and $\theta = 1$ however, we find a slightly modified version of equation (4), since we now expanded our Taylor series about another point. Using the technique described in section 3.2, we obtain

$$e = \mathcal{O}\left(\frac{(2 - \alpha)m^2 + (k - \alpha(k - 1))m + \theta k}{2(\alpha m + k)}\Delta t\right). \quad (25)$$

The stability of (23)–(24) is determined through the characteristic polynomial

$$\xi^2 = \rho^k \left(\xi + \left(\frac{\alpha(1 - \frac{1}{\rho})k\xi}{1 - \mu} + (1 - \alpha)(\xi - 1) \right) \mu \right). \quad (26)$$

Only values $\alpha(k, \mu)$ for which both zeroes of (26) are smaller than one in magnitude for all $\rho \in [0, 1)$ lead to $[0,1]$ -stable schemes. Figure 5 (left) shows the

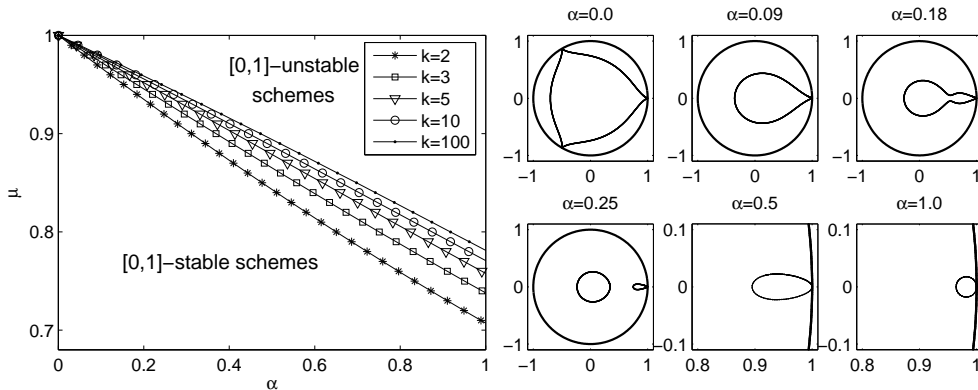


Figure 5: Left: the region in the $\alpha\mu$ -plane which corresponds to $[0,1]$ -stable schemes for different values of k ; this is the subset of $[0,1] \times [0,1]$ below the boundaries of $[0,1]$ -stability shown in the figure. Right: stability regions of the combination scheme (23)–(24) with $k = 3$, $\mu = 0.95$ and for different values of α .

numerically computed regions in the $\alpha\mu$ -plane (with $\alpha, \mu \in [0,1]$) that correspond to $[0,1]$ -stable schemes for different values of k . From this figure it is clear that in the $\alpha\mu$ -plane, the boundaries of $[0,1]$ -stability can quite well be approximated by lines of the form

$$\alpha = C_k(1 - \mu) \quad (27)$$

An approximation to the most accurate, overall $[0,1]$ -stable combination scheme is then obtained by choosing C_k as large as possible, subject to the constraint that the line (27) completely lies below the corresponding boundary of $[0,1]$ -stability. In other words, the combination scheme should resemble the PM as much as possible in order to be as accurate as possible, while it should still resemble the MSEM enough to remain overall $[0,1]$ -stable. The values of C_k can be determined numerically. It turns out that $C_2 = 3$, $C_3 \approx 3.65$, $C_4 \approx 3.96$ and $C_5 \approx 4.11$. In the remainder of this paper, we will refer to the scheme (23)–(24) with α given by (27) and the numerical values above as the “optimal PM-MSEM combination scheme”. As an illustration, Figure 5 (right) shows the stability regions of the combination scheme (23)–(24) with $k = 3$, $\mu = 0.95$ and for different values of α . For $\alpha = 0$ or $\alpha = 1$ we obtain the stability regions of the MSEM or the PM, and $\alpha = C_3(1 - \mu) \approx 0.18$ indeed corresponds to a scheme near the limit of $[0,1]$ -stability.

Using (27), the global error (25) becomes

$$e = \mathcal{O} \left(\frac{2m^3 + (3 - C_k)km^2 + ((1 - C_k)k^2 + (C_k + \theta)k)m + \theta k^2}{2k((1 + C_k)m + k)} \Delta t \right),$$

which can be approximated by

$$e = \mathcal{O} \left(\frac{1}{(1 + C_k)k} m^2 \right) = \mathcal{O} \left(\frac{k}{1 + C_k} S^2 \right). \quad (28)$$

as $m \rightarrow \infty$. Since C_k increases as k increases, the relative gain in accuracy compared to the MSEM (2) also increases with k (for the MSEM we obtained $e = \mathcal{O}(kS^2)$).

Table 3: The global error at $t_{\text{end}} = 0.1$ for the diffusion equation, computed with different acceleration schemes.

MSEM				TM			
$k \backslash S$	16	32	64	$k \backslash S$	16	32	64
1	3.84e-5	1.62e-4	6.63e-4	2	1.05e-5	4.25e-5	2.66e-4
2	7.69e-5	3.23e-4	1.33e-3	3	4.28e-5	2.58e-4	1.58e-3
3	1.15e-4	4.85e-4	2.00e-3	2(*)	1.00e-5	3.98e-5	1.58e-4

Variants of MSEM (with $k = 1$)				Optimal PM-MSEM Combi scheme			
$k \backslash S$	16	32	64	$k \backslash S$	16	32	64
Var. 1	1.86e-5	7.94e-5	3.28e-4	2	1.83e-5	7.88e-5	3.27e-4
Var. 2	1.12e-5	6.20e-5	2.90e-4	3	2.31e-5	1.01e-4	4.20e-4

6 Numerical Experiments

In this section, we use the different methods considered in this paper to accelerate an explicit time-stepper for the diffusion equation, the Fisher equation and for a model of a pendulum problem derived through a regularisation of the Euler-Lagrange differential-algebraic equations (DAEs). The experiments confirm many of the theoretical results derived in sections 3–5.

6.1 Example 1: the Diffusion Equation

The diffusion equation

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2}$$

with Dirichlet boundary conditions is solved on $x \in [0, 1]$ and $t \geq 0$. The spatial step size is $1/1000$, so there are 999 internal points and unknowns. A time-stepper was constructed using the forward Euler scheme with a step size of $2 \cdot 10^{-7}$, consistent with the stability limit of the forward Euler scheme for this problem. We first used the time-stepper for 512 time steps, starting from the initial condition $u(x, t) = |20x(x - 1/4)(x - 3/4)(x - 1)|$. The last solution of the preprocessing step is chosen as initial condition at $t = 0$. By using the data collected in this preprocessing step, the multistep schemes can also be used directly at $t = 0$. Different acceleration schemes were used to compute the solution at $t_{\text{end}} = 0.1 = 500000\Delta t$. We estimated the (global) error in the infinity norm by comparing with the solution of the unaccelerated time-stepper (instead of with the exact solution). Since t_{end} is not necessarily an integer multiple of the time step of the acceleration scheme, we used a cubic spline interpolation algorithm to obtain a solution at the end time.

The results are shown in Table 3. The top-left table was computed with the MSEM (2) for different values of k and S . We observe that the error behaves as $e = \mathcal{O}(m^2/k) = \mathcal{O}(kS^2)$. The top-right table was computed with the TM. For $k = m = 2$, the error behaves worse than expected. The problem is caused by the fact that the stability region of this method touches the real axis several

times. Therefore, some of the modes, which should be damped rapidly, now only damp very slowly. By choosing a somewhat smaller value of m , e.g. $m = 1.96$, the stability regions widens a little bit in the direction of the imaginary axis, resolving the problem. A similar damping of higher harmonics is also used in the context of Chebyshev methods [12]. The results using this modified TM are shown in the last row of the top-right table (indicated with 2(*)) — note that for these methods the value of S will be slightly smaller than indicated in the table header. We now observe that the error approximately grows as $\mathcal{O}(S^2)$. For $k = 3$, the problem above does not occur, and we directly observe $e = \mathcal{O}(S^{2.58})$. The bottom-left table was computed with the more accurate variants of the MSEM from section 4.1, based on more than two points and using $k = 1$. Variant 1 is the scheme based on (15) using three points. It is confirmed that the error is proportional to S^2 , and the error is only half as large as when the original MSEM was used. Variant 2 is the scheme based on (16) using three points. Asymptotically for large m , Variant 2 is the same as Variant 1. By construction however, the error will be smaller for lower values of m . Finally, the bottom-right table was computed with the optimal PM-MSEM combination scheme from section 5. We indeed see that for $k = 2$, the error is nearly the same as the error of Variant 1 with $k = 1$, and that the error growth is as expected from equation (28) as k increases.

Although the new schemes in the two bottom tables have a better accuracy than the original MSEM, the error of the TM with $k = 2$ is still about two times smaller than the error of the variants of the MSEM (when damping is used).

6.2 Example 2: the Fisher Equation

For the second example, we consider the Fisher equation

$$\frac{\partial u(x, t)}{\partial t} = \frac{\partial^2 u(x, t)}{\partial x^2} + \lambda u(1 - u), \quad (29)$$

and we used exactly the same setup as in Example 1. It is well known that if $\lambda > \pi^2$, all solutions of (29) tend towards a nonzero equilibrium solution. The results are shown in Table 4 for $\lambda = 100$. The numerical properties that were clearly recognizable in Table 3 are now somewhat obfuscated, but the same trends can be observed for this nonlinear reaction-diffusion equation. Again, the error of the TM with $k = 2$ is about two times smaller than the error of the variants of the MSEM (when damping is used).

6.3 Example 3: the Pendulum Problem

Our final example is taken from [5]. A pendulum with unit length and unit mass is described by the ODEs

$$\begin{aligned} x' &= u, & y' &= v \\ u' &= -2\lambda x, & v' &= -1 - 2\lambda y \end{aligned} \quad (30)$$

Table 4: The global error at $t_{\text{end}} = 0.1$ for the Fisher equation computed with different acceleration schemes.

MSEM				TM			
$k \backslash S$	16	32	64	$k \backslash S$	16	32	64
1	1.47e-5	6.12e-5	2.39e-4	2	4.01e-6	1.64e-5	1.49e-4
2	2.94e-5	1.21e-4	4.41e-4	3	1.64e-5	9.60e-5	4.83e-4
3	4.39e-5	1.78e-4	5.99e-4	2(*)	3.85e-6	1.52e-5	5.94e-5

Variants of MSEM (with $k = 1$)				Combination schemes MSEM-PM			
$k \backslash S$	16	32	64	$k \backslash S$	16	32	64
Var. 1	7.14e-6	3.03e-5	1.22e-4	2	7.02e-6	3.01e-5	1.22e-4
Var. 2	4.31e-6	2.37e-5	1.09e-4	3	8.86e-6	3.84e-5	1.55e-4

$$\lambda = \frac{(x^2 + y^2) - 1 + 4\epsilon(xu + yv)}{4\epsilon^2(x^2 + y^2)},$$

which are derived through a regularisation of the Euler-Lagrange DAEs. We used $\epsilon = 10^{-7}$ and the initial conditions $x = 0, y = -1, u = 2.5, v = 0$. Again, we built a time-stepper using the forward Euler scheme, now with step size $\Delta t = \epsilon$ (two eigenvalues of this time-stepper then lie near zero). As in the previous examples, we did a preprocessing step to provide the multistep acceleration schemes with the required initial data. We then used several TMs and (variants of) the MSEM to compute the solution at $t_{\text{end}} = 0.5 = 5 \cdot 10^6 \Delta t$. We again used a cubic spline interpolation algorithm to obtain a solution at t_{end} .

The result is shown in Figure 6. The global error shown is now the absolute error of only the component x . Quite surprisingly, the TM with $k = 2$ performs much worse than expected (also when damping is used). A detailed explanation of this phenomenon can be found in [4]. The main issue is that after each extrapolation step, the constraint of the DAE is no longer satisfied. As the constraining force in the equation for the velocity derivative in (30) then becomes very large, the values of variables u and v change rapidly in the first following forward Euler time step. In the second time step, not only u and v but also x and y change rapidly. Because of the specific choice of ϵ and Δt , the constraint is almost perfectly satisfied after this second time step, and therefore x, y, u and v will again behave smoothly. From this reasoning it is clear why the TM (or any scheme based on the PM) with $k = 2$ will be quite inaccurate and possibly unstable: the extrapolation is done based on the fast correction towards the manifold of the constraint instead of on the expected movement along the smooth trajectory. Figure 6 shows that methods based on the MSEM will not suffer from this problem, as they only use the result *after* the second time step. Using a TM with a larger value of k also remedies this problem, but as we have seen the error will then no longer behave as $\mathcal{O}(S^2)$. Another idea is to let the TM with $k = m = 2$ act not on the original time-stepper $\Phi_{\Delta t}$, but on $\Phi_{\Delta t}^2$ (i.e., each time step consists of two time steps with the original time-stepper). In that case the error will again behave as $\mathcal{O}(S^2)$, and its value

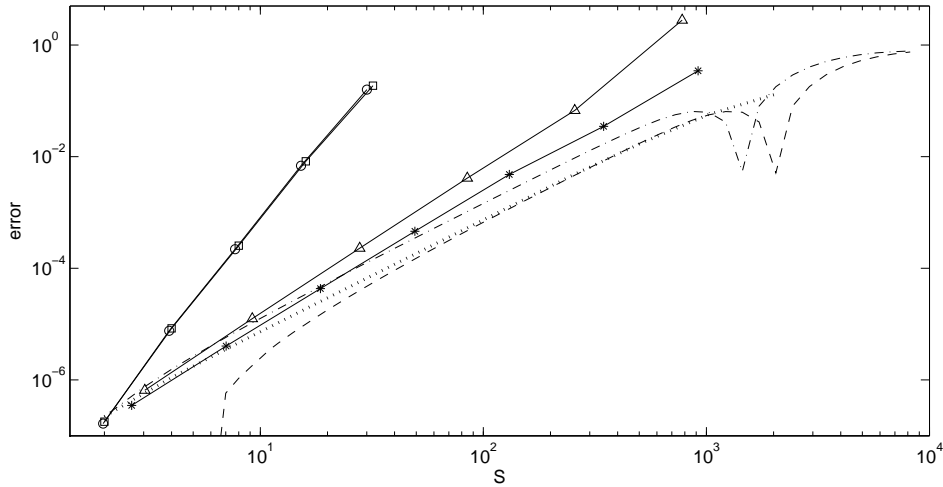


Figure 6: The global error at $t_{\text{end}} = 0.5$ for the pendulum problem. The solid lines are TMs: with $k = m = 2$ (square markers), with $k = 2$ and $m = 1.96$ (circle markers), with $k = 4$ and $m = 6.6$ (star markers) and with $k = 6$ and $m = 12.2$ (triangle markers). The dash-dotted line is the MSEM scheme (2) with $k = 1$ and the dashed line is the scheme (16) with $k = 1$. The dotted line is the TM with $k = m = 2$ applied to the time-stepper $\Phi_{\Delta t}^2$.

will be approximately the same as the error of the variants of the MSEM, since Examples 1 and 2 showed that the TM with $k = 2$ is about twice as accurate when applied to the same time-stepper. This is indeed confirmed in Figure 6.

7 Conclusions

In this paper we studied the accuracy of the Projective Method (PM), the Telescopic Method (TM) and the Multistep State Extrapolation Method (MSEM). These methods are an essential component in solving multiscale problems efficiently using the equation-free approach. We derived local and global error formulae for these methods, and showed that the global error of the MSEM can be much larger than the error of the PM, but that it is comparable to the global error of the TM and of first order Chebyshev methods (which are both also intended for parabolic problems). Furthermore, we studied the accuracy and stability of two classes of variants of the MSEM. We showed that in both cases more accurate, overall $[0,1]$ -stable methods can be found. However, we also showed that overall $[0,1]$ -stable methods should always be based on first order extrapolation schemes, and that the error is always of order $\mathcal{O}(S^2)$, with S the speedup. Therefore, there is not much room for a further substantial improvement of these methods' accuracy (without making further assumptions on the time-stepper). The theoretical results are confirmed using a linear and two nonlinear numerical examples. These experiments also show that the global error of the damped TM with $k = 2$ is slightly smaller than the global error of the variants of the MSEM. If however $k = 2$ cannot be used, e.g. as in Example

3, the variants of the MSEM might perform equally well or even better than the TMs.

Acknowledgements

It is a pleasure to acknowledge Prof. C. W. Gear for several interesting discussions, and for his comments on an early draft of this paper. This paper presents research results of the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office. The scientific responsibility rests with its authors.

References

- [1] V. Alexiades, G. Amiez, and P. Gramaud. Super-time-stepping acceleration of explicit schemes for parabolic problems. *Communications in Numerical Methods in Engineering, to appear*, 12:31–42, 1996.
- [2] W. E. Analysis of the heterogeneous multiscale method for ordinary differential equations. *Communications in Mathematical Sciences*, 1(3):423–436, 2003.
- [3] L. Eriksson, C. Johnson, and A. Logg. Explicit time-stepping for stiff ODEs. *SIAM J. Sci. Comput.*, 25(4):1142–1157, 2003.
- [4] C. W. Gear. Towards explicit methods for DAEs. *submitted to BIT*, 2005.
- [5] C. W. Gear and I. G. Kevrekidis. Projective methods for stiff differential equations: Problems with gaps in their eigenvalue spectrum. *SIAM J. Sci. Comput.*, 24(4):1091–1106, 2003.
- [6] C. W. Gear and I. G. Kevrekidis. Telescopic methods for parabolic differential equations. *J. Comp. Phys.*, 187(1):95–109, 2003.
- [7] W. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.*, volume 8. Springer-Verlag, 2nd edition, 1992.
- [8] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and K. Theodoropoulos. Equation-free multiscale computation: Enabling microscopic simulators to perform system-level tasks. *Comm. Math. Sciences*, 1(4):715–762, 2003.
- [9] V. I. Lebedev. Explicit difference schemes with variable time steps for solving stiff systems of equations. *Numerical Analysis and Its Applications, Proc. WNAA '96, Bulgaria, ed. Vulkov, Wasniewski, and Yalamov, Springer, Berlin*, pages 274–283, 1997.
- [10] B. P. Sommeijer. Increasing the real stability boundary of explicit methods. *Comp. Math. Applic.*, 19(6):37–49, 1990.

- [11] C. Vandekerckhove, D. Roose, and K. Lust. Numerical stability analysis of an acceleration scheme for step size constrained time integrators. *Journal of Computational and Applied Mathematics*, 2006.
- [12] J. G. Verwer. Explicit Runge-Kutta methods for parabolic partial differential equations. *Appl. Numer. Math.*, 22:359–379, 1996.