LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-JRNL-212640

Second-Order Accurate Projective Integrators for Multiscale Problems

Steven L. Lee, C. W. Gear

June 1, 2005

This revised article will appear in Journal of Computational and Applied Mathematics

Second-order accurate projective integrators for multiscale problems

Steven L. Lee [a] and C. William Gear [b]

[a] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory (LLNL), P.O. Box 808, Livermore, CA 94551*

[b] *Professor Emeritus, University of Illinois at Urbana-Champaign; Senior Scientist, Princeton University; and, LLNL Participating Guest. 17 Honeybrook Drive, Princeton, NJ 08540*

Abstract

We introduce new projective versions of second-order accurate Runge-Kutta and Adams-Bashforth methods, and demonstrate their use as outer integrators in solving stiff differential systems. An important outcome is that the new outer integrators, when combined with an inner telescopic projective integrator, can result in fully explicit methods with adaptive outer step size selection and solution accuracy comparable to those obtained by implicit integrators. If the stiff differential equations are not directly available, our formulations and stability analysis are general enough to allow the combined outer-inner projective integrators to be applied to black-box legacy codes or perform a coarse-grained time integration of microscopic systems to evolve macroscopic behavior, for example.

Key words: Stability, stiff, explicit, teleprojective integration, parabolic, multiscale

1 Introduction

Projective integration methods were first introduced [8] as explicit methods for solving stiff systems that have a relatively large gap between their fast and slow time scales. The fast

Email addresses: slee@llnl.gov (Steven L. Lee), wgear@princeton.edu (C. William Gear).

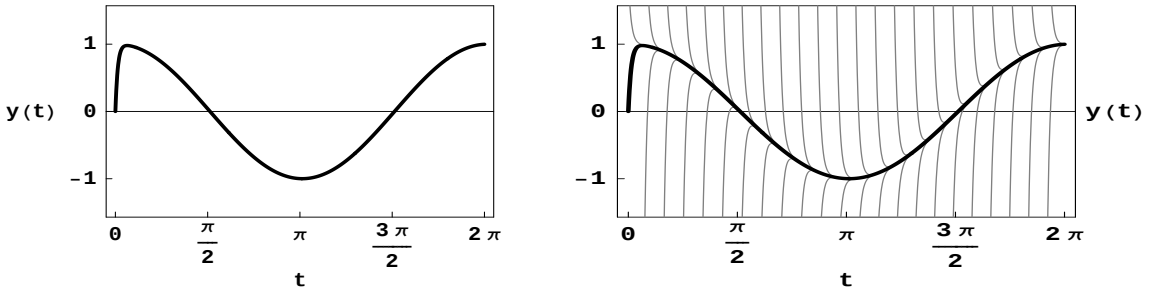
scales die out quickly but make the system stiff; they correspond to the Jacobian eigenvalues with large negative real parts for the linearized problem. The slow scales correspond to eigenvalues of smaller magnitude and are the solution components of practical interest. For stiff systems without a large gap in their wide range of time scales, telescopic projective methods were developed and are described in [9]. The conventional wisdom, however, is that all reasonable methods for stiff systems have to be implicit, and that some type of Newton-like method must be used to solve the nonlinear system and thereby avoid a severe restriction on the step size [3]. Regardless of the use of explicit or implicit methods, the computational challenge of stiff multiscale systems stems from trying to resolve efficiently the slow time scale behavior while incorporating the effects from the fast and intermediate time scales.

We are primarily concerned with the integration of stiff ordinary differential equation (ODE) initial value problems

$$y'(t) = f(t, y), \qquad y(t_0) = y_0 \tag{1}$$

where y and f are N-dimensional vectors and N is large. Such large systems arise from the method of lines spatial discretization of parabolic partial differential equations (PDEs), for example. But we also have in mind problems for which it is not practical to use implicit methods. This could occur because a black-box, legacy code is being used to carry out the time-stepping for the stiff system, and larger time steps are desired. Alternatively, the actual size of the problem or severe nonlinearities may make the use of Newton-like methods problematic on, say, massively parallel machines. The latter is related to the difficulty of developing robust and scalable preconditioners for multi-dimensional PDEs. Finally, projective integration is a cornerstone to enabling the Equation-free (EF) and Heterogeneous Multiscale Method (HMM) approaches to solving multiscale problems [12,5]. The mathematical analysis and algorithmic development of projective integration is essential for addressing these various multiscale research topics.

Projective integration has a wide range of applicability and, among all classes of integrators, is most closely related to stabilized explicit methods. This class of explicit methods has finite but greatly elongated stability regions that are often suitable for parabolic problems with negative and strictly real eigenvalues. This elongation is accomplished by using a sequence of step sizes that gives an enlarged stability region, together with a low-order polynomial approximation to the exponential operator for that stability region. Second-order accurate methods are implemented in the codes `RKC` [17] and `ROCK2` [2]. Higher-order stabilized explicit methods are given by `DUMKA3` [14] and `ROCK4` [1], for example. The HMM approach for ODEs [4,6] and the stabilization techniques given in [7] are also closely related techniques. As we are mainly interested in an initial study of second-order accurate projective integration methods with respect to implicit methods, a direct comparison with (more mature) stabilized explicit methods will be reported elsewhere.



(a) Plot for $y(0) = 0$ (b) Plot for $y(0) = 0$, and other initial values

Fig. 1. Solutions for $y'(t) = -25\ [y(t) - \cos(t)]$.

2 Projective integration methods

The idea behind projective integration methods is best conveyed by applying it to a simple multiscale problem. In particular, we examine the scalar, stiff initial value problem

$$y'(t) = -25\ [y(t) - \cos(t)] \tag{2}$$

for $y(0) = 0$ and $t \in [0, 2\pi]$. The exact solution

$$y(t) = 625/626\ [\cos(t) + 1/25 \sin(t) - \exp(-25t)] \tag{3}$$

has a fast time scale due to the exponential term. After that fast term essentially vanishes at $t \approx 1$, we mainly want to resolve the smooth and slowly varying behavior due to the trigonometric terms. Figure 1(a) shows the exact solution $y(t)$. That solution, along with solutions corresponding to different initial values, are shown in Figure 1(b). Despite the dramatic bunching of all trajectories toward a common cosine-like manifold, solution curves for different initial values do not actually merge or intersect—they only approach the slow manifold asymptotically. Note that this problem has only one fast component. In general, we will be interested in systems with many fast components of differing speeds (e.g., the multi-dimensional heat equation or other dissipative parabolic PDEs).

Forward Euler, and other conventional explicit methods, are notoriously inefficient for numerically integrating stiff initial value problems. The fundamental difficulty is that explicit methods are adversely affected by the behavior of nearby solution curves. The trouble begins when a predicted solution is slightly off the slow manifold, and on a nearby solution curve that steeply approaches the manifold. The slope of that predicted solution is therefore extremely large in magnitude because of the steep approach of the so-called fast transients. This wrong impression of how the actual solution is changing forces the explicit method to take relatively small step sizes because the large slope implies the current solution is changing rapidly. This curse of stiffness persists throughout the entire time interval, except for a short initial phase when the fast time scales have not decayed greatly. For step sizes that are too large, the typical situation is that the predicted solutions become increasingly worse

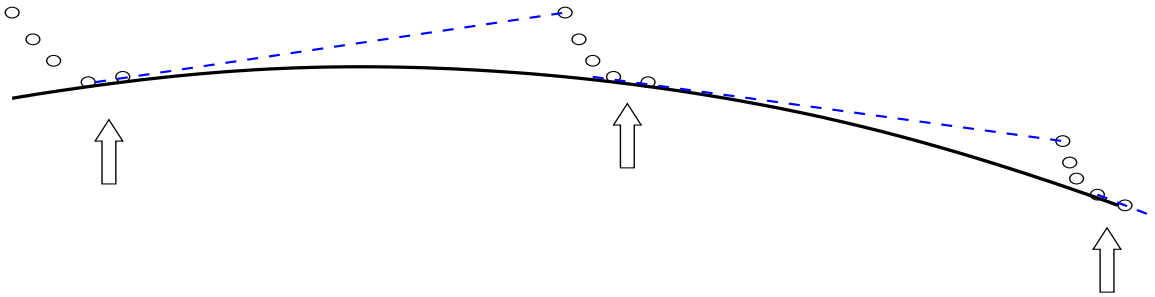


Fig. 2. Example of projective forward Euler taking small, damping steps followed by a large projective step. The black curve is the slow manifold for the stiff problem. The open dots are the solution values that stably follow the fast transients toward the manifold. The arrows indicate where chord slopes are evaluated for taking the projective step along the dashed line.

and move further away from the manifold; that is, the explicit method becomes unstable.

Projective integration methods are explicit methods that can be tailored to efficiently exploit the multiscale features that are characteristic of stiff systems; namely, fast transients asymptotically merging into a slow manifold. Intuitively, the idea of projective integration is straightforward. Using Figure 2 as a rough example, an *inner integrator* (e.g., forward Euler) is applied with a constant step size small enough to follow stably the fast transients toward the slow manifold. After several of these steps, a chord slope is computed based on the current and previous solution values. That slope is essentially an estimate of how the slow manifold is changing. Using the current estimate of the solution and its chord slope near the manifold, a large projective step is taken in the direction of the chord. The combination of the inner damping steps and the projective step constitutes one outer projective forward Euler step. The important result is that the predicted solution is now based on the estimated behavior of the manifold rather than that of the transients. The small, damping steps use a step size commensurate with the fast time scales, and neutralize the fast transients. The projective step size can then be chosen based on the slow time scale behavior and accuracy requirements of the stiff system.

Projective integration is a process that can be applied to a legacy code, the output from a microscopic simulation of a more detailed model, or any other type of step-by-step integrator. The process is exemplified by the projective forward Euler method.

Projective Forward Euler (PFE)

(i). Using a suitable inner integrator, advance k steps of length h_0 from y_n to obtain the solution y_{n+k}. For the inner integrator, the main requirements are that it is stable and of at least first-order accuracy. Since we want explicit methods, the step size h_0 is chosen for stability and typically such that the method is strongly damping for the fastest components. For example, if the inner integrator is forward Euler, the best choice is the reciprocal of the spectral radius of the Jacobian $\partial f/\partial y$; that is, $h_0 = 1/\max_i |\lambda_i|$. The purpose of these k steps is to damp the fast components.

(ii). Perform one more inner integration step to compute y_{n+k+1}, and use it to approximate

the derivative as the chord slope

$$v'_{n+k+1,n+k} = \frac{y_{n+k+1} - y_{n+k}}{h_0} \tag{4}$$

(iii). Perform the projective step

$$y_{n+s} = y_{n+k+1} + (Mh_0)\ v'_{n+k+1,n+k} \tag{5}$$

to advance the solution a distance Mh_0 for real M. The chord slope calculation in eq. (4) and projective step in eq. (5) are vector operations that are cheap to compute.

Defining $s = k + 1 + M$, the total length of this PFE integration step (called an *outer integration step*) is sh_0. While the notation and our examples may suggest that M and s are integers, there is no such restriction and y_{n+r} should be interpreted as the approximation to y at the time $t_n + rh_0$ for any real r.

Projective integration can be applied efficiently to stiff systems that have a wide separation between the fast and slow time scales. A key requirement is that the fast scales are well-clustered so that the inner steps are effective in damping the transients toward the low-dimensional, slow manifold. For the stiff initial value problem $y' = f(t, y)$, the fast scales correspond to the set of Jacobian $\partial f/\partial y$ eigenvalues that have the largest and most negative real parts. The slow scales correspond to the set of eigenvalues that are significantly closer to the origin, relative to the fast scales. If each of these sets is tightly clustered and a relatively large gap exists between the time scales, then the stiff system can be integrated with an outer integrator step that internally consists of: a few inner damping steps to neutralize the fast scales, then a chord slope calculation near the manifold and a large projective step to follow the slow scales. The length of the projective step, and consequently the overall efficiency of the projective integrator, is strongly related to the size of the gap. For stiff systems with no clear separation of scales, telescopic projective (i.e., teleprojective) integrators are efficient methods for carrying out the time integration.

3 Teleprojective integration

As presented in [9], we can iterate (or telescope) the projective integration process by using the current outer projective integrator as an inner integrator within yet another outer projective integrator. This layering of projective integrators can be repeated as many times as desired. Figure 3 gives an example of PFE, with $L = 2$ layers, that generates a telescopic PFE step of $100h_0$. At the innermost layer ($L = 0$), the integrator steps are of size h_0. At higher layers, the number of inner damping steps is $k = 3$ and the multiplier for the projective step is $M = 6$. For layer 1 and layer 2, the projective steps are $6h_0$ and $60h_0$, respectively. If an additional layer of PFE were used, it would have 4 inner steps of $100h_0$, a projective step of $600h_0$, and yield a telescopic step of $1,000h_0$.

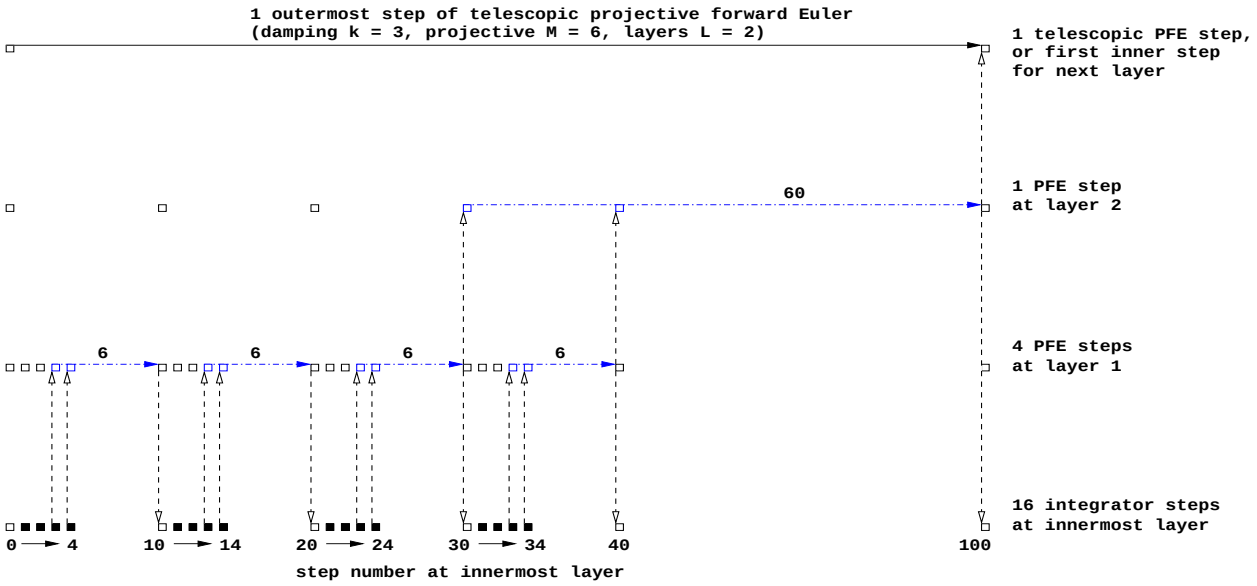


Fig. 3. Projective forward Euler with $L = 2$ layers of telescoping.

Naturally, the stability and accuracy of teleprojective integration depends on a suitable choice of values for h_0, k, M and L for the stiff system. Note that if the innermost integrator were conventional forward Euler, the chord slope $v'_{n+k+1,n+k}$ calculated in eq. (4) would be exactly $f(t_{n+k}, y_{n+k})$. The use of function evaluations for directly calculating derivatives is common to Runge-Kutta-Chebyshev methods such as [1,2,17,14], and the explicit time stepping methods for stiff ODEs described in [7]. In contrast, we use chord slopes because we want to accommodate legacy codes or inner integrators other than forward Euler, and the use of chord slopes is an essential ingredient of teleprojective integration (see Figure 3). Chord slopes also simplify the study of stability because the properties of the outer integrator can be analyzed independently of the choice of the inner integrator [8].

Further discussion and analysis are greatly facilitated if we assume that at each layer of telescoping, k and M are constant. We restrict ourselves to systems in which any positive Jacobian eigenvalues are small, and initially we assume that all eigenvalues are close to the real axis. This allows us to consider stability along the real axis only, and infer instability in its neighborhood by continuity.

The step size h_0 is chosen so that the one-step error amplification of the innermost integrator, $\rho(h_0\lambda)$, satisfies $|\rho| < 1$ for all eigenvalues λ, except for those with nonnegative real parts. The linear stability polynomial for one PFE step is

$$\sigma(\rho) = \rho^{k+1} + M(\rho^{k+1} - \rho^k) = [(M+1)\rho - M]\,\rho^k. \tag{6}$$

Note that $\sigma(0) = 0$ and $\sigma(1) = 1$ independent of k and M. The stability region for given values of k and M can be obtained by plotting $|\sigma(\rho)| = 1$ for ρ in the complex-plane. If the stability region includes all $\rho \in [0, 1]$; we refer to it as a $[0, 1]$-stability region and such

Table I

Critical values for $[0,1]$-stable projective forward Euler

	A			B		
	PFE ($L=1$ layer)			Telescopic PFE ($L>1$ layers)		
k	M_0	β	$\widehat{\rho}$	M_∞	β	$\widehat{\rho}$
1	4.8284	0.1716	0.4142	2	0.3333	0.3333
2	8.4435	0.2980	0.5961	3	0.25	0.5
3	12.0446	0.3881	0.6925	6.6560	0.4613	0.6520
4	15.6411	0.4555	0.7519	8.3172	0.4326	0.7141
5	19.2357	0.5081	0.7922	12.2147	0.5520	0.7703

integrators as $[0,1]$-stable. For PFE, with $L>1$ layers, the stability polynomial is

$$\sigma_{j+1}(\rho) = [(M+1)\sigma_j(\rho) - M]\,\sigma_j^k(\rho) \tag{7}$$

for $j = 1,\ldots,L-1$ where $\sigma_1(\rho) = \sigma(\rho)$ is given in eq. (6). If the given values for k, M and L result in a stability region that includes all $\rho \in [0,1]$, then the teleprojective integrator is $[0,1]$-stable.

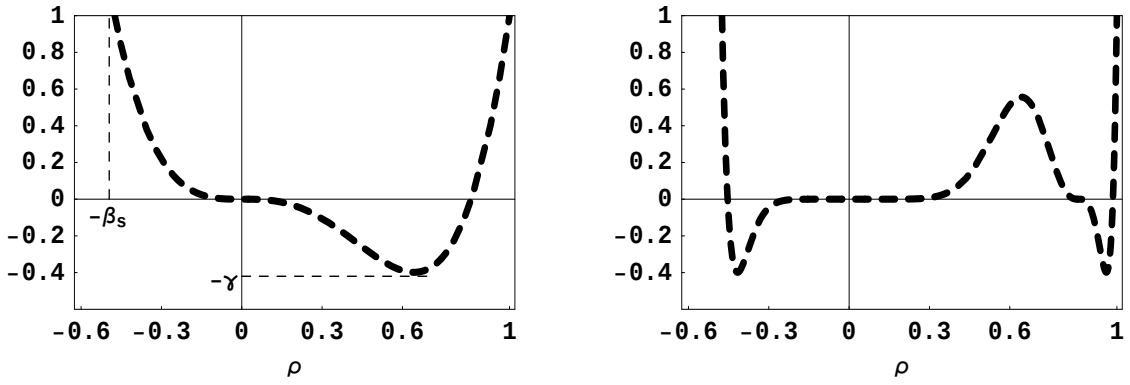
In the analysis that follows, we study the stability interval for real values of ρ rather than the entire stability region in the complex ρ-plane. This is sufficient for parabolic problems and, as we will see, choosing less aggressive values for the parameters results in stability regions that include a neighborhood of $[0,1)$; see, for example, Figure 5(a).

We are primarily interested in $[0,1]$-stable methods which have an interval of stability on the real axis that includes $[0,1]$. Integrators that are not $[0,1]$-stable are still potentially useful for systems with gaps in the eigenvalue spectrum. The chief advantage of $[0,1]$-stable integrators is that no separation of time scales is assumed or required. Let the real stability domain that includes $[0,1]$ be $[-\beta_s, 1]$ for nonnegative β_s. (Because we require at least first-order accuracy, $1+\epsilon$ cannot be stable for small $\epsilon > 0$.) Let the real stability range of $\sigma(\rho)$ for $\rho \in [-\beta_s, 1]$ be $[-\gamma, 1]$ for nonnegative γ.

Table I gives critical values regarding $[0,1]$-stability for PFE and telescopic PFE. These values are nonnegative real numbers, with $\beta \in [0,1]$ and $\widehat{\rho} \in [0,1]$. For a given value of k, Table IA means the following. For $0 \le M < M_0$, the PFE real stability domain $[-\beta_s, 1]$ always includes $[-\beta, 1]$ because at these lesser values of M, we have $|\sigma(\rho)| < 1$ for all ρ within the interval $[-\beta, 1]$. For $M = M_0$,

$$|\sigma(\widehat{\rho})| = |\sigma(-\beta)| = 1 \tag{8}$$

and $\beta_s = \beta$. Finally, for $M > M_0$, $[0,1]$-stability is lost because $|\sigma(\rho)|$ exceeds one for ρ in the



(a) $\sigma(\rho)$ for $L = 1$ layer (b) $\sigma(\rho)$ for $L = 2$ layers

Fig. 4. Real stability polynomial $\sigma(\rho)$ for PFE ($k = 3, M = 6$)

$[0, 1]$-interval. The rule of thumb for PFE is that the method is $[0, 1]$-stable if $M \leq 3.59\,k$; a derivation of this condition is given in [8]. For Table IB, and a given value of k, the meaning of β and $\widehat{\rho}$ is slightly different. For $0 \leq M < M_\infty$, the telescopic PFE real stability domain $[-\beta_s, 1]$ always includes $[-\beta, 1]$—regardless of the number of layers used. For $M = M_\infty$,

$$\sigma(\widehat{\rho}) = -\beta \tag{9}$$

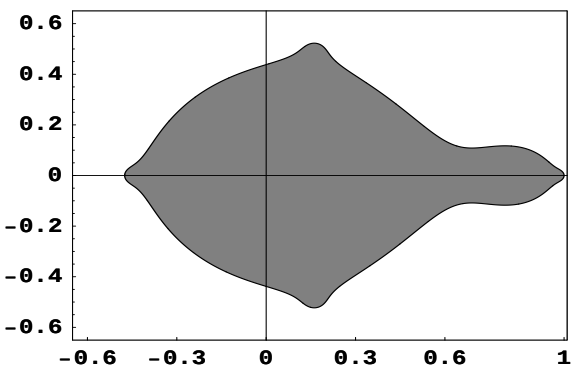
and $\gamma = \beta$. Hence $\sigma(\rho)$ maps $[-\gamma, 1]$ onto itself. In general, for $[0, 1]$-stable real polynomials where $[-\gamma, 1] \subseteq [-\beta, 1]$, we will informally say that the γ-range is less than or equal to the β-domain. For telescopic PFE and a given value of k, this condition is significant because it ensures that, when telescoping is applied, the integrator remains $[0, 1]$-stable all the way up to the top layer since at each layer its γ-range is not wider than $[-\beta, 1]$. For reference, we note that the $[0, 1]$-stability-related intervals for telescopic PFE are nested as follows for $0 \leq M \leq M_\infty$:

$$[-\gamma, 1] \subseteq [-\beta, 1] \subseteq [-\beta_s, 1] \subseteq [-1, 1]. \tag{10}$$

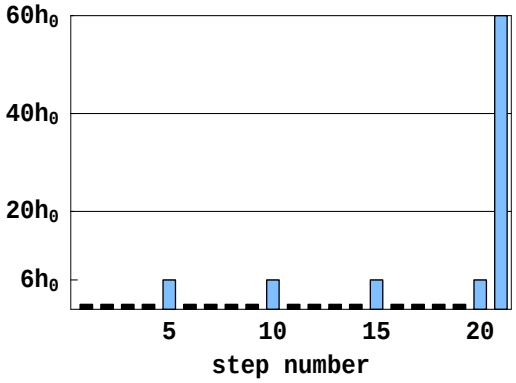
A detailed analysis of the $[0, 1]$-stability of telescopic PFE is given in [9].

Figures 4(a) and 4(b) show plots of the real stability polynomial for PFE ($k = 3$, $M = 6$) when the number of layers is $L = 1$ and $L = 2$. The integrator remains $[0, 1]$-stable for arbitrarily many layers because, from Table IB and for $k = 3$, we have $M = 6 < 6.6560 = M_\infty$.

We conclude the discussion of the telescopic example by examining its stability region and step size sequence in generating a telescopic step of $100h_0$. As evident in Figure 5(a), the integrator is $[0, 1]$-stable since the stability region encloses all real $\rho \in [0, 1]$. From Figure 5(b), we find that the telescopic step of $100h_0$ is actually the sum of 21 step sizes: 16 innermost integrator steps and 5 projective steps. The main cost of the telescopic step comes from performing the 16 innermost integrator steps of size h_0, which damp and stabilize the fastest time scale of the stiff system. The projective steps, which are larger, develop and stabilize



(a) Stability region in complex ρ-plane



(b) Step size sequence for telescopic step $100h_0$

Fig. 5. Stability region and step size sequence for PFE ($k = 3, M = 6, L = 2$)

the system at the fast and intermediate time scales. The key result is that the layering of $[0, 1]$-stable PFE integrators has the effect of sequencing the variety of step sizes so that internal instabilities do not arise in each telescopic step.

In general, using L layers of PFE generates a telescopic step of size $(k + 1 + M)^L h_0$ using only $(k + 1)^L$ innermost integrator steps. Relative to forward Euler, the efficiency of PFE improves by a factor of $(k+1+M)/(k+1)$ per layer. This efficiency stems from the fact that the projective steps, at layers $j > 1$, are based on chord slopes rather than direct evaluations of $f(t, y)$. The chord slopes are calculated cheaply as a finite difference of previously computed solutions. But clearly we cannot continue adding layers and improving efficiency indefinitely. The limiting consideration is that the size of the telescopic step needs to be based on the accuracy requirements for the stiff system. Furthermore, we want also to investigate the case where the step taken at the final layer is more than first-order accurate. Therefore, with the objective of developing the solution of a stiff system for accuracy, we will do so by using telescopically large but stable step sizes for the inner damping steps of a second-order accurate outer projective integration method. The inner steps will be generated by $[0, 1]$-stable telescopic PFE and the outer integrator will be the projective counterpart to conventional second-order accurate Runge-Kutta and Adams-Bashforth methods.

4 Second-order accurate projective integrators

One way to develop a second-order accurate projective integrator is to model it after the conventional trapezoidal method for ODEs. In the latter, a forward step is taken based on a weighted average of the slope at the current time step and the predicted slope at the next time step. The projective counterpart involves taking several damping steps prior to estimating a chord slope near the current and next time step. Then, the projective step to the next time step is taken based on a weighted average of those chord slopes. As first derived in [8], the weights are based on the number of inner damping steps k and the multiplier M for the projective step. An example of this approach is given in the top layer of Figure 6.

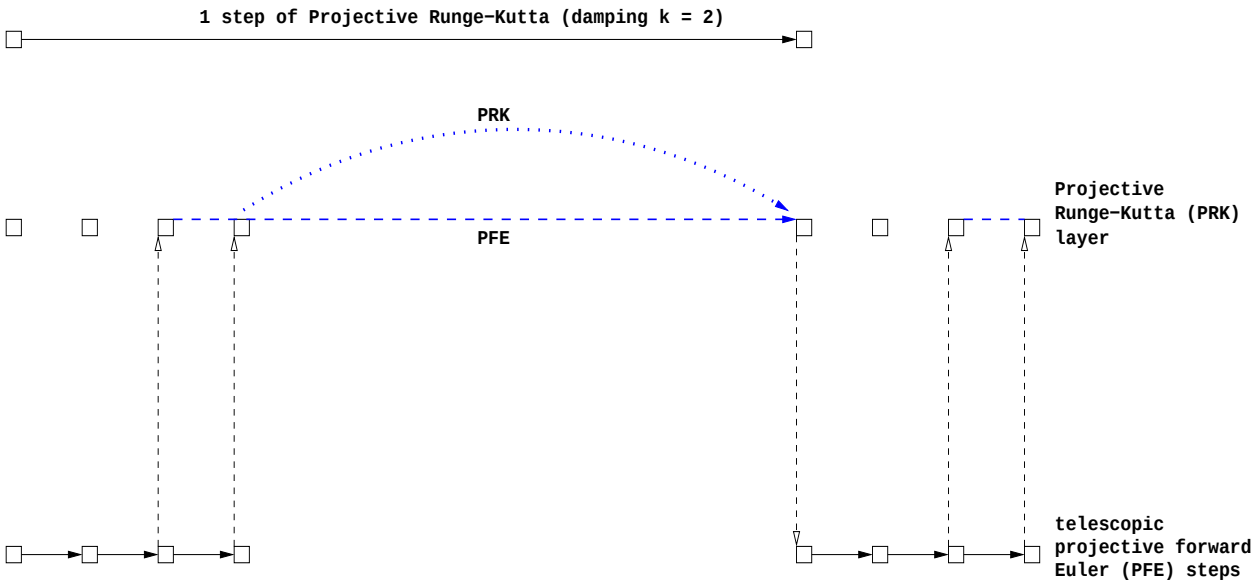


Fig. 6. Projective Runge-Kutta as an outer integrator for telescopic projective forward Euler.

Such predictor-corrector implementations are useful because an error estimate can be made from their difference. If reliable, a norm of the error estimate can be used to either reject the current step or select the next step size for the outer integrator. A full development of efficient error estimation techniques for projective integrators is beyond the scope of this article, but it is certainly a topic worthy of further research. The main drawback for conventional predictor-corrector methods is that they require the expense of (at least) two slope evaluations per step. In contrast, conventional Adams-Bashforth methods require only one slope evaluation per step. Therefore, as a potentially more cost-effective alternative to projective Runge-Kutta, we will also develop a projective counterpart to the Adams-Bashforth method.

4.1 Projective Runge-Kutta

The projective Runge-Kutta (PRK) method we derive has two stages. The formula is

$$y_{n+s} = y_{n+k+1} + Mh_0 \left[\alpha v'_{n+k+1,n+k} + (1-\alpha) v'_{n+s+k_1+1,n+s+k_1}\right], \tag{11}$$

where k and k_1 are the number of inner damping steps used at the step starting from t_n and t_{n+s}, respectively. The real scalars α and M are the method coefficients for the projective step, and the latter v' term in eq. (11) is the chord slope calculated after the initial PFE step to t_{n+s}. The stability polynomial is

$$\sigma_{\mathrm{PRK}}(\rho) = \rho^{k+1} + M\left[\alpha(\rho^{k+1} - \rho^k) + (1-\alpha)(\rho^{k_1+1} - \rho^{k_1})\sigma_{\mathrm{PFE}}(\rho)\right], \tag{12}$$

and it closely resembles the formula (11). Note that $\rho^{k+1} - \rho^k$ corresponds to the calculation of the chord slope near the current step t_n, and σ_{PFE} is the PFE stability polynomial from eq. (6).

The standard local truncation error analysis proceeds by determining the error in y_{n+s} under the assumption that $y_n = y(t_n)$. We compute this error, and then determine the weighting coefficient α that eliminates the second-order error term. To begin, we need to know the second-order error introduced at each step of the inner integrator. Since we do not want to specify what inner integrator is used, assume that the inner integrator starts from the correct solution $y(t_n)$ and takes one step of size h_0; that is,

$$y_{n+1} = y(t_n) + h_0 y'_n + \frac{(1-\xi_n)h_0^2}{2} y''_n + \mathcal{O}(h_0^3) = y(t_{n+1}) - \frac{\xi_n h_0^2}{2} y''_n + \mathcal{O}(h_0^3). \tag{13}$$

Thus, ξ_n is the coefficient of the scaled second-order error term in the inner integrator. If the inner integrator is forward Euler, $\xi_n = 1$. If the inner integrator is second- or higher-order (possibly a legacy code), $\xi_n = 0$. Note that because the inner integrator is at least first-order, all intermediate terms in an integration starting from the correct value $y(t_n)$ will have no lower than second-order errors.

The analysis continues by writing the projective Runge-Kutta formula (11) in a way that keeps through second-order terms in h_0. We replace terms like $h_0 v'_{n+k+1,n+k}$ with

$$\begin{aligned} h_0 v'_{n+k+1,n+k} &= y_{n+k+1} - y_{n+k} = h_0 y'_{n+k+1/2} - \frac{h_0^2 \xi_{n+k}}{2} y'' + \mathcal{O}(h_0^3) \\ &= h_0 y'(t_{n+k+1/2}) - \frac{h_0^2 \xi_{n+k}}{2} y'' + \mathcal{O}(h_0^3) \end{aligned} \tag{14}$$

and we replace y_{n+k+1} with

$$y_{n+k+1} = y(t_{n+k+1}) - \frac{h_0^2}{2} \sum_{i=0}^{k} \xi_{n+i} y'' + \mathcal{O}(h_0^3). \tag{15}$$

Note that we do not need to specify the point at which y'' is evaluated since any change (within the range of the step) introduces only third-order error terms. With these substitutions, the PRK formula (11) can be written as

$$\begin{aligned} y_{n+s} = y(t_{n+k+1}) &- \frac{h_0^2}{2} \sum_{i=0}^{k} \xi_{n+i} y'' + M h_0 [y'(t_{n+k+1+M/2}) \\ &+ \alpha [y'(t_{n+k+1/2}) - y'(t_{n+k+1+M/2})] + (1-\alpha)[y'(t_{n+s+k_1+1/2}) - y'(t_{n+k+1+M/2})]] \\ &- \frac{M h_0^2}{2} \left[\alpha \xi_{n+k} + (1-\alpha)\xi_{n+s+k_1}\right] y'' + \mathcal{O}(h_0^3). \end{aligned} \tag{16}$$

Noting that

$$y(t_{n+k+1}) + Mh_0 y'(t_{n+k+1+M/2}) = y(t_{n+s}) + \mathcal{O}(h_0^3) \tag{17}$$

we have

$$y_{n+s} = y(t_{n+s}) - \frac{Mh_0^2 y''}{2}\left[2\alpha(M+1+k_1) - \left(M+1+2k_1 - \frac{1}{M}\xi_n^{\mathrm{PRK}}\right)\right] + \mathcal{O}(h_0^3) \tag{18}$$

where

$$\xi_n^{\mathrm{PRK}} = \sum_{i=0}^{k} \xi_{n+i} + M\left[\alpha\xi_{n+k} + (1-\alpha)\xi_{n+s+k_1}\right]. \tag{19}$$

For second-order accurate projective Runge-Kutta,

$$\alpha = \frac{M+1+2k_1 - \frac{1}{M}\xi_n^{\mathrm{PRK}}}{2(M+1+k_1)} = \frac{M+1+2k_1 - \frac{1}{M}\left[\sum_{i=0}^{k}\xi_{n+i} + M\xi_{n+s+k_1}\right]}{2(M+1+k_1) + \xi_{n+k} - \xi_{n+s+k_1}} \tag{20}$$

since this choice eliminates the second-order error term in eq. (18). If the inner integrator is forward Euler, $\xi_n = 1$ is independent of the step number n, so $\xi_n^{\mathrm{PRK}} = (k+1+M)\xi_n = s$.

If the outer projective Runge-Kutta integrator uses telescopic PFE for each inner damping step, all we need is the form of the error term in each layer of the inner integrator and expressed in the manner of (13). We define $\xi_{n,j}$ to be the scaled error coefficient for the j-th layer at step n. If $\xi_{0,n}$ is independent of n (it is one for a forward Euler inner integrator, and zero for a second- or higher-order inner integrator) and if all lower layers use the same values of k and M, we can easily show that $\xi_{n,j}$ is independent of n. Writing $\xi_{n,j} = \xi_j^*$ when $\xi_{n,j}$ is independent of n, we have

$$\xi_{j+1}^* = \frac{M(M+1)}{s^2} + \frac{\xi_j^*}{s}. \tag{21}$$

When, as is likely to happen in a production code, k and M change from step to step and layer to layer, $\xi_{n,j}$ for layer j can be computed on-the-fly. The cost is trivial because, for up to second-order accuracy, it is sufficient to match the first three terms in the power series expansion of the solution of $y' = \lambda y$. This allows the $\xi_{n,j}$ term to be calculated for each layer of PFE. If the $\xi_{n,j}$ terms at each of the $k+1$ steps at the j-th layer are $\xi_{n+i,j}$ for $i = 0, 1, \ldots, k$, then at the next layer

$$\xi_{n,j+1} = \frac{M(M+1)}{s^2} + \frac{\left(\sum_{i=0}^{k}\xi_{n+i,j} + M\xi_{n+k,j}\right)}{s^2}. \tag{22}$$

Table II

Critical values for $[0,1]$-stable second-order accurate projective integrators

	A			B		
	Projective Runge-Kutta			Projective Adams-Bashforth		
k	M_0	β	$\widehat{\rho}$	M_0	β	$\widehat{\rho}$
1	7.7958	0.1137	1/2	2.1747	0.3150	0.4142
2	14.1501	0.3333	2/3	4.3115	0.2980	0.5961
3	20.4726	0.3310	3/4	6.4480	0.4655	0.6925
4	26.7848	0.4847	4/5	8.5844	0.4555	0.7519
5	33.0924	0.4596	5/6	10.7208	0.5652	0.7922

Table IIA shows critical values for $[0,1]$-stable projective Runge-Kutta. We assume forward Euler is the inner integrator, which means $L = 0$ and $\xi_{n,0} = 1$. We also assume $k = k_1$ so that the number of inner damping steps is the same at t_n and t_{n+s}, respectively. An examination of the PRK real stability polynomial reveals that as M becomes large, eventually $[0,1]$-stability is lost because $\sigma_{\mathrm{PRK}}(\rho)$ peaks at $+1$ for ρ within $[0,1]$. For a given value of k, the critical values for M and ρ (i.e., M_0 and $\widehat{\rho}$) can be obtained by solving the polynomial equations $\sigma_{\mathrm{PRK}}(\rho) = 1$ and $\partial\sigma_{\mathrm{PRK}}(\rho)/\partial\rho = 0$. In general, for any integer $k > 0$,

$$\widehat{\rho} = k/(k+1) \tag{23}$$

and additional calculations in `Mathematica` reveal that

$$M_0 = \frac{1 - \eta + \sqrt{1 + k(2-\eta)\eta}}{\eta} \tag{24}$$

where $\eta = k^k(k+1)^{-(k+1)}$. For $0 \le M < M_0$, the PRK real stability domain always includes $[-\beta, 1]$ and, for $M = M_0$, $|\sigma_{\mathrm{PRK}}(\widehat{\rho})| = 1$; see Table IIA. Finally, as k becomes large, the limiting value is

$$M_0 = \left[\mathrm{e} + \sqrt{\mathrm{e}\ (2+\mathrm{e})}\right] k \approx 6.2996\, k. \tag{25}$$

This value provides a rule of thumb for predicting when a sufficiently large projective step can result in a loss of $[0,1]$-stability for the outer PRK integrator. In terms of cost, the number of inner integration steps required for the outer PRK step is $2(k+1)$.

4.2 Projective Adams-Bashforth

The formula for second-order accurate projective Adams-Bashforth (PAB) is

$$y_{n+s} = y_{n+k+1} + Mh_0[\alpha v'_{n+k+1,n+k} + (1-\alpha)v'_{n-s_{-1}+k_{-1}+1,n-s_{-1}+k_{-1}}], \tag{26}$$

where the latter v' term is the chord slope from the previous time step (which is assumed to use a k value of k_{-1}, an M value of M_{-1} and $s_{-1} = k_{-1} + 1 + M_{-1}$). The projective Adams-Bashforth stability polynomial is

$$\sigma^2_{\mathrm{PAB}}(\rho) = \left[\rho^{k+1} + \alpha M\ (\rho^{k+1} - \rho^k)\right]\sigma_{\mathrm{PAB}}(\rho) + M(1-\alpha)(\rho^{k_{-1}+1} - \rho^{k_{-1}}). \tag{27}$$

An analysis similar to that for projective Runge-Kutta and its error terms reveals that

$$\begin{aligned}
y_{n+s} =& y(t_{n+k+1}) - \frac{h_0^2}{2}\sum_{i=0}^{k}\xi_{n+i}y'' + Mh_0[y'(t_{n+k+1+M/2}) \\
&+\alpha[y'(t_{n+k+1/2}) - y'(t_{n+k+1+M/2})] + (1-\alpha)[y'(t_{n-s_{-1}+k_{-1}+1/2}) - y'(t_{n+k+1+M/2})]] \\
&-\frac{Mh_0^2}{2}\left[\alpha\xi_{n+k} + (1-\alpha)\xi_{n-s_{-1}+k_{-1}}\right]y'' + \mathcal{O}(h_0^3). \\
=& y(t_{n+k+1}) + Mh_0y'(t_{n+k+1+M/2}) \\
&-\frac{Mh_0^2y''}{2}\left[-2\alpha(M+1+k) + M + 1 + 2(M+1+k) + \frac{1}{M}\xi_n^{\mathrm{PAB}}\right] \\
&+\mathcal{O}(h_0^3)
\end{aligned} \tag{28}$$

where

$$\xi_n^{\mathrm{PAB}} = \sum_{i=0}^{k}\xi_{n+i} + M[\alpha\xi_{n+k} + (1-\alpha)\xi_{n-s_{-1}+k_{-1}}]. \tag{29}$$

The value of α for a second-order accurate method is therefore given by

$$\alpha = 1 + \frac{M + 1 + \frac{1}{M}\xi_n^{\mathrm{PAB}}}{2(M+1+k)} = 1 + \frac{M + 1 + \frac{1}{M}\left[\sum_{i=0}^{k}\xi_{n+i} + M\xi_{n-s_{-1}+k_{-1}}\right]}{2(M+1+k) - \xi_{n+k} + \xi_{n-s_{-1}+k_{-1}}}. \tag{30}$$

As with projective Runge-Kutta, if k and M are constant and the inner integrator is forward Euler, $\xi_n^{\mathrm{PAB}} = s$. The earlier recurrence relation, (22), can be used to compute $\xi_{n,j}$ in each layer of telescoping.

A useful, preliminary analysis of PAB is possible if we assume a common value of k and M at the current and previous time step, and a forward Euler inner integrator. (In implementing a production code, however, we will want to vary k and M at each time step.) For a common

Table III

Comparison of second-order accurate integrators for the same cost:

Projective Runge-Kutta and Projective Adams-Bashforth

	$s = k + 1 + M_0$		β_s	
cost	PRK	PAB	PRK	PAB
4	9.7958	10.4480	0.1137	0.4655
6	17.1501	16.7208	0.3333	0.5652
8	24.4726	22.9933	0.3310	0.6311
10	30.7848	29.2657	0.4847	0.6781

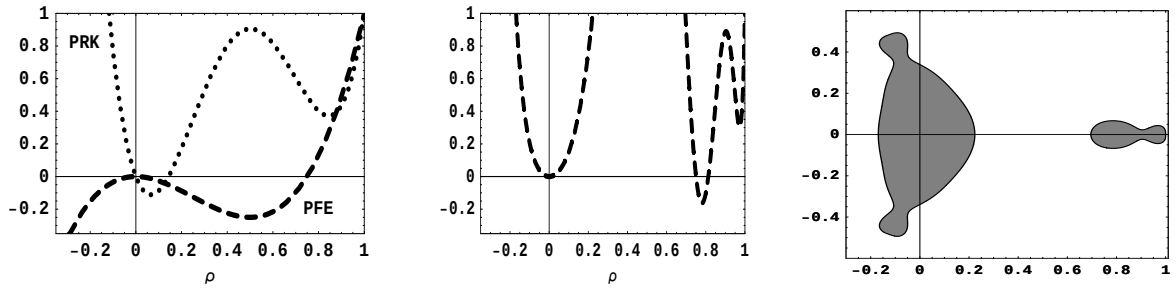
value of k, an examination of the PAB real stability polynomial reveals that as M becomes large, eventually $[0, 1]$-stability is lost because at least one of the roots of eq. (27) gives $|\sigma_{\rm PAB}(\rho)| > 1$ for ρ within $[0, 1]$. Unfortunately, it is not possible to obtain formulas for M_0 and $\widehat{\rho}$ explicitly in terms of k. Table IIB gives critical values for $[0, 1]$-stability for a given value of k. For $0 \leq M < M_0$, the PAB real stability domain always includes $[-\beta, 1]$; for $M = M_0$, $|\sigma_{\rm PAB}(\widehat{\rho})| = 1$. Empirically, by observing M_0 for increasingly large k, we find that the limiting value is

$$M_0 \approx 2.14\, k. \tag{31}$$

With regard to cost, the number of inner integration steps required for the outer PAB step is $k + 1$.

5 Discussion

In making a fair comparison between these two projective integrators, recall that one PRK and PAB outer step costs $2(k + 1)$ and $k + 1$ inner integrator steps, respectively. For the same cost, we consider the largest possible $[0, 1]$-stable outer step s and its corresponding real stability domain $[-\beta_s, 1]$. The main results are given in Table III. Note that as the cost increases, that PRK gives s values that are slightly larger than for PAB. The potential advantage of a larger outer step s is somewhat offset by the fact that β_s is smaller for PRK than for PAB. As we demonstrate in §5, a $[0, 1]$-stable outer integrator with a small β_s is not desirable because an inner integrator with a large real stability range (i.e., γ-range) may result in a combined outer-inner integrator that is not $[0, 1]$-stable.



(a) $\sigma_{\mathrm{PRK}}(\rho)$, $\sigma_{\mathrm{PFE}}(\rho)$. (b) $\sigma_{\mathrm{PRK}}(\sigma_{\mathrm{PFE}}(\rho))$. (c) $\sigma_{\mathrm{PRK}}(\sigma_{\mathrm{PFE}}(\rho))$.

Fig. 7. Stability plots for projective Runge-Kutta (PRK, $k = 1, M = 7.5$), projective forward Euler (PFE, $k = 2, M = 3$), and outer PRK with PFE as an inner integrator.

6 Stability analysis for higher-order accurate teleprojective integrators

Our approach for combining higher-order accurate outer projective integrators with layers of projective forward Euler is as follows. For stabilizing and developing the stiff system at fast and intermediate time scales, the accuracy of first-order projective integrators is sufficient. In particular, we intend to use $[0, 1]$-stable PFE and increase the number of layers until accuracy requirements become the important consideration for the telescopic step. The same k and M values are applied at each layer. For following the stiff system accurately at the slow time scales, a higher-order accurate projective integrator is added as the final layer.

Potential problems with the $[0, 1]$-stability of this approach can be demonstrated with the simple example in Figure 7. The example involves using $[0, 1]$-stable PRK ($k = 1, M = 7.5$) as an outer integrator and $[0, 1]$-stable PFE ($k = 2, M = 3, L = 1$) as the inner integrator. (Telescoping is not being used in this example.) Plots of their respective real stability polynomials are given in Figure 7(a). By inspection, each method is $[0, 1]$-stable because $|\sigma| \leq 1$ for all ρ within $[0, 1]$. However, Figure 7(b) reveals that the combined method, PRK with PFE as an inner integrator, is not $[0, 1]$-stable. A peak value of 4.10 (not shown) is reached at $\rho = 0.5$. Figure 7(c) shows the stability region of the combined method as plotted in the complex ρ-plane. It turns out that $[0, 1]$-stability can be restored (i.e., no gap for $\rho \in [0, 1]$) by reducing the outer PRK value of M from 7.5 to 3 or less. But this reduction of M is an unsatisfactory and inefficient option that we generally want to avoid.

What causes this combined outer-inner integrator not to be $[0, 1]$-stable? The problem is revealed by the diagrammed equation

$$\sigma_{\mathrm{outer}}(\underbrace{\sigma_{\mathrm{inner}}(\overbrace{\rho}^{0.5\in[0,1]})}_{\sigma=-0.25}) = 4.10 > 1. \tag{32}$$

The violation occurs for other real values in the neighborhood of $\rho = 0.5$. Briefly put, the trouble is that the γ-range for the inner PFE integrator is wider than the real stability domain for the outer PRK integrator. As can be observed from Figure 7(a), the PFE range

is $[-0.25, 1]$ and the domain for PRK stability is only $[-0.118, 1]$. This mismatch is the cause of lost $[0, 1]$-stability when combining the two integrators. The gap along the real axis is due to the values of ρ that give

$$-0.25 \leq \sigma_{\mathrm{PFE}}(\rho) \leq -0.118,$$

because those lead directly to values with $|\sigma_{\mathrm{PRK}}(\sigma_{\mathrm{PFE}}(\rho))| > 1$.

The remedy is for the mismatch to be the other way around in order for a composition of $[0, 1]$-stable integrators to yield a combined integrator that is also $[0, 1]$-stable. That is, the γ-range of the inner integrator needs to be less than or equal to the real stability β_s-domain of the outer integrator. In general, a potential difficulty is that γ for the inner integrator may not be readily available or easy to determine. Fortunately, for $[0, 1]$-stable telescopic PFE with k damping steps, Table IB gives the γ-range least upper-bound $[-\gamma, 1] \subseteq [-\beta, 1]$ at each layer and consequently for the telescopic step. For second-order accurate methods and a given number of damping steps, Table II gives the greatest lower bound $[-\beta, 1] \subseteq [-\beta_s, 1]$ on the real stability domain for $[0, 1]$-stable projective Runge-Kutta and Adams-Bashforth outer integrators. The main result is that the combined outer-inner integrator is $[0, 1]$-stable if the condition

$$[-\gamma, 1]_{\mathrm{inner}} \subseteq \underbrace{[-\beta, 1]_{\mathrm{inner}} \subseteq [-\beta, 1]_{\mathrm{outer}}} \subseteq [-\beta_s, 1]_{\mathrm{outer}} \tag{33}$$

is satisfied by the inner and outer β values (see underbrace). The latter values are determined by the number of damping steps used by the $[0, 1]$-stable inner and outer integrators, respectively, and (as noted) are readily available from Tables IB and II.

7 Test problems

We have developed a research code in `Mathematica` that implements a preliminary version of these second-order accurate teleprojective integrators. We now examine several significant features of the integrators, comment on their efficiency in comparison to implicit solvers for stiff initial value problems, and give numerical results for some simple test problems.

7.1 Order of accuracy

The purpose of this study is to assess numerically the order of accuracy achieved by the projective Runge-Kutta (PRK) and Adams-Bashforth (PAB) integrators in §4 as the outer step changes. (Our analysis that showed second order was in terms of the inner step size that, in practice, is fixed by the need to damp the fastest component.) The simplest possible

case to consider is a single, scalar ODE test problem. For the behavior of the true error, we assume

$$|\text{error}| \approx CH^p \quad \text{so that} \quad \log(|\text{error}|) \approx \log(C) + p\ \log(H). \tag{34}$$

The error behavior can then be parameterized from a least-squares fit of the logarithmically-scaled data for true error versus a constant outer integrator step size H: the error constant is C, and the slope p is the order of accuracy.

A stiff test problem is not suitable for these studies because local errors are strongly damped over time. Instead, a nonstiff, nonlinear autonomous problem is integrated over a time interval for which the solution is not quite at its steady state value. The scalar initial value problem is:

$$y'(t) = \frac{(y(t) - 20001)(y(t) - 1)}{20000}, \qquad y(0) = 10001, \tag{35}$$

for $t \in [0, 15]$. The exact solution and the final value are:

$$y(t) = 1 + \frac{20000}{1 + e^t}, \qquad y(15) = 1.00612. \tag{36}$$

From $t = 0$ to $t = 15$, the initial value rapidly drops four orders of magnitude as it approaches its steady state value of 1.

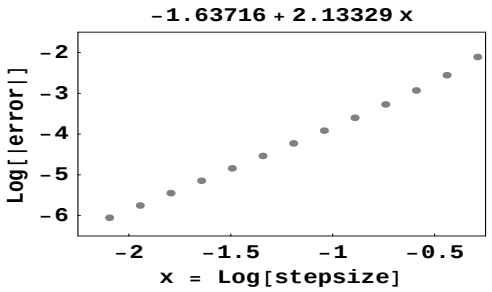
The order of accuracy is estimated on the following basis. The inner integrator is telescopic PFE; the outer integrators are PRK or PAB. For PFE, $k = 2$, $M = 3$, and $L = 3$ (with $s = k + 1 + M = 6$). For the innermost step size, we choose $h_0 = 10^{-8}$ which means each inner damping step is of length $s^L h_0 = 2.16 \times 10^{-6}$. If we require that the outer integrators are $[0, 1]$-stable, the maximum $[0, 1]$-stable outer step size can be estimated as

$$H = (k + 1 + M_0)\ s^L h_0 \tag{37}$$

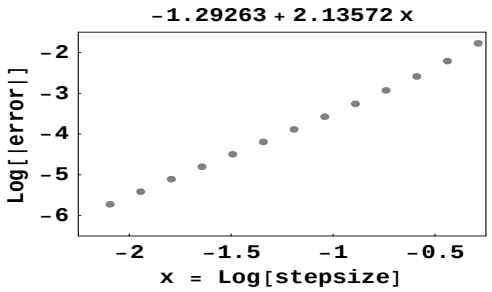
where M_0 depends on the value of k used by the outer integrator. For outer PRK and PAB with $k = 2$, we get

$$H = (17.1501)\ 2.16 \times 10^{-6} = 3.70 \times 10^{-5} \quad \text{and}$$
$$H = (7.3115)\ 2.16 \times 10^{-6} = 1.58 \times 10^{-5}$$

respectively. But $[0, 1]$-stability is not required for this one-scale example problem, so from (37) we find that large H with $M \gg M_0$ can be used in resolving the slow scale behavior since there are no fast and intermediate scales to cause instability. The outer step sizes we consider start at $H = 0.008$, and increase by a factor of $\sqrt{2}$ to $H = 0.512$. Figure 8 shows the observed order of accuracy for the outer step sizes as based on the least-squares slope. The PRK and



(a) Projective Runge-Kutta ($k = 2$)



(b) Projective Adams-Bashforth ($k = 2$)

Fig. 8. Order of accuracy estimated as the slope of a least-squares fit of error versus outer step size. PAB slopes are nearly identical, $p = 2.13$, but the error constant C is smaller for PRK than for PAB.

7.2 Local error estimation and adaptive outer step size selection

In general, the overall order of accuracy of PRK and PAB is approximately 2. But locally this estimate can be off the mark if, for example, the error in an inner integrator has the opposite sign of the error in the outer integrator. Since the local error is some combination of the two, there can be cancellation. Nonetheless, the assumption of second-order accurate behavior is useful for the purposes of local error estimation and adaptive outer step size selection.

Our research code currently carries out local error estimation via Richardson extrapolation. A solution y_1 is computed based on an outer step size H_n; then, the solution y_2 is computed based on two outer steps of size $H_n/2$. The local error is estimated as

$$\text{error}_n = \frac{y_2 - y_1}{2^p - 1} \tag{38}$$

where p is the order of accuracy of the integrator. Naturally, for outer PRK and outer PAB we choose $p = 2$. Outer step size selection or rejection is then a function of the estimated local error, current step size, and user-specified error tolerances (absolute and relative) so that

$$H_{n+1} = H_n \left(\frac{C_n}{||\text{error}_n||} \right)^{1/3} \tag{39}$$

for a positive method-dependent constant C_n and a weighted, tolerance-dependent norm of the estimated local error. If the new step size increases or decreases significantly, the number of layers of telescoping may need to be adjusted to accommodate the change. Such adaptivity for inner telescoping is the subject of ongoing research. Currently, we assume the inner telescopic PFE parameters remain fixed. The outer integrator uses a constant

number of damping steps k, but varies the projective step M based on the next selected outer step size. Finally, we note that local error estimation via Richardson extrapolation is expensive due to the overhead of two additional integrations in computing y_2. It is likely that better approaches can be developed based on predictor-corrector or embedded Runge-Kutta techniques [16].

7.3 Cost analysis with respect to implicit integrators

We now present a simplified cost analysis for solving large-scale stiff initial value problems. The derivation is similar to the approach used for implicit and (stabilized) explicit methods in [3,18]. The main results assume that we are solving a spatially-discretized parabolic partial differential equation (PDE) with a uniform mesh width of Δ in each of the d spatial directions. A typical example is a d-dimensional diffusion equation with a time-dependent source term and Dirichlet boundary conditions within a d-dimensional region of unit length in each spatial direction. The total number of equations is approximately Δ^{-d} and we can estimate the most negative, real eigenvalue of the sparse Jacobian as $-4d\Delta^{-2}$. Note that the step size for conventional forward Euler stability is therefore $h_0 \approx 1/(4d\Delta^{-2})$.

For implicit integrators such as those based on backward differentiation formulas (BDFs), the BDF time-discretized equations are usually solved using one or more Newton iterations at each time step. The update to each Newton iterate is obtained by solving a large-scale linear system of equations. For one-dimensional PDEs, these linear systems can generally be solved efficiently with Gaussian elimination or a direct method for band-structured matrices. With two- and three-dimensional PDEs, we assume an inexact preconditioned Newton-Krylov method is used. For a preconditioned Krylov method, we estimate that $\mathcal{O}(\Delta^{-1/2})$ iterations are needed for approximately solving the linear system to the required accuracy within the BDF integrator. The total number of function evaluations for one time step is then $N_{\mathrm{f}} \approx N_L\ \mathcal{O}(\Delta^{-1/2})$, where N_L is the number of linear systems solved. Because the amount of work performed in a function evaluation is proportional to the number of equations $\mathcal{O}(\Delta^{-d})$, the total computational work for integrating a unit time interval with step size H is

$$W_{\mathrm{implicit}} \approx H^{-1} N_L\ \mathcal{O}(\Delta^{-(d+1/2)}). \tag{40}$$

A similar cost analysis can be carried out for the second-order accurate projective integrators. The total number of function evaluations for a $[0,1]$-stable outer PRK step comes from the $2(k_{\mathrm{PRK}}+1)$ damping steps, each of which requires $(k+1)^L$ function evaluations by the inner telescopic PFE integrator. The total number of function evaluations for one outer PRK step is $N_{\mathrm{f}} = 2(k_{\mathrm{PRK}}+1)(k+1)^L$, where k_{PRK} is sufficiently large so that the outermost projective step culminates in a step size H of acceptable accuracy. The total computational work for integrating the unit time interval with step size H is

$$W_{\mathrm{PRK}} \approx 2H^{-1}(k_{\mathrm{PRK}}+1)(k+1)^L\mathcal{O}(\Delta^{-d}) \tag{41}$$

and for projective Adams-Bashforth

$$W_{\mathrm{PAB}} \approx H^{-1}(k_{\mathrm{PAB}}+1)(k+1)^L \mathcal{O}(\Delta^{-d}). \tag{42}$$

Ideally the implicit and projective integrators tend to select a common step size H that is based on the accuracy requested for the initial value problem. In that case, the integrators are equally expensive when the cost of the linear algebra associated with the implicit step is comparable to the cost of executing the outer PRK (or PAB) step. The latter cost can be predicted reasonably well, and is usually dominated by the $(k+1)^L$ evaluations used to bridge the time scales from h_0 to within (roughly) an order of magnitude of H. Projective integrators are also highly scalable if the function evaluations are easily parallelized. For implicit integrators, the difficulty of dealing with the linear algebra bottleneck is highly problem-dependent. For complicated multiscale problems, it may be problematic to implement robust linear solvers or efficient preconditioning techniques—especially ones that scale well on massively parallel machines. A good preconditioner is also needed for the linear (and nonlinear) convergence tests to work properly. Furthermore, without a good preconditioner, the desired implicit step size may need to be shortened if the resulting linear systems cannot be solved efficiently. Nonetheless, implicit methods are still the most commonly used class of integrators for stiff initial value problems. For large-scale applications, an excellent survey of preconditioning techniques for Jacobian-free Newton-Krylov methods for implicit integrators is given in [11].

7.4 Numerical results

We now demonstrate and compare the performance of projective and implicit integrators on a spatially-discretized parabolic PDE problem. The test problem is the two-dimensional diffusion equation

$$u_t = u_{xx} + u_{yy} + g(x,y,t). \tag{43}$$

The space and time intervals are $x, y \in [0,1]$ and $t \in [0, 1.5]$. The source term $g(x,y,t)$ is chosen so that the exact PDE solution is

$$u(x,y,t) = 1/\left[1+\exp(8(x+y-t))\right], \tag{44}$$

which is used for properly specifying the initial condition and Dirichlet boundary conditions. The spatial discretization uses second-order centered finite differences with a mesh width of $\Delta = 1/(n+1)$ in each spatial direction, which provides $N = n^2$ interior gridpoints and unknowns within the unit square. The absolute and relative error tolerances are both set to 10^{-3}. For the innermost step size, $h_0 = 1/(8\Delta^{-2})$.

TABLE IV

Solver statistics for projective integrators: 2D diffusion PDE

	A				B			
	Projective Runge-Kutta ($k = 3$)				Projective Adams-Bashforth ($k = 3$)			
N	L	f-evals	steps	error	L	f-evals	steps	error
100	1	1,325	27 (0)	9.6e-5	1	651	25 (0)	4.9e-4
400	2	2,524	26 (0)	7.6e-5	2	1,226	24 (0)	4.6e-4
1,600	3	4,827	25 (0)	2.9e-4	3	2,426	24 (0)	6.1e-4
6,400	4	9,627	25 (0)	2.4e-4	4	4,826	24 (0)	7.0e-4

The outer projective integrators PRK ($k = 3$) and PAB ($k = 3$) are used with telescopic PFE ($k = 1$, $M = 2$) as the inner integrator. The number of inner telescoping layers L is chosen large enough so that the adaptive outer integrator step size is based on accuracy considerations. Recall that Richardson extrapolation (38) is used for local error estimation. For problems of different sizes N, Table IV shows the number of function evaluations (f-evals), number of accepted and (rejected) steps, and the maximum absolute error in a solution component. The error is the time integration error, which is the difference between the spatially-discretized numerical solution and a reference solution of the ODEs as computed with a stringent tolerance.

Table IV gives the numerical results for solving the test problem using projective integrators. For each problem size, projective Runge-Kutta uses about twice as many function evaluations as projective Adams-Bashforth, but the PRK error is at least twice as small as that for PAB. The number of accepted and (rejected) steps are nearly identical. The solver statistics roughly demonstrate that PRK can be twice as costly as PAB, but that the smaller error constant for PRK results in less solution error. Finally, note that when the problem size N quadruples, the total number of function evaluations only doubles. The reason for this is that the innermost step size h_0 is reduced by (nearly) four, but an inner layer of telescoping is added to bridge the now wider range between the fastest time scale and the slow time scales of interest. Applying an extra layer of telescoping increases the inner PFE integrator step size by $k + 1 + M = 4$, but only increases its cost by a factor of $k + 1 = 2$ since $k = 1$ and $M = 2$ for the additional inner layer of PFE.

We now compare the projective integrator results with the implicit BDF method that is implemented in `CVODE` [10]. It is a variable step size, variable order BDF integrator and, for fair comparison, we restrict its maximum order of accuracy to two. For the linear system of equations that arise at each time step, we investigate two options: the use of a direct, banded matrix solver with a finite-difference generated Jacobian; and, the use of the Jacobian-free form of GMRES [15] with a simple preconditioner. The results highlight the importance of efficiently dealing with the linear algebra workload generally associated with implicit methods. One property to exploit is the structure of the Jacobian. It is a pentadiagonal, band matrix with a half-bandwidth equal to $\sqrt{N}$. Cost-effective techniques include using finite-

TABLE V

Solver statistics for implicit BDF integrator: 2D diffusion PDE

	A				B				
	CVODE/banded matrix solver				CVODE/preconditioned GMRES				
N	f-evals	steps	Newton	error	f-evals	steps	Newton	GMRES	error
100	61	34 (0)	37	3.3e-4	1,040	40 (0)	45	988	6.7e-3
400	79	32 (0)	35	3.8e-4	2,828	38 (0)	43	2,778	3.6e-3
1,600	127	38 (0)	43	5.6e-4	7,063	63 (1)	72	6,981	9.6e-4
6,400	199	32 (0)	35	2.7e-4	8,752	44 (0)	50	8,692	1.6e-3

differencing in which several columns of the banded Jacobian are estimated simultaneously with one function evaluation. Another technique is to re-use the factored Jacobian (or preconditioner) over as many time steps as possible. For a preconditioner, a CVODE routine for approximating the tridiagonal portion of the Jacobian is used. Furthermore, we allow GMRES to use as many iterations as needed to solve the preconditioned system of linear equations so that CVODE can attempt to take the step size that it wants. For the other parameter values in CVODE, default values are used.

Table V shows the results of using a second-order accurate implicit solver when the previously mentioned direct and preconditioned iterative linear equation solvers are applied. As with the projective integrator results in Table IV, roughly 40 steps are needed to integrate the problem from time $t = 0$ to $t = 1.5$. Only in one case is a step size rejected (CVODE/GMRES for $N = 1,600$), and that surprisingly leads to 63 steps and thousands of extra function evaluations. A comparison between Tables IV and V also indicates that the integrators achieve roughly the same accuracy (about 1.0e-3). The cost of projective integrators relative to implicit integrators will depend mainly on how expensive (both in terms of work and storage) it is to solve the linear systems. With the banded matrix solver, Table VA shows that the linear systems can be solved cheaply—especially since CVODE only computes the Jacobian once and re-uses it across all the time steps. With GMRES, the preconditioner is computed once or twice but the main cost is from executing the GMRES iterations. Note that the average number of GMRES iterations per Newton iteration increases as the problem size increases, which indicates that the preconditioner is less effective as the problem becomes more stiff.

At first glance, it is reasonable to view these results as a firm example for which projective integrators and CVODE/preconditioned GMRES give comparable step size selection and solution accuracy; though, in terms of cost, projective Adams-Bashforth is preferable. It is important to note that the cost of the implicit CVODE/GMRES integrator can be greatly reduced with a judicious choice of preconditioner—preferably with the $\mathcal{O}(\Delta^{-1/2})$ cost as estimated in (40). (But for more general, complicated or realistic problems, we cannot count on easily coming up with such an effective preconditioner.) Similarly, the cost of projective integration can be reduced by about a factor of three if more efficient local error estimation

techniques (other than Richardson extrapolation) can be developed. Other improvements may come from developing projective versions of other conventional methods besides explicit Runge-Kutta and Adams-Bashforth. In summary, we expect that for applications for which both implicit and projective integrators can be used, there will be a distinct range of problems and/or computer architectures that are exclusively favorable to each approach.

8 Conclusions and perspective

The mathematical analysis, algorithmic development and numerical results from the previous sections provide a foundation for an unconventional, fully explicit approach to solving multiscale problems. The key innovation amounts to using an inner layer of first-order accurate explicit integrators to develop and stabilize the stiff system at the fast and intermediate time scales. A second-order accurate outer integrator can then be used to take time steps commensurate with the slow time scales and accuracy requirements for the multiscale problem.

We have applied the new, second-order accurate projective integrators to stiff initial value problems, and demonstrated step size selection and solution accuracy comparable to conventional implicit integrators for a two-dimensional parabolic PDE. Yet projective integrators potentially have a much wider range of applicability for large-scale scientific computing and multiscale modeling [12,13]. In developing capabilities (e.g., local error estimation, adaptive step size selection, stiffness detection) on par with implicit integrators for stiff initial value problems, these features in projective integrators readily carry over to handling stiff systems for which implicit integrators are not applicable.

Acknowledgments

The authors gratefully acknowledge the support of the National Nuclear Security Administration's Advanced Simulation and Computing (NNSA/ASC) program at Lawrence Livermore National Laboratory. This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory, under contract No. W-7405-ENG-48.

References

[1] A. Abdulle, Fourth order Chebyshev methods with recurrence relation, SIAM J. Sci. Comput. 23 (6) (2002) 2041–2054.

[2] A. Abdulle, A. A. Medovikov, Second order Chebyshev methods based on orthogonal polynomials, Numer. Math. 90 (2001) 1–18.

[3] T. F. Chan, K. R. Jackson, The use of iterative linear-equation solvers in codes for large systems of stiff IVPs for ODEs, SIAM J. Sci. Comput. 7 (2) (1986) 378–417.

[4] W. E, Analysis of the heterogeneous multiscale method for ordinary differential equations, Comm. Math. Sci. 1 (3) (2003) 423–436.

[5] W. E, B. Engquist, The heterogeneous multiscale methods, Comm. Math. Sci. 1 (1) (2003) 87–132.

[6] B. Engquist, Y. H. Tsai, Heterogeneous multiscale methods for stiff ordinary differential equations, Mathematics of Computation 74 (252) (2005) 1707–1742.

[7] K. Eriksson, C. Johnson, A. Logg, Explicit time-stepping for stiff ODEs, SIAM J. Sci. Comput. 25 (4) (2003) 1142–1157.

[8] C. W. Gear, I. G. Kevrekidis, Projective methods for stiff differential equations: Problems with gaps in their eigenvalue spectrum, SIAM J. Sci. Comput. 24 (4) (2003) 1091–1106.

[9] C. W. Gear, I. G. Kevrekidis, Telescopic projective methods for parabolic differential equations, J. Comput. Phys. 187 (1) (2003) 95–109.

[10] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Softw. 31 (3) (2005) 363–396, Special issue on the Advanced Computational Software (ACTS) Collection.

[11] D. A. Knoll, D. E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.

[12] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level tasks, Comm. Math. Sci. 1 (4) (2003) 715–762.

[13] I. G. Kevrekidis, C. W. Gear, G. Hummer, Equation-free: The computer-aided analysis of complex multiscale systems, AICHE Journal 50 (7) (2004) 1346–1355.

[14] A. A. Medovikov, High order explicit methods for parabolic equations, BIT 38 (2) (1998) 372–390.

[15] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Comput. 7 (3) (1986) 856–869.

[16] M. Sofroniou, G. Spaletta, Construction of explicit Runge-Kutta pairs with stiffness detection, Mathematical and Computer Modelling 40 (11) (2004) 1157–1169.

[17] B. P. Sommeijer, L. F. Shampine, J. G. Verwer, RKC: An explicit solver for parabolic PDEs, J. Comput. Appl. Math. 88 (2) (1998) 315–326.

[18] P. J. van der Houwen, The development of Runge-Kutta methods for partial differential equations, Appl. Num. Math. 20 (3) (1996) 261–272.