LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-224892

On-the-Fly Local Error Estimation for Projective Integrators

S. L. Lee, C. W. Gear

October 2, 2006

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

ON-THE-FLY LOCAL ERROR ESTIMATION FOR PROJECTIVE INTEGRATORS

STEVEN L. LEE* AND C. WILLIAM GEAR†

Abstract. A novel and efficient technique is developed for estimating the local error per step when first- and second-order accurate projective integrators are applied to stiff multiscale systems. The estimation can be done on-the-fly; that is, the accumulated local error is readily estimated at the end and during the course of computing the solution at each outer time step. We demonstrate the effectiveness of the new error estimation technique when using projective integrators to solve stiff ordinary differential equation (ODE) initial-value problems.

Key words. Local error estimation, stiffness, explicit, multiscale, teleprojective integration, on-the-fly

AMS subject classifications. 65L05, 65L06, 65L20

1. Introduction. In a sequence of recent papers [2, 3, 7], first- and second-order accurate projective integrators for stiff multiscale problems were introduced and developed. In the latter, we focused on second-order accurate projective Runge-Kutta (PRK) and Adams-Bashforth (PAB) as outer integrators for a telescopic projective forward Euler (PFE) inner integrator. At each outer step, the combined outer-inner integrator parameters (outer step size, number of inner damping steps, number of telescoping layers, length of outer projective step) may change, and it does not seem possible to derive a simple technique for estimating the local error incurred at each outer step. In [7], we merely estimated the local error (per outer step) via Richardson extrapolation—a general technique with an overhead expense that often increases the amount of computation by at least a factor of two.

In this paper, we derive recurrence formulas that enable the calculation of local error coefficients as the inner integrator steps proceed (i.e., on-the-fly). The local error at each outer integrator step can then be calculated once the appropriate solution derivatives have been estimated. While it may appear that these recurrences are complicated, it must be remembered that we will mainly be considering projective integrators for stiff systems for which it may be problematic to use fully implicit integrators (e.g., backward differentiation formulas (BDFs)). The latter is often used in concert with preconditioned Newton-Krylov methods for solving the nonlinear systems that arise at each time step (e.g., `CVODE` [4]). Unfortunately, for petascale systems, robust Jacobian preconditioning techniques may be difficult to devise or may not be highly scalable on massively parallel machines. Alternatively, if the stiff differential equations are not directly available, projective integrators can also be applied as a wrapper for accelerating a black-box time-stepper code for the stiff system. Finally, projective integration is a cornerstone for enabling the equation-free (EF) and heterogeneous multiscale method (HMM) approaches to solving multiscale problems [5, 1].

The remainder of this paper is organized as follows. In the next section, we briefly review projective integration methods. In Section 3, we develop a novel local error estimation technique for first- and second-order accurate projective integrators. In Section 4, we apply the adaptive projective integrators (with local error estimation and simple step size control) to stiff ODE test problems. Finally, in Section 5, we state our conclusions and give a perspective on the significance of these new results.

2. Projective integration methods. The key computational challenge for stiff multiscale systems is the problem of trying to resolve efficiently the slow time scale behavior while incorporating the effects from the fast and intermediate time scales. Projective integrators are explicit methods that can be tailored to efficiently exploit the multiscale features that are characteristic of stiff systems. Furthermore, projective integration is a process that can be applied to a legacy time-stepper code, the output from a microscopic simulation of a more detailed model, or any type of step-by-step inner integrator. A projective integration

*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory (LLNL), P.O. Box 808, L-560, 7000 East Avenue, Livermore, California, 94551 (slee@llnl.gov).

†Professor Emeritus, University of Illinois at Urbana-Champaign; Senior Scientist, Princeton University; and, LLNL Visiting Researcher. 17 Honeybrook Drive, Princeton, NJ 08540 (wgear@princeton.edu).

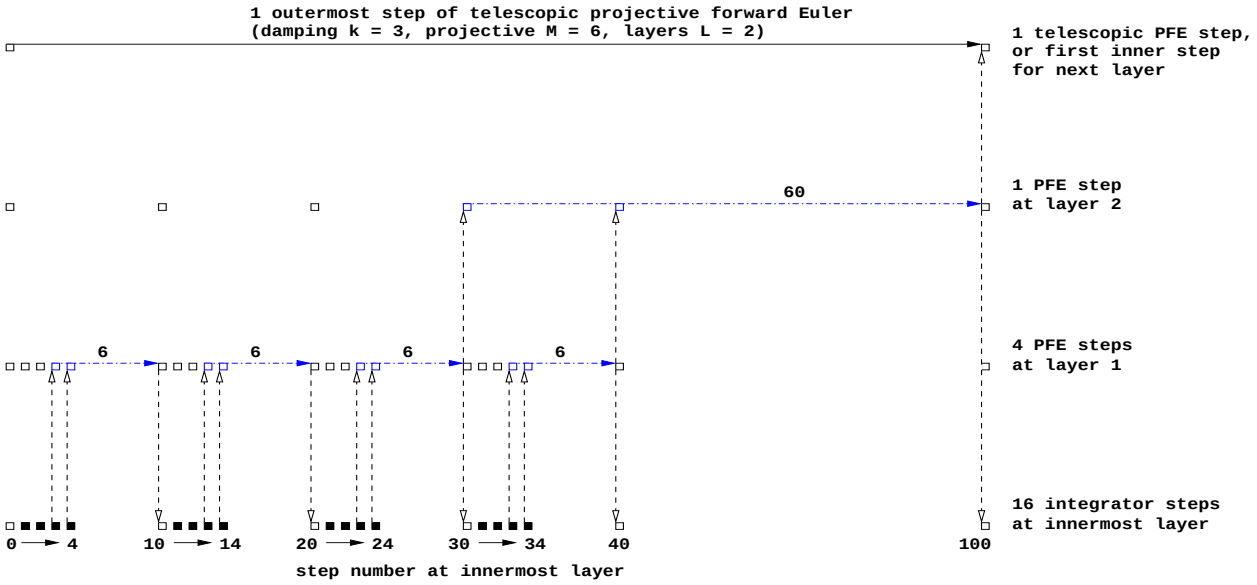

FIG. 2.1. *Projective forward Euler with $L = 2$ layers of telescoping.*

for advancing the solution y by $s = k + 1 + M$ inner integrator steps has the form: Starting from y_0, take $k + 1$ inner integrator steps (each of size h_0) and then a chord slope-based step so that

$$y_s = y_{k+1} + M(y_{k+1} - y_k).$$

The combination of the $k + 1$ inner damping steps and the projective step constitutes one outer projective forward Euler (PFE) step of size sh_0. PFE can be applied efficiently to stiff systems that have a wide separation between the fast and slow time scales. An analysis of PFE is provided in [2].

For stiff systems with no clear separation of time scales, telescopic projective integrators are needed so that the system evolves stably for the fast and intermediate time scales. In brief, the teleprojective integration process involves using one projective integrator (e.g., PFE) as an inner integrator within yet another outer projective integrator [3]. This layering of projective integrators can be repeated as many times as desired. Figure 2.1 shows an example of PFE using $L = 2$ layers of telescoping. A constant step size integrator (e.g., conventional forward Euler) is used at layer 0. PFE is used at layers 1 and 2, with the damping parameter $k = 3$ and projective step multiplier $M = 6$ at each layer. The telescopic step advances by 100 forward Euler steps at the cost of only 16 function evaluations. This improved efficiency, which increases as more layers are added, stems from the use of chord slopes in taking projective steps of size 6 and 60 at layers 1 and 2, respectively. The actual number of telescoping PFE layers to apply will depend on the accuracy requirements for the stiff system. For the final layer, when the slow time scales need to be resolved for accuracy, we can use an integrator that finishes that final outer step with second-order accuracy (e.g., projective Runge-Kutta or projective Adams-Bashforth). The stability and accuracy of such combined outer-inner integrators partly depends on a suitable choice of parameter values k, M and L for the stiff system; these matters are addressed in [7]. By equipping the overall telescoping approach with better local error estimation capabilities, this new class of fully explicit integrators will be able to take (with greater efficiency) outer steps commensurate with the slow time scales and accuracy requirements for the stiff multiscale problem.

3. Local Error Estimation. The *local error* of an integration method is the error introduced in a single step of the method *starting from correct values*. The *global error* can then be expressed as a sum of

local errors, each amplified (or damped) by the effect of a combination of the differential equation and its difference approximation. (This is somewhat of a simplified view, but will suffice for the discussion in this paper.) Step size selection can then be used to control the local error—and typically one controls the local error either on a per step basis or a per unit step basis, the former being more appropriate to stiff equations.

In the context of telescopic projective integration methods, this concept of step size control needs to be refined. At the inner steps we are not controlling the step size for accuracy, we are using small steps to gain stability. Only at the outermost layer do we control the step size for accuracy. Hence, we need to define the local error for the outermost step. It will be defined as the error due to one outermost step, assuming that we started with the correct value. The outermost step consists of a series of integration steps at the next lower layer using an *inner integrator* (whose steps themselves may be projective steps built on top of even lower layers), followed by the projective process which extrapolates forward. Hence, to find the local error at the outermost layer, we need to find the errors in each of these inner steps and the error due to the projective step. Thus, we will need to determine the local error in integration steps at every layer.

Let us start by considering a single projective forward Euler (PFE) step that starts from a correct $y_0 = y(t_0)$ and integrates through y_1 and on up to y_{k+1} using step size h before taking a projective step to y_s where $s = k + 1 + M$. Let us also suppose that we know the local error in the inner integrator. Since we are only going to discuss up through third-order error terms in this paper, we will ignore all terms of order four and above from now on. Furthermore, all formulas are of order at least one, so the only terms we will consider in the errors involve multiples of $h^2 y''$, $h^3 y'''$ and $h^3 J y''$ where J is the system Jacobian. In local errors over a projective step, we are considering only a bounded number of steps (independent of h) so that we are not concerned with the exact time at which y''' and Jy'' are evaluated since changing it only adds a fourth-order term that we are ignoring. We are, however, concerned where y'' is evaluated. Hence, we will define

$$\Upsilon_j(h) = \left[-\frac{h^2}{2} y_j'', \ -\frac{h^3}{6} y''', \ -\frac{h^3}{2} J y'' \right]^T$$

and write the local error in the inner step in integrating from y_j to y_{j+1} as

$$d_j = y_{j+1} - y(t_{j+1})$$

where

$$d_j(h) = [\xi_j, \ \gamma_j, \ \eta_j] \ \Upsilon_{j+1}(h) \equiv D_j \Upsilon_{j+1}(h).$$

Note that D_j (and E_j introduced below) are both row vectors consisting of the coefficients of the local (global) error terms. Also, note that the index on y'' is $j + 1$. We are not, at the moment, specifying the current layer of telescoping.

We are now interested in computing the local error in one step of the next higher layer of telescoping. Thus we start with a correct y_0 and compute the error in each succeeding step up to the projective step. Let the *global error* in y_j starting from a correct y_0 be $e_j(h)$; that is,

$$e_j(h) = y_j - y(t_j)$$

and $e_0 = 0$. Let us express $e_j(h)$ as

$$e_j(h) = [\psi_j, \ \phi_j, \ \theta_j] \ \Upsilon_j(h) \equiv E_j \ \Upsilon_j(h). \tag{3.1}$$

Now we wish to express $e_{j+1}(h)$ in terms of $e_j(h)$ and $d_j(h)$. We have

$$e_{j+1}(h) = (I + hJ)e_j(h) + d_j(h) = (I + hJ)E_j \Upsilon_j(h) + D_j \Upsilon_{j+1}(h), \tag{3.2}$$

where $(I + hJ)$ is a first-order approximation to e^{hJ}—which itself is a first-order approximation to the amplification of the error due to the differential equation. Since the error at the previous step is second-order, multiplying it by a first-order approximation is correct to third-order terms. We would like to evaluate the $\Upsilon_j(h)$ at t_{j+1}, which can be done by using

$$\Upsilon_i(h) = T(j-i)\Upsilon_j(h) \tag{3.3}$$

where the translation operator T is

$$T(q) = \begin{bmatrix} 1 & -3q & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

From eq. (3.1), eq. (3.2) and eq. (3.3), we immediately obtain

$$\begin{aligned} \psi_{j+1} &= \psi_j + \xi_j, \\ \phi_{j+1} &= \phi_j + \gamma_j - 3\psi_j, \\ \theta_{j+1} &= \theta_j + \eta_j + \psi_j. \end{aligned}$$

If ξ_j, γ_j and η_j are independent of j (which will normally be true for the innermost integrator and is true for other inner layers if the parameters, k and s, of the projective step do not change), then we have

$$\psi_j = j\xi, \qquad \phi_j = j\gamma - 3j(j-1)\xi/2, \qquad \theta_j = j\eta + j(j-1)\xi/2.$$

When we make the projective step from $\{t_k, t_{k+1}\}$ to t_s using

$$y_s = (M+1)y_{k+1} - My_k,$$

we will get

$$e_s(h) = (M+1)e_{k+1}(h) - Me_k(h) + d_k^{\mathrm{PFE}}(h) \tag{3.4}$$

where d_k^{PFE} is the *local error* in the linear extrapolation from $\{t_k, t_{k+1}\}$, and is given by

$$d_k^{\mathrm{PFE}}(h) = [M(M+1),\ M(M^2-1),\ 0]\ \Upsilon_{k+1}(h). \tag{3.5}$$

Substituting eq. (3.1) and eq. (3.5) into eq. (3.4), we get

$$\begin{aligned} e_s(h) &= E_s\Upsilon_s(h) \\ &= (M+1)E_{k+1}\Upsilon_{k+1}(h) - ME_k\Upsilon_k(h) + [M(M+1),\ M(M^2-1),\ 0]\ \Upsilon_{k+1}(h) \\ &= \left[((M+1)E_{k+1} + [M(M+1),\ M(M^2-1),\ 0])T(M) - ME_kT(M+1)\right]\ \Upsilon_s(h) \end{aligned}$$

or

$$E_s = \left((M+1)E_{k+1} + [M(M+1),\ M(M^2-1),\ 0]\right)\ T(M) - ME_k\ T(M+1).$$

This enables us to compute the ψ_s, ϕ_s, and θ_s error coefficients of the global error at the end of the projective step starting from a correct y_0. This is precisely the local error in one PFE step, which is also an inner step at the next higher telescoping layer.

Let the superscript $q \geq 0$ refer to the current telescoping layer, with $q = 0$ indicating the innermost integrator. Thus

$$E_s^0\Upsilon_s^0(h) = [\psi_s^0,\ \phi_s^0,\ \theta_s^0]\ \Upsilon_s^0(h)$$

is the error at the end of the first PFE step. Since

$$\Upsilon_1^{q+1}(sh) = R(s)\Upsilon_s^q(h)$$

where the scaling operator R is

$$R(s) = \begin{bmatrix} s^2 & 0 & 0 \\ 0 & s^3 & 0 \\ 0 & 0 & s^3 \end{bmatrix},$$

we have

$$\left[\xi^{q+1},\ \gamma^{q+1},\ \eta^{q+1}\right] = E_s^q R^{-1}(s).$$

Finally, if the innermost integrator is forward Euler, then $\xi_j^0 = 1$, $\gamma_j^0 = -2$ and $\eta_j^0 = 0$.

Now the above provides us with formula for computing the error coefficients at each layer; namely, for $q = 0, 1, \cdots$:

$$\psi_0^q = \phi_0^q = \theta_0^q = 0;$$

$$\begin{aligned} \psi_{j+1}^q &= \psi_j^q + \xi^q, \\ \phi_{j+1}^q &= \phi_j^q + \gamma^q - 3\psi_j^q, \\ \theta_{j+1}^q &= \theta_j^q + \eta^q + \psi_j^q; \end{aligned}$$

$$\begin{aligned} \psi_s^q &= (M+1)\psi_{k+1}^q - M\psi_k^q + M(M+1), \\ \phi_s^q &= (M+1)\phi_{k+1}^q - M\phi_k^q - 3M(M+1)(\psi_{k+1}^q - \psi_k^q) - M(M+1)(2M+1), \\ \theta_s^q &= (M+1)\theta_{k+1}^q - M\theta_k^q; \end{aligned}$$

$$\xi^{q+1} = \frac{\psi_s^q}{s^2}, \qquad \gamma^{q+1} = \frac{\phi_s^q}{s^3}, \qquad \eta^{q+1} = \frac{\theta_s^q}{s^3}.$$

Thus we can compute the second-order and third-order error terms in each of the inner layers, and then finally combine them to obtain the scaled local error coefficients for the outer projective integrator step.

3.1. Projective Runge-Kutta (PRK). Let us suppose a PRK step is taken at the outermost layer q. Let H denote its step size, and $h = H/s$ denote the inner damping step size at layer $q-1$. As described in [7, §4.1], one PRK step size H has the form

$$y_s = y_{k+1} + M(\alpha[y_{k+1} - y_k] + (1-\alpha)[y_{s+k_1+1} - y_{s+k_1}]). \tag{3.6}$$

The values $k+1$ and k_1+1 are the number of inner damping steps taken at the step starting from t_0 and t_s, respectively. The projective step multiplier is M, and the scalar α is used for computing a weighted average of the chord slopes. We want to find the *local error* in the PRK step, so we assume that we are starting from a correct $y(t_0)$. The predictor phase of PRK starts with a PFE step to compute the predicted value, p_s^{q-1}, which has an error of

$$D_P^q(H) = E_s^{q-1}\Upsilon_s(h).$$

Next we take k_1 steps of size H/s from p_s^{q-1} at layer $q-1$. From the above analysis, the additional local error introduced is

$$e_{k_1}^{q-1}(h) = E_{k_1}^{q-1}\Upsilon_{s+k_1}(h).$$

Thus the total error after these k_1 lower layer steps to order three is

$$y_{s+k_1} - y(t_{s+k_1}) = (I + hk_1J)D_P^q(H) + e_{k_1}^{q-1}(h).$$

We then take one more step and, after this step, the total error is

$$y_{s+k_1+1} - y(t_{s+k_1+1}) = (I + h(k_1+1)J)D_P^q(H) + e_{k_1+1}^{q-1}(h).$$

Noting that

$$y(t_s) = y(t_{k+1}) + M(\alpha[y(t_{k+1}) - y(t_k)] + (1-\alpha)[y(t_{s+k_1+1}) - y(t_{s+k_1})]) - d_s^{\mathrm{PRK}}(h,\alpha), \tag{3.7}$$

where the PRK discretization error, $d_s^{\mathrm{PRK}}(h,\alpha)$, is given by

$$\begin{aligned} d_s^{\mathrm{PRK}}(h,\alpha) = &-\frac{h^2}{2}y_s''[2M\alpha(M+1+k_1) - M(M+1+2k_1)] \\ &-\frac{h^3}{6}y_s'''[3M\alpha(k_1-M)(M+1+k_1) + M(M^2 - 3k_1(1+k_1) - 1)], \end{aligned} \tag{3.8}$$

we can subtract eq. (3.7) from eq. (3.6) to get the PRK local error we are looking for:

$$\begin{aligned} e_s^{q-1}(h) &= y_s - y(t_s) \\ &= e_{k+1}^{q-1}(h) + M(\alpha[e_{k+1}^{q-1}(h) - e_k^{q-1}(h)] + (1-\alpha)[e_{s+k_1+1}^{q-1}(h) - e_{s+k_1}^{q-1}(h)]) + d_s^{\mathrm{PRK}}(h,\alpha). \end{aligned} \tag{3.9}$$

The various $e^{q-1}(h)$ errors can be expressed in terms of $\Upsilon_s(h)$ (omitting the common $q-1$ superscripts) as

$$\begin{aligned} e_k(h) &= [\psi_k, && \phi_k - 3(M+1)\psi_k, && \theta_k &&]\,\Upsilon_s(h), \\ e_{k+1}(h) &= [\psi_{k+1}, && \phi_{k+1} - 3M\psi_{k+1}, && \theta_{k+1} &&]\,\Upsilon_s(h), \\ e_{s+k_1}(h) &= [\psi_s + \psi_{k_1}, && \phi_s + \phi_{k_1} + 3k_1\psi_{k_1}, && \theta_s + \theta_{k_1} + k_1\psi_s &&]\,\Upsilon_s(h), \\ e_{s+k_1+1}(h) &= [\psi_s + \psi_{k_1+1}, && \phi_s + \phi_{k_1+1} + 3(k_1+1)\psi_{k_1+1}, && \theta_s + \theta_{k_1+1} + (k_1+1)\psi_s]\,\Upsilon_s(h). \end{aligned}$$

Now, all we have to do is express $e_s^{q-1}(h)$ in the form

$$e_s^{q-1}(h) = \left[\xi^{PRK}(\alpha),\ \gamma^{PRK}(\alpha),\ \eta^{PRK}(\alpha)\right]\ \Upsilon_1(H)$$

and we have the local error for one PRK outer step size H. In this expression, α must be chosen to make $\xi^{\mathrm{PRK}}(\alpha) = 0$ to get a second-order accurate method and consequently determine the third-order error coefficients γ^{PRK} and η^{PRK}.

Computationally, one step of PRK with local error estimation can be accomplished as follows. After one outer PFE step size H yields the prediction

$$y_s^{\mathrm{PFE}} = (M+1)y_{k+1} - My_k,$$

the corrector step is

$$y_s^{\mathrm{PRK}} = y_s^{\mathrm{PFE}} + (M\alpha - M)[(y_{k+1} - y_k) - (y_{s+k_1+1} - y_{s+k_1})].$$

Note that α appears linearly in eq. (3.9) and eq. (3.8). It will be computationally convenient to work with $M\alpha$ and express eq. (3.8) as $d_s^{\mathrm{PRK}}(h,\alpha) = [M\alpha, 1]\ D^{\mathrm{PRK}}\Upsilon_s(H)$ where

$$D^{\mathrm{PRK}} = \begin{bmatrix} 2(M+1+k_1) & 3(k_1-M)(M+1+k_1) & 0 \\ -M(M+1+2k_1) & M(M^2 - 3k_1(1+k_1) - 1) & 0 \end{bmatrix},$$

and similarly express the $e(h)$ errors in eq. (3.9) by

$$[M\alpha, 1] \;\; E^{\mathrm{PRK}}\Upsilon_s(h) = [M\alpha, 1] \;\; \begin{bmatrix} e_{k+1}^{q-1}(h) - e_k^{q-1}(h) - (e_{s+k_1+1}^{q-1}(h) - e_{s+k_1}^{q-1}(h)) \\ \\ e_{k+1}^{q-1}(h) + M(e_{s+k_1+1}^{q-1}(h) - e_{s+k_1}^{q-1}(h)) \end{bmatrix}.$$

The PRK local error can be compactly expressed as

$$e_s^{q-1}(h) = [M\alpha, 1]\,(E^{\mathrm{PRK}} + D^{\mathrm{PRK}})\Upsilon_s(h).$$

Now we can determine the value of $M\alpha$ that gives second-order accuracy from the linear system

$$[M\alpha, 1] \underbrace{\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix}}_{C = E^{\mathrm{PRK}} + D^{\mathrm{PRK}}} \underbrace{\begin{bmatrix} -(H/s)^2\; y_s''/2 \\ -(H/s)^3\; y'''/6 \\ -(H/s)^3\; Jy''/2 \end{bmatrix}}_{\Upsilon_s(h)} = \underbrace{-\xi^{\mathrm{PRK}}\frac{H^2}{2}y_s'' - \gamma^{\mathrm{PRK}}\frac{H^3}{6}y''' - \eta^{\mathrm{PRK}}\frac{H^3}{2}Jy''}_{e_s^{q-1}(h) = e_1^q(H)}. \tag{3.10}$$

Second-order accuracy is achieved by setting $C = E^{\mathrm{PRK}} + D^{\mathrm{PRK}}$, and then zeroing the ξ^{PRK} term in eq. (3.10) by satisfying $M\alpha\, C_{11} + C_{21} = 0$. The result is

$$M\alpha = -C_{21}/C_{11}$$

and that the third-order error coefficients are:

$$\gamma^{\mathrm{PRK}} = \frac{(-C_{21}/C_{11})\; C_{12} + C_{22}}{s^3}, \qquad \eta^{\mathrm{PRK}} = \frac{(-C_{21}/C_{11})\; C_{13} + C_{23}}{s^3}.$$

3.2. Projective Adams-Bashforth (PAB). As with PRK, the current outer PAB step size is H and the inner damping steps are each of size $h = H/s$. Let us denote the previous PAB outer and inner step sizes as H_{-1} and h_{-1}, respectively. The projective step multiplier at the current and previous step are M and M_{-1}. Starting from t_0, the number of inner damping steps is $k+1$. One step of PAB [7, §4.2] has the form

$$y_s = y_{k+1} + M\left(\alpha\left[y_{k+1} - y_k\right] + (1-\alpha)r\left[y_{-M_{-1}} - y_{-(1+M_{-1})}\right]\right), \tag{3.11}$$

where

$$r = h/h_{-1}.$$

The last term in eq. (3.11) is the chord slope used in the previous outer step, multiplied by $(1-\alpha)h$. The scalar α is used for computing a weighted average of the current and previous chord slopes. As a two-step method, we note in general that h (and possibly the number of inner layers, $q-1$) will vary for each outer PAB step. In computing the local error committed in the PAB step, we assume that the y values are correct at the start of the current and previous outer steps.

Noting that

$$y(t_s) = y(t_{k+1}) + M\left(\alpha\left[y(t_{k+1}) - y(t_k)\right] + (1-\alpha)r\left[y(t_{-M_{-1}}) - y(t_{-(1+M_{-1})})\right]\right) - d_s^{\mathrm{PAB}}(h,\alpha), \tag{3.12}$$

we can subtract eq. (3.12) from eq. (3.11) to obtain the PAB local error

$$e_s(h) = e_{k+1}(h) + M(\alpha[e_{k+1}(h) - e_k(h)] + (1-\alpha)r[e_{-M_{-1}}(h_{-1}) - e_{-(1+M_{-1})}(h_{-1})]) + d_s^{\mathrm{PAB}}(h,\alpha). \tag{3.13}$$

By expansion, we find that the PAB discretization error is $d_s^{\rm PAB}(h,\alpha) = [M\alpha, 1]\ D^{\rm PAB}\Upsilon_s(h)$ with

$$D_{11} = 1 + 2M - (1 + 2M_{-1})r^{-1} - 2s, \tag{3.14}$$

$$D_{12} = -1 - 3M(M+1) + (1 + 3M_{-1}(1 + M_{-1}))r^{-2} + 3(1 + 2M_{-1})r^{-1}s + 3s^2, \tag{3.15}$$

$$D_{13} = 0, \tag{3.16}$$

$$D_{21} = M(-M + (1 + 2M_{-1})r^{-1} + 2s), \tag{3.17}$$

$$D_{22} = M(M^2 - (1 + 3M_{-1}(1 + M_{-1}))r^{-2} - 3(1 + 2M_{-1})r^{-1}s - 3s^2), \tag{3.18}$$

$$D_{23} = 0. \tag{3.19}$$

From eq. (3.13), the current $e(h)$ errors can be expressed in terms of $\Upsilon_s(h)$ as

$$e_k(h) = [\psi_k(h), \quad \phi_k(h) - 3(M+1)\psi_k(h), \quad \theta_k(h)] \quad \Upsilon_s(h), \tag{3.20}$$

$$e_{k+1}(h) = [\psi_{k+1}(h), \ \phi_{k+1}(h) - 3M\psi_{k+1}(h), \quad \theta_{k+1}(h)]\, \Upsilon_s(h). \tag{3.21}$$

The previous errors, computed with inner step size h_{-1}, need to be translated forward in time to t_s and then scaled so that the low-order terms are based on the current inner step size h; thus, we have

$$e_{-M_{-1}}(h_{-1}) = \left[\psi_{-M_{-1}}(h_{-1}), \quad \phi_{-M_{-1}}(h_{-1}), \quad \theta_{-M_{-1}}(h_{-1})\right] \quad T(q_1)R(1/r)\Upsilon_s(h), \tag{3.22}$$

$$e_{-(1+M_{-1})}(h_{-1}) = \left[\psi_{-(1+M_{-1})}(h_{-1}), \ \phi_{-(1+M_{-1})}(h_{-1}), \ \theta_{-(1+M_{-1})}(h_{-1})\right] T(q_2)R(1/r)\Upsilon_s(h) \tag{3.23}$$

where

$$q_1 = M_{-1} + rs \qquad \text{and} \qquad q_2 = 1 + M_{-1} + rs.$$

The ψ, ϕ, θ values used in eqs. (3.22)–(3.23) are actually the $\psi_{k+1}, \phi_{k+1}, \theta_{k+1}$ and ψ_k, ϕ_k, θ_k values calculated in the previous outer step (for the value of k used in the previous outer step).

The PAB step with local error estimation proceeds by taking one outer PFE step so that

$$y_s^{\rm PAB} = y_s^{\rm PFE} + (M\alpha - M)\left[(y_{k+1} - y_k) - r\left(y_{-M_{-1}} - y_{-(1+M_{-1})}\right)\right].$$

The PAB coefficient $M\alpha$ that gives second-order accuracy is determined from the linear system

$$e_s(h) = [M\alpha, 1]\ C\ \Upsilon_s(h) = \left[\xi^{\rm PAB}, \ \gamma^{\rm PAB}, \ \eta^{\rm PAB}\right]\ \Upsilon_1(H),$$

where $C = E^{\rm PAB} + D^{\rm PAB}$ is a 2×3 matrix. The entries for $D^{\rm PAB}$ are given in eqs. (3.14)–(3.19). The matrix $E^{\rm PAB}$ comes from eqs. (3.13) and (3.20)–(3.23) for the expansion of the $e(h)$ and $e(h_{-1})$ errors:

$$[M\alpha, 1]\ E^{\rm PAB}\ \Upsilon_s(h) = [M\alpha, 1] \begin{bmatrix} e_{k+1}(h) - e_k(h) - r\left(e_{-M_{-1}}(h_{-1}) - e_{-(1+M_{-1})}(h_{-1})\right) \\ \\ e_{k+1}(h) + Mr\left(e_{-M_{-1}}(h_{-1}) - e_{-(1+M_{-1})}(h_{-1})\right) \end{bmatrix}.$$

Finally, the PAB local error is made second-order accurate by using $C = E^{\rm PAB} + D^{\rm PAB}$ to get $M\alpha = -C_{21}/C_{11}$ and the result that

$$\xi^{\rm PAB} = 0, \qquad \gamma^{\rm PAB} = \frac{(-C_{21}/C_{11})\ C_{12} + C_{22}}{s^3}, \qquad \eta^{\rm PAB} = \frac{(-C_{21}/C_{11})\ C_{13} + C_{23}}{s^3}.$$

3.3. Multistep State Extrapolation Methods. In [8], Sommeijer describes a simple technique for increasing the real stability interval of explicit integrators (i.e., one-step explicit Runge-Kutta methods of first- through fourth-order accuracy). In the context of projective integrators, we can apply the technique in taking an outer step of size $H = (k + M)h = sh$ for integer $k \geq 1$. The general idea is first to compute y_M

by extrapolating the solutions y obtained from a few previously computed outer steps (including the current one). Then, y_s is obtained from y_M after performing k projective integrator steps of size $h = H/s$. For the latter integrator, PFE or PRK are obvious candidates since they are the first- and second-order accurate projective counterparts of one-step explicit Runge-Kutta methods. For the extrapolation to y_M, a variety of linear, quadratic or higher-order schemes can be applied. An extensive stability and accuracy analysis for such Multistep State Extrapolation Methods (MSEMs) is provided in [10, 9]. The potential advantage of a relatively large extrapolation step is the reduced cost in computing the solution at the next outer step. However, if M/k is too large, k explicit (and stable) integrator steps may not be sufficient to regain stability and/or recover enough of the accuracy lost via the extrapolation.

We will now briefly consider local error estimation for the simplest MSEM for projective integrators, which we shall call extrapolated projective forward Euler (XPFE). For the first part of the XPFE step, we have

$$y_M = y_0 + r\ (y_0 - y_{-1}) \tag{3.24}$$

where

$$r = (Mh)/H_{-1}.$$

The last term in eq. (3.24) is the chord slope across the previous outer step, multiplied by Mh. The local error for this extrapolation, $e_M(h)$, comes from

$$e_M(h) = e_0(h) + r\left(e_0(h) - e_{-M/r}(h)\right) + d_M(h),$$

where the discretization error is

$$d_M(h) = -\frac{h^2}{2}y''_M\left[M^2(1+1/r)\right] - \frac{h^3}{6}y'''\left[-M^3(1+r)(1+2r)r^{-2}\right].$$

In this case, $e_M(h) = d_M(h)$ because the standard local error analysis assumes no errors are present in y at the current and previous outer steps. For the full XPFE step, additional local error $e_k^{\mathrm{PFE}}(h)$ is introduced in taking k PFE steps of size h from t_M out to t_s. Local error is also incurred from error propagation due to the differential equation. Thus, an intermediate approximation for the XPFE local error is

$$e_s^{\mathrm{XPFE}}(h) = (I + hkJ)\ e_M(h) + e_k^{\mathrm{PFE}}(h). \tag{3.25}$$

Finally, $e_M(h)$ in eq. (3.25) needs to be translated forward for evaluation at time t_s so that the XPFE error coefficients in E_s^{XPFE} satisfy

$$e_s^{\mathrm{XPFE}}(h) = E_s^{\mathrm{XPFE}}\Upsilon_s(h)$$

for the low-order error terms. The XPFE scaled error coefficients are

$$\left[\xi^{\mathrm{XPFE}},\ \gamma^{\mathrm{XPFE}},\ \eta^{\mathrm{XPFE}}\right] = E_s^{\mathrm{XPFE}}R^{-1}(s).$$

MSEMs of greater sophistication than eq. (3.24) have been developed for computing y_M, and are described in [9]. For example, such MSEMs include: linear (or quadratic) schemes with improved accuracy from using more than two (or three) previously computed solutions for the extrapolation; or, a linear extrapolation scheme with an adjustable scalar parameter $\alpha \in [0, 1]$ for balancing better stability against increased accuracy. Now, if we continue in a slightly more general way, y_s is obtained from y_M after performing k steps with a time-stepper code Φ with constant step size h:

$$y_s = \Phi_h^k(y_M).$$

The time-stepper code can be a projective integrator (possibly with, say, an inner telescopic PFE integrator), or some type of black-box code. After determining the local error for the extrapolation $e_M^{\mathrm{MSEM}}(h)$ and the additional local errors introduced (and propagated) by k iterations of the time-stepper Φ, the error coefficients in E_s for the MSEM-Φ integrator follows from

$$e_s(h) = (I + hkJ)\ e_M^{\mathrm{MSEM}}(h) + e_k^{\Phi}(h) = E_s \Upsilon_s(h).$$

The MSEM-Φ scaled error coefficients are then given by $E_s R^{-1}(s)$.

3.4. Summary. Before concluding this section, we find it important to note the close resemblances between second-order accurate projective Runge-Kutta and Adams-Bashforth integrators and their well-known conventional counterparts. In fact, the conventional integrators are obtained when we use a forward Euler inner integrator with zero layers of telescoping ($L = 0$) and zero inner damping steps (e.g., $k = k_1 = 0$) for the outer integrator.

At the outermost layer, PRK is a one (outer) step method and that step is mainly based on a weighted average of slopes estimated near the current and next time step (t and t_s). If ξ is the second-order scaled error coefficient for each constant inner damping step size h, the linear system eq. (3.10) yields

$$M\alpha = \frac{M(M+1+2k_1) - \left(\sum_{i=0}^{k} \xi_i + M\xi_{s+k_1}\right)}{2(M+1+k_1) + \xi_k - \xi_{s+k_1}}, \tag{3.26}$$

which is equivalent to the PRK weighting coefficient α as derived in [7]. A similar, simplified expression occurs when the two-step method PAB uses a constant outermost step size H, and a constant inner damping step size within each outermost step:

$$M\alpha = M + \frac{M(M+1) + \left(\sum_{i=0}^{k} \xi_i + M\xi_{-(1+M_{-1})}\right)}{2(M+1+k) - \xi_k + \xi_{-(1+M_{-1})}}. \tag{3.27}$$

Note: Our PAB α coefficient in [7] is incorrect due to errors in sign and subscripts, and should instead be equivalent to eq. (3.27). The PRK and PAB coefficients from eq. (3.26) and eq. (3.27) asymptotically approach the conventional Runge-Kutta ($\alpha \to 1/2$) and Adams-Bashforth ($\alpha \to 3/2$) coefficients as the projective step size interval (Mh) becomes larger relative to the interval covered by the inner damping steps (e.g., $M \gg \max(k, k_1)$ for PRK). Finally, the PAB analysis in Section 3.2 is important because the $M\alpha$ for second-order accuracy is valid even if the outermost step H changes (though we continue to assume that the inner damping step sizes are constant within each outermost step).

From Sections 3.1–3.3, recurrence formulas for the on-the-fly calculation of local error coefficents have now been derived for: first-order accurate projective forward Euler (with telescoping); second-order accurate projective Runge-Kutta and Adams-Bashforth; and, Multistep State Extrapolation Methods. The technique is obviously more complicated than Richardson extrapolation, but it is fairly straightforward and cheap to implement (with careful programming). Furthermore, on-the-fly error estimation can be applied even if the inner integrator is not telescopic PFE—but rather some black-box time-stepper code (whose scaled error coefficients per step can be estimated). In this paper, however, we mainly focus on using projective integrators as stiff ODE initial-value problem solvers. As demonstrated in Section 4, the new local error estimation techniques enable us to estimate the second- and third-order error coefficients, and leads to adaptive projective integrators that can solve increasingly stiff time-dependent systems with improved efficiency (relative to using Richardson extrapolation for error estimation).

4. Numerical results. We will now compare the performance of local error estimation techniques for the PFE, XPFE, PRK and PAB integrators as applied to a stiff ODE initial-value system. The local error estimation techniques are Richardson extrapolation, and the new one developed in Section 3. The main purpose is to demonstrate that the new local error estimation technique works well on a standard stiff

ODE test problem, and that it roughly gives a factor of three reduction in cost relative to using Richardson extrapolation.

For Richardson extrapolation (RE), a solution y_1 is computed for the outer step size H_n; then, the solution y_2 is computed using two outer steps of size $H_n/2$. The local error is estimated as

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

where p is the order of accuracy of the outer integrator. Outer step size selection or rejection is then based on the estimated local error, current step size, and user-specified error tolerances (absolute and/or relative) so that

$$H_{n+1} = H_n \left(\frac{C_n}{\|\text{error}_n\|} \right)^{1/(p+1)} \tag{4.1}$$

for a positive method-dependent constant C_n and a weighted, tolerance-dependent norm. Typically, a step H_n is rejected if $\|\text{error}_n\| > 1$. Note that RE has the overhead expense of the two additional integrations required to compute y_2. The desire to eliminate this additional expense was one of the primary motivations for the development of the new local error estimation technique, in order to reduce the cost (i.e., total number of function evaluations $f(t, y)$) for the time integration.

The test problem is the two-dimensional diffusion equation

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

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

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

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 Jacobian spectral radius is $\rho(J) = 8(n+1)^2$, and the eigenvalues are strictly real (because the Jacobian is symmetric) and spread across the interval $(0, -\rho(J)]$.

For the outer projective integrator step size H, we apply $K + 1 = 3$ damping steps of size $h = H/S$ with S specified as follows: PFE ($S = 7$), PRK ($S = 14$), PAB ($S = 7$). For the XPFE integrator, we use $K = 3$ and $S = 7$ so that the cost for one outer step size H is the same as for PFE. For the inner integrator, we use telescopic PFE ($k = 1, s = 3.95$) with innermost step size $h_0 \approx 1/\rho(J)$ and with enough inner telescoping layers L so that the telescopic step is of size $h = s^L h_0$. Under these conditions for the test problems, the combined outer-inner projective integrators are $[0, 1]$-stable—which means the outer step size H can be adaptively chosen based on accuracy (not stability) considerations. A full discussion on the stability of combined outer-inner projective integrators can be found in [7].

The absolute and relative error tolerances are both set to 10^{-3}. After estimating the weighted-norm of the local error, step size selection is based on eq. (4.1) with $C_n = 1$. For the numerical experiments with on-the-fly local error estimation, we do not implement a step size rejection strategy. Instead, a step that incurs a large local error is naturally followed by a reduction in the next step size as dictated by eq. (4.1). Given that the local error has the general form

$$\text{error} = -\xi \frac{H^2}{2} y'' - \gamma \frac{H^3}{6} y''' - \eta \frac{H^3}{2} J y'', \tag{4.2}$$

derivative terms at t_{n+1} can be estimated using

$$H^2 y''_{n+1} = H\left[f(t_{n+1}, y_{n+1}) - f(t_n, y_n)\right]$$

	A			B		
PFE:	Richardson extrapolation			On-the-fly		
N	f-evals	steps	error	f-evals	steps	error
100	1,198	56 (0)	4.2e-4	253	20 (0)	3.7e-3
400	2,136	55 (0)	4.3e-4	409	17 (0)	9.3e-3
1600	4,138	56 (0)	4.1e-4	800	18 (0)	3.4e-3
6400	8,146	56 (0)	4.7e-4	1,628	18 (0)	1.1e-2

TABLE 4.1
Solver statistics for projective forward Euler (PFE): 2D diffusion PDE.

	A			B		
XPFE:	Richardson extrapolation			On-the-fly		
N	f-evals	steps	error	f-evals	steps	error
100	736	23 (1)	6.1e-3	228	18 (0)	8.7e-3
400	1,721	35 (1)	4.7e-3	451	21 (0)	6.9e-3
1600	3,761	49 (1)	3.4e-3	855	21 (0)	6.8e-3
6400	8,165	56 (1)	3.2e-3	1,832	22 (0)	7.1e-3

TABLE 4.2
Solver statistics for extrapolated projective forward Euler (XPFE): 2D diffusion PDE.

and

$$H^3 y_{n+1}^{'''} = -12(y_{n+1} - y_n) + 6H\left[f(t_{n+1}, y_{n+1}) + f(t_n, y_n)\right].$$

In contrast with Section 3, note that the subscripts for y_n and y_{n+1} now refer to computed solutions at the current and next outer step. Finally, in this study, the local error estimate used for each projective integrator is simply—

$$\text{PFE} : -\xi\frac{H^2}{2}y_{n+1}^{''}, \quad \text{XPFE} : -\xi\frac{H^2}{2}y_{n+1}^{''}; \quad \text{PRK} : -\gamma\frac{H^3}{6}y_{n+1}^{'''}, \quad \text{PAB} : -\gamma\frac{H^3}{6}y_{n+1}^{'''}.$$

For first-order accurate PFE and XPFE, only the ξ term is needed for estimating the $\mathcal{O}(H^2)$ error. For second-order accurate PRK and PAB, the ξ term in eq. (4.2) is eliminated by the proper choice of $M\alpha$ as derived in Sections 3.1–3.2, respectively. Of the two $\mathcal{O}(H^3)$ error terms in eq. (4.2), we focus on the effectiveness of using only the γ term in approximating the local error. The η term can also be included, but there is the additional expense of obtaining $Jy^{''}$ via finite-differences or with a user-supplied routine. The relative merits of including the η term is certainly a topic worthy of further investigation.

For problems of different sizes N, Tables 4.1–4.4 show the number of function evaluations, number of accepted and (rejected) time 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.

The numerical results can be summarized as follows. For first-order accurate PFE and XPFE, Tables 4.1–4.2 show that solving the test problems with the new local error estimation technique (instead of RE) yields about a factor of four reduction in the total number of function evaluations. For second-order accurate PRK and PAB, Tables 4.3–4.4 show the reduction in cost is about a factor of three and two, respectively. Evidently, the higher expense of using RE is partly due to its more conservative step size selection; that is, the integrators end up taking more time steps with RE (for PFE, XPFE and PRK). Fortunately, for a given test problem, the increased expense of using RE is somewhat offset by the higher accuracy obtained relative to using on-the-fly error estimation. At this point, we remark that a more extensive investigation

	A			B		
PRK:	Richardson extrapolation			On-the-fly		
N	f-evals	steps	error	f-evals	steps	error
100	1,344	40 (0)	2.4e-4	397	29 (0)	4.6e-3
400	2,194	41 (0)	1.8e-4	640	26 (0)	3.8e-3
1600	4,202	42 (0)	3.1e-4	1,374	28 (0)	3.5e-3
6400	8,334	43 (0)	2.1e-4	2,912	30 (0)	5.8e-3

TABLE 4.3
Solver statistics for projective Runge-Kutta (PRK): 2D diffusion PDE.

	A			B		
PAB:	Richardson extrapolation			On-the-fly		
N	f-evals	steps	error	f-evals	steps	error
100	702	25 (0)	1.6e-4	341	27 (0)	6.8e-4
400	1,302	25 (0)	1.9e-4	602	24 (0)	2.1e-3
1600	2,550	25 (0)	1.8e-4	1,129	23 (0)	2.2e-3
6400	4,998	25 (0)	2.0e-4	2,331	25 (0)	9.0e-4

TABLE 4.4
Solver statistics for projective Adams-Bashforth (PAB): 2D diffusion PDE.

would require additional test runs that involve, for example: a range of tighter absolute and/or relative error tolerances; including a $Jy^{''}$ estimate and the η term for the PRK and PAB local error formula in eq. (4.2); a robust step size control strategy; and, a small set of other test problems (with varied degrees of stiffness, nonlinearity, and so on). The current results are a first proof-of-principle demonstration that the derivation and implementation of the techniques in Section 3 are the basis for a cost-effective local error estimation technique for solving stiff systems with a new class of explicit, multiscale integrators.

5. Conclusions and perspective. Adaptive projective integrator codes for solving stiff multiscale systems are still in their early stages of development. We have described a novel local error estimation technique for first- and second-order accurate projective integrators. This capability is an important component in the overall design of an adaptive step size stiff initial-value problem solver, or for enabling the adaptive time integration of coarse-grained multiscale systems [5, 6].

A comprehensive evaluation of on-the-fly error estimation techniques will involve the development and fine-tuning of a more advanced step size control strategy for when the outer step is accepted (or rejected). A robust step size control strategy also allows for fair comparisons between adaptive projective integrators and existing stiff initial-value problems solvers (e.g., implicit, exponential, and stabilized explicit integrators). With regard to local error estimation alone, we view these on-the-fly techniques as a cheap and not-too-complicated alternative that are likely to be more efficient than Richardson extrapolation.

Acknowledgements. 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] W. E, B. Engquist, The heterogeneous multiscale methods, Comm. Math. Sci. 1 (1) (2003) 87–132.
[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] S. L. Lee, C. W. Gear, Second-order accurate projective integrators for multiscale problems, Accepted by J. Comput. Appl. Math. (2006). See also Lawrence Livermore Technical Report UCRL-JC-212640 for recent revisions.

[8] B. P. Sommeijer, Increasing the real stability boundary of explicit methods, Computers & Mathematics with Applications 19 (6) (1990) 37–49.

[9] C. Vandekerckhove, D. Roose, Accuracy analysis of time integration schemes for stiff multiscale problems, Tech. Rep. TW 457, Katholieke Universiteit Leuven, Department of Computer Science (Apr. 2006).

[10] C. Vandekerckhove, D. Roose, K. Lust, Numerical stability analysis of an acceleration scheme for step size constrained time integrators, Tech. Rep. TW 426, Katholieke Universiteit Leuven, Department of Computer Science (Apr. 2005).