Convergence Results for the MATLAB ode23 Routine

H. Lamba* A.M. Stuart*

February 9, 1998

Abstract

We prove convergence results on finite time intervals, as the user-defined tolerance $\tau \to 0$, for a class of adaptive timestepping ODE solvers that includes the ode23 routine supplied in MATLAB Version 4.2. In contrast to existing theories, these convergence results hold with error constants that are uniform in the neighbourhood of equilibria; such uniformity is crucial for the derivation of results concerning the numerical approximation of dynamical systems. For linear problems the error estimates are uniform on compact sets of initial data. The analysis relies upon the identification of explicit embedded Runge-Kutta pairs for which all but the leading order terms of the expansion of the local error estimate are $\mathcal{O}(\|f(u)\|^2)$.

AMS Subject Classification: 34C35, 65L07, 65L20, 65L50.

Key words: Error Control, Adaptivity, Convergence, Tolerance Proportionality.

1 Introduction

Consider the autonomous initial value problem

$$\frac{du}{dt} = f(u), \quad u(0) = U \text{ where } f : \mathbb{R}^m \to \mathbb{R}^m \tag{1.1}$$

over some finite time interval $[0, T]$ with solution $u(t) = S(U, t)$. Convergence results for fixed timestep Runge-Kutta and multistep approximations to the solution of (1.1) have been long known. However most routines actually used to solve (1.1) employ some form of local error control to allow the stepsize to change during the integration. By requiring that an estimate of the local error at each timestep is less than some user-defined tolerance τ, and by allowing the timesteps to increase or decrease subject to this constraint, it is assumed (or hoped) that the numerical solution will converge to the true solution as $\tau \to 0$.

For fixed timestepping implementations, with timestep Δt, the error estimates take the form $C(B, T)\Delta t^r$ for all initial data $U \in B$, where B is a compact set. The uniformity of $C(\bullet, T)$ on compact sets is crucial to many results in the approximation of dynamical systems.

*Scientific Computing and Computational Mathematics Program, Division of Mechanics and Computation, Durand 262, Stanford University, Palo Alto CA 94305-4040 USA. Supported by NSF Grant DMS-95-04879.

It is therefore natural to try and seek estimates of the form $C(B,T)\tau^\gamma$, for all U contained in compact B, for adaptive timestepping algorithms.

Although adaptive timestepping methods have been used for many years, there are still very few rigorous convergence results. This is partly because the timestep selection algorithm, and hence the timestep sequence itself, is discontinuous with respect to the initial data U; this creates considerable problems in the analysis. Also, while there are only a few basic ideas underlying the error control mechanism, there are numerous possible differences in implementation making the mathematical description of a complete routine extremely complicated and perhaps too specific to be of general interest.

Stetter [11, 12] and Stoffer and Nipp [13] proved convergence results under assumptions which implied that as $\tau \to 0$ the maximum timestep, $\Delta t_{\max}$, also tended to 0. However Hall [5] showed that if the numerical trajectory enters the neighbourhood of a stable equilibrium point then the maximum timestep will approach the linear stability limit, independent of τ. Higham and Stuart [6] assumed instead that the timesteps obeyed the Tolerance Proportionality Assumption, i.e. that the truncation error $T(u,\Delta t) \leq K\tau^\gamma \Delta t$ for some positive constants K and γ. Under this assumption it is straightforward to show that the global error decreases as $\mathcal{O}(\tau^\gamma)$. This is a weaker assumption than those in [11, 12, 13] but still fails to hold in general. Stuart [14] took a different approach and examined the validity of the Tolerance Proportionality Assumption for a specific algorithm where the local error estimate is defined by a pair of Runge-Kutta methods. It was shown that the Tolerance Proportionality Assumption holds away from neighbourhoods of equilibria, denoted by the set $\Gamma(0)$, and the set where the leading term of the local error expansion, scaled by $\|f(u)\|$, disappears, denoted by $\Psi(0)$. As a result, the error estimates obtained in [14] are not uniform with respect to the distance of the exact trajectory from these sets.

In this paper we continue the analysis in [14]. We shall require all the higher-order terms in the expansion of the local error estimate to be $\mathcal{O}(\|f(u)\|^2)$ and also the much weaker condition that the lower-order Runge-Kutta method does not increase its order when applied to linear problems (both these conditions are satisfied by the embedded Runge-Kutta pair used in MATLAB Version 4.2 ode23). Then, by using an improved estimate of the truncation error close to equilibria, we are able to obtain convergence results that hold uniformly in the neighbourhood of equilibria $\Gamma(0)$. Thus only proximity to the set $\Psi(0)$ causes a lack of uniformity in the the error constants. In summary we derive error estimates of the form $C(B,T)\tau^\gamma$ for initial data in sets B whose forward image under the semigroup generated by (1.1) over the time interval $[0,T]$ is disjoint from $\Psi(0)$. The set $\Psi(0)$ is empty for autonomous linear problems defined by invertible matrices and hence a corollary of our results for this class of linear vector fields is that the MATLAB Version 4.2 ode23 routine has error estimates of the form $C(B,T)\tau^\gamma$ for any compact set of initial data B.

The requirement on the expansion of the local error estimate is satisfied by all explicit Runge-Kutta pairs of order $(p-1,p)$ with precisely p stages (for such pairs, the higher-order method must use p stages since the maximal order of an explicit ν-stage Runge-Kutta method is ν, but the lower-order method may use $p-1$ or p stages — either is allowable). Although the analysis does not require that the Runge-Kutta pair be embedded (i.e. that the two methods use the same intermediate stages) such pairs are always used in practice and so we shall henceforth refer only to embedded pairs. In particular the embedded Fehlberg (2,3) pair lies within this class and this is the pair used in the MATLAB Version 4.2 ode23 routine [1].

Such pairs only exist for $p \leq 4$ since the maximum order of an explicit ν-stage Runge-Kutta method for $\nu > 4$ is strictly less than ν (see [3]) but we note that by imposing extra conditions on the Runge-Kutta coefficients it may be possible to find higher-order embedded pairs with order $(p-1, p)$ and ν-stages $(\nu > p)$ for which the results obtained here also hold.

The MATLAB Version 4.2 ode23 routine is relatively simple, containing only the essential components of an adaptive timestepping algorithm. It has also been widely used and results concerning this algorithm are likely to be of special interest. Therefore we prove convergence results for a class of algorithms that specifically includes this routine. However the analysis presented here extends to general adaptive algorithms under very weak assumptions on the timestep selection mechanism [7]. In particular, the improved convergence results in the neighbourhood of equilibria hold for the same class of explicit Runge-Kutta pairs defined above.

The paper proceeds as follows. In §2 we describe the MATLAB Version 4.2 ode23 algorithm and the generalisation that we study. In §3 the convergence results are proved, first for linear ODEs and then for the general nonlinear case. The main convergence result is given in Theorem 3.13. In §4 we examine the two situations in which the convergence results can fail, namely, when the initial timestep estimate is spuriously accepted and when the exact solution of (1.1) passes through the set $\Psi(0)$. An argument is outlined to show that if the initial data and initial timestep estimate are treated as uniformly distributed random variables then for the MATLAB Version 4.2 ode23 routine, the probability of wrongly accepting the initial timestep tends to 0 as $\tau \to 0$. A corollary of Theorem 3.13 incorporating this probabilistic result for the MATLAB Version 4.2 ode23 routine is stated. Also, since the global error bound is not uniform with respect to the closeness of $u(t)$ to $\Psi(0)$ we follow [14] and, by calculating the probability of the exact solution $u(t)$ entering given neighbourhoods of $\Psi(0)$ for random initial conditions U, find the probability that the constant in the global error bound exceeds any given value. Finally, in §5 we briefly examine the new ode23 routine supplied in the recently released MATLAB Version 5.0. This routine has a more sophisticated timestep selection algorithm than the previous version but uses an embedded Runge-Kutta pair that does not fit into the class described above and we construct an example where this routine fails on a linear problem.

If the orders of the Runge-Kutta pairs used are $p-1$ and p then it is assumed throughout that $f \in C^{p+1}(\mathbb{R}^m, \mathbb{R}^m)$ (note that this also ensures the local existence and uniqueness of the exact solution $u(t)$) — more precise regularity conditions on f will also appear in the statements of the main results where appropriate.

2 The MATLAB Version 4.2 ode23 routine

For succinctness all MATLAB routines mentioned will refer to those supplied with MATLAB Version 4.2 unless otherwise stated. The MATLAM ode23 routine uses the embedded Fehlberg (2,3) pair of explicit Runge-Kutta methods in extrapolated error-per-step (XEPS) mode. That is, the higher-order method is used to advance the solution and the local error estimates are controlled on an error-per-step basis. Denoting the 3$^{\text{rd}}$-order method used to advance the approximation from a point $u \in \mathbb{R}^m$ by time t as $S^{(1)}(u, t)$ and the 2$^{\text{nd}}$-order method as

$S^{(2)}(u,t)$ we obtain

$$
\begin{aligned}
\eta_1 &= u, && (2.1)\\
\eta_2 &= u + tf(\eta_1), && (2.2)\\
\eta_3 &= u + \frac{t}{4}[f(\eta_1) + f(\eta_2)], && (2.3)\\
S^{(1)}(u,t) &= u + \frac{t}{6}[f(\eta_1) + f(\eta_2)] + \frac{2t}{3}f(\eta_3), && (2.4)\\
S^{(2)}(u,t) &= u + \frac{t}{2}[f(\eta_1) + f(\eta_2)]. && (2.5)
\end{aligned}
$$

The approximation to the local error (error-per-step) is

$$E(u,t) = \|S^{(1)}(u,t) - S^{(2)}(u,t)\|_\infty \tag{2.6}$$

and the values U_n and integration timesteps Δt_n are chosen by the following algorithm:

$$
\begin{aligned}
U_{n+1} &= S^{(1)}(U_n, \Delta t_n), \quad U_0 = U, && (2.7)\\
\text{where } \Delta t_n^{(0)} &= \Delta t_{\text{init}} \ \text{ if } \ n = 0 \ \text{ or} && (2.8)\\
& \quad \min\left(D, \theta\left(\frac{\tau \max(1, \|U_{n-1}\|_\infty)}{E(U_{n-1}, \Delta t_{n-1})}\right)^{\frac{1}{3}} \Delta t_{n-1}, T - \sum_{j=0}^{n-1} \Delta t_j\right) \ \text{ if } \ n > 0, && (2.9)\\
\Delta t_n^{(k)} &= \min\left(D, \theta\left(\frac{\tau \max(1, \|U_n\|_\infty)}{E(U_n, \Delta t_n^{(k-1)})}\right)^{\frac{1}{3}} \Delta t_n^{(k-1)}, T - \sum_{j=0}^{n-1} \Delta t_j\right) \ \text{ for } \ k \geq 1, && (2.10)\\
\text{and } \ \Delta t_n &= \Delta t_n^{(l)} \text{where } \ l = \min_{k \geq 0}\{k : E(U_n, \Delta t_n^{(k)}) \leq \tau \max(1, \|U_n\|_\infty)\}. && (2.11)
\end{aligned}
$$

The above algorithm generates an approximate solution U_n to $u(t_n)$ where $t_n = \sum_{j=0}^{n-1} \Delta t_j$. The parameter $\theta < 1$ is fixed at 0.9 within the routine, the maximum stepsize is automatically chosen to be $D = T/16$ and the tolerance τ is specified by the user. Unlike many routines, this algorithm does not include a maximum stepsize ratio. However the main convergence results presented here can be proved with minor modifications if a maximum stepsize ratio is included. Also, the initial guess Δt_{init} for the first timestep is fixed at $\Delta t_{\text{init}} = D/8 = T/128$. For integrations over long times this initial guess will be very large. It is much more natural for Δt_{init} to decrease with τ and this point is discussed in §4.1.

It is perhaps useful to think of the algorithm (2.1)–(2.11) as consisting of two separate parts; the *local error estimate*, defined by (2.1)—(2.6), and the *timestep selection mechanism* defined by (2.7)—(2.11). Rather than prove results only for the algorithm (2.1)–(2.11) we shall prove convergence results for a class of algorithms using explicit Runge-Kutta pairs of order $(p-1, p)$ with precisely p stages. Such embedded pairs only exist for $p \leq 4$ (see Theorem 323B [3]) and include the Fehlberg (2,3) pair; note however that this class does *not* include the MATLAB Version 4.2 ode45 routine which employs an identical timestep selection mechanism but uses a $(4, 5)$ embedded Runge-Kutta pair with 6 stages. We shall also consider all 4 combinations of error-per-step/error-per-unit-step and extrapolation/non-extrapolation modes using a timestep selection mechanism that includes (2.7)–(2.11) as a special case, but generalises it. The complete description of the class of timestepping algorithms under consideration is given by

$$\eta_i = u + t\sum_{j=1}^{i-1} a_{ij} f(\eta_j), \quad i = 1, \ldots, p \tag{2.12}$$

$$S^{(1)}(u,t) = u + t\sum_{j=1}^{p} b_j^{(1)} f(\eta_j), \tag{2.13}$$

$$S^{(2)}(u,t) = u + t\sum_{j=1}^{p} b_j^{(2)} f(\eta_j) \tag{2.14}$$

where $S^{(1)}$ is the method used to advance the solution. In extrapolation mode, $S^{(1)}$ will be of order p and $S^{(2)}$ of order $p-1$, otherwise the orders are reversed. The approximation to the local error is given by

$$E(U_n, \Delta t_n) = \|S^{(1)}(U_n, \Delta t_n) - S^{(2)}(U_n, \Delta t_n)\| / \Delta t_n^{\rho} \tag{2.15}$$

for some choice of norm $\| \bullet \|$, and where $\rho = 0$ in error-per-step mode and $\rho = 1$ in error-per-unit-step mode. At each step the local error estimate will be tested against a function $\sigma(\tau, U_n)$ where, for every compact set $J \subset \mathbb{R}^m$, there exist constants $C_1(J)$ and $C_2(J)$ such that

$$0 < C_1(J) \le \sigma(\tau, u)/\tau \le C_2(J) < \infty \ \ \forall \tau > 0 \ \text{ and } \ \forall u \in J. \tag{2.16}$$

Thus a wide range of error estimates with absolute and relative components are included in the analysis, including the specific combined absolute/relative error estimate used in MATLAB ode23.

Integration timesteps Δt_n are chosen by the following algorithm:

$$\begin{aligned}
U_{n+1} &= S^{(1)}(U_n, \Delta t_n), \quad U_0 = U, && (2.17)\\
\text{where } \Delta t_n^{(0)} &= \Delta t_{\text{init}} \ \text{ if } \ n = 0 \ \text{ or} && (2.18)\\
& \min\left(D, \theta\left(\frac{\sigma(\tau, U_{n-1})}{E(U_{n-1}, \Delta t_{n-1})}\right)^{\frac{1}{p-\rho}} \Delta t_{n-1}, T - t_n\right) \ \text{ if } \ n > 0, && (2.19)\\
\Delta t_n^{(k)} &= \min\left(D, \theta\left(\frac{\sigma(\tau, U_n)}{E(U_n, \Delta t_n^{(k-1)})}\right)^{\frac{1}{p-\rho}} \Delta t_n^{(k-1)}, T - t_n\right) \ \text{ for } \ k \ge 1, && (2.20)\\
\text{and } \ \Delta t_n &= \Delta t_n^{(l)} \text{where } \ l = \min_{k \ge 0}\{k : E(U_n, \Delta t_n^{(k)}) \le \sigma(\tau, U_n)\}. && (2.21)
\end{aligned}$$

We shall assume that $\theta < 1$, D and Δt_{init} are chosen by the algorithm but without specifying how. In §4.1 we consider the specific choice of Δt_{init} used by MATLAB ode23 and state a corollary of the main convergence result.

3 Convergence Results

In this section we prove convergence results for the class of algorithms (2.12)–(2.21). The results and methods of proof are similar to [14] but are stronger because of our restriction to specific explicit Runge-Kutta pairs and the use of tighter error bounds in the neighbourhood of equilibria.

After some necessary definitions and preliminary results we analyse the convergence for linear problems. Although this linear case is included in the general nonlinear analysis that follows, it is a useful exercise to consider it separately as it gives an indication of the strength of the results that can be expected in the general case.

3.1 Preliminaries

We have the following local Taylor series expansion for the evolution operator $S(u,t)$ of (1.1),

$$S(u,t) = \sum_{j=0}^{r+1} \frac{1}{j!}\beta_j^{(0)}(u,0)t^j + \frac{t^{r+2}}{(r+1)!}\int_0^1 (1-s)^{r+1}\beta_{r+2}^{(0)}(u;st)ds \quad \forall r \in \mathbb{Z}^+$$

where

$$\beta_j^{(0)}(u,t) = \frac{\partial^j}{\partial t^j}\{S(u,t)\}.$$

Similarly, for the Runge-Kutta approximations we have

$$S^{(k)}(u,t) = \sum_{j=0}^{r+1} \frac{1}{j!}\beta_j^{(k)}(u,0)t^j + \frac{t^{r+2}}{(r+1)!}\int_0^1 (1-s)^{r+1}\beta_{r+2}^{(k)}(u;st)ds \quad \forall r \in \mathbb{Z}^+,\ k=1,2 \tag{3.1}$$

where

$$\beta_j^{(k)}(u,t) = \frac{\partial^j}{\partial t^j}\{S^{(k)}(u,t)\}.$$

We now define positive integers s and q_1 by

$$\beta_j^{(1)}(u,0) \equiv \beta_j^{(0)}(u,0),\ \ j=0,\dots,s,$$

$$\beta_j^{(2)}(u,0) \equiv \beta_j^{(0)}(u,0),\ \ j=0,\dots,q_1,$$

where agreement between the β's does not occur at the next order in each case. Note that $s = p = q_1 + 1$ in extrapolation mode and $s = p - 1 = q_1 - 1$ in non-extrapolation mode.

Since $\beta_j^{(1)}(u,0) \equiv \beta_j^{(2)}(u,0),\ \ j = 0,\dots,p-1$ we obtain $E(u,t) = \mathcal{O}(t^q)$ where $q = p - \rho$. The ratio s/q plays a key role in the convergence proofs and so for clarity we list the values taken by s, q and s/q in each of the four operating modes:

$$\begin{array}{lrcrcl}
\text{Non-extrapolated error-per-step} & \text{EPS} & : & s/q & = & (p-1)/p\ <1 \\
\text{Extrapolated error-per-step} & \text{XEPS} & : & s/q & = & p/p\ =1 \\
\text{Non-extrapolated error-per-unit-step} & \text{EPUS} & : & s/q & = & (p-1)/(p-1)\ =1 \\
\text{Extrapolated error-per-unit-step} & \text{XEPUS} & : & s/q & = & p/(p-1)\ >1.
\end{array} \tag{3.2}$$

The truncation error

$$T(u,t) = \|S(u,t) - S^{(1)}(u,t)\|, \tag{3.3}$$

is the error committed by the advancing Runge-Kutta method at each timestep and the following result is proved as Lemma 2.5 of [14].

Lemma 3.1 *Let $J \subset \mathbb{R}^m$ be bounded. There is a constant $K = K(J, D)$ such that, for all $u \in J$, $t \in [0, D]$,*

$$T(u, t) \leq K\|f(u)\|t^{s+1}. \tag{3.4}$$

We define E_m (not to be confused with $E(U_m, \Delta t_m)$) to be the total error between the exact solution and the computed value after the m^{th} timestep. That is,

$$E_m = \|u(t_m) - U_m\|.$$

We shall always consider a solution, or families of solutions, of (1.1) defined for $t \in [0, T]$ with $U \in B$, B compact. Using the notation $S(B, t) = \bigcup_{U \in B} S(U, t)$ we shall assume that $S(B, t) \subseteq I \; \forall t \in [0, T]$ where I is some bounded set in $\mathbb{R}^m$. We then define $J = \overline{\mathcal{N}}(I, d)$ as the closure of the d-neighbourhood of I for some fixed $d > 0$, independent of τ. We will show that our numerical approximation lies in J and our results will hence involve constants depending upon J.

3.2 The linear case

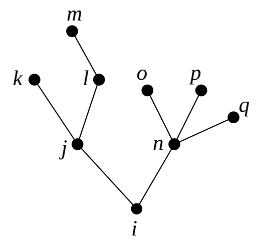
We first prove convergence results for the linear problem, $f(u) = Au,\ u(0) = U$ where A is an invertible $m \times m$ matrix. We shall need the following lemma.

Lemma 3.2 *Let $f(u) = Au$ where A is an invertible $m \times m$ matrix. If $S^{(1)}(u, t)$ and $S^{(2)}(u, t)$ are an explicit Runge-Kutta pair of orders p and $p - 1$ with precisely p stages then for some $\alpha \geq 0$*

$$E(u, t) = \alpha\|t^q A^p u\|.$$

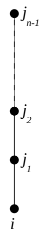
Proof Although the lemma can be proved directly, we give a proof using rooted trees as this provides useful insights when we come to consider the expansion of the local error estimate for nonlinear problems.

A rooted tree is a connected graph containing no cycles with one node identified as the 'root'. Each rooted tree with precisely n nodes corresponds to an expression appearing in the i^{th} component of $\beta_n^{(0)}(u, 0)$ for the general nonlinear problem. These expressions are the elementary differentials of order n and this correspondence is achieved as follows. Let $f^i_{j_1, j_2, \ldots, j_n}$ denote an n^{th} partial derivative of the i^{th} component of f. Now attach the label i to the root of the tree and labels $j, k, l \ldots$ to the other nodes. Then for each node write down f with a superscript equal to the label of that node and subscripts given by the nodes that are connected away from the root node. For example, the rooted tree



corresponds to $f^i_{jn} f^j_{kl} f^k f^l_m f^m f^n_{opq} f^o f^p f^q$ (using the summation convention) which is an elementary differential of order 9. It is proved in [3] that each $\beta_n^{(k)}(u, 0),\ k = 0, 1, 2$ are linear combinations of all the elementary differentials of order n.

Note that if a rooted tree has a ‘branch’ at any node then there must be at least two end nodes and the corresponding elementary differential will be $\mathcal{O}(\|f(u)\|^2)$. Thus for every $n \geq 1$, there exists precisely one elementary differential of order n which is $\mathcal{O}(\|f(u)\|)$ but not $\mathcal{O}(\|f(u)\|^2)$; this corresponds to the rooted tree with n nodes and no branches shown below



and equals $f^i_{j_1} f^{j_1}_{j_2} \dots f^{j_{n-2}}_{j_{n-1}} f^{j_{n-1}}$. In the proof of Theorem 321A of [3] it is shown that no elementary differential of order $n > \nu$ of this form appears in $\beta_n(u, 0)$ for a ν-stage explicit Runge-Kutta method (this in fact proves that the order of a ν-stage explicit Runge-Kutta method is $\leq \nu$).

Because f is linear, all second partial derivatives and higher vanish. Thus the non-vanishing elementary differentials correspond precisely to the branchless rooted trees shown above. Since the two explicit Runge-Kutta methods agree upto order $p-1$, the local error estimate consists of precisely one term, proportional to $f^i_{j_1} f^{j_1}_{j_2} \dots f^{j_{p-2}}_{j_{p-1}} f^{j_{p-1}}$. This, together with the boundedness of J and the fact that $f^i_j = \{A\}_{ij}$, proves the lemma. $\square$

The higher-order method cannot increase its order on linear problems since the maximal order of a p-stage method on linear problems is p. If the lower-order method increases its order to p on linear problems then both methods agree ($\alpha = 0$) resulting in a vacuous error estimate. We therefore exclude this case from our analysis.

We now prove the main convergence result for the linear case. For each of the four possible operating modes, the exponent γ in the rate of convergence $\mathcal{O}(\tau^\gamma)$ has been optimised with respect to the constraint that $K_1(\bullet, T)$ be uniform on compact sets.

Theorem 3.3 *Let $f(u) = Au$ in (1.1), where A is an invertible matrix. Consider the numerical approximation over the time interval $[0, T]$ generated by the algorithm (2.12)–(2.21) in XEPS, EPS, XEPUS or EPUS mode and assume that the lower-order method does not increase its order when applied to linear problems. Then for all compact sets $B \subset \mathbb{R}^m$ there exist constants $K_1(B, T), K_2(B, T)$ such that for all initial data $U \in B$, all $\Delta t_{\text{init}} > 0$, and all sufficiently small τ,*

$$E_n = \|u(t_n) - U_n\| \leq K_1 \tau^\gamma, \tag{3.5}$$

$$U_n \in J \quad \forall n : t_n \leq T \tag{3.6}$$

and

$$T(U_n, \Delta t_n) \leq K_2 \Delta t_n \tau^\gamma \tag{3.7}$$

where $\gamma = 1$ for XEPS, EPUS and XEPUS modes and $\gamma = (p-1)/p$ for EPS mode.

Proof Since B is bounded and the problem is linear there exists a bounded set I such that $S(B, t) \subseteq I \subset \mathbb{R}^m$ $\forall t \in [0, T]$ which in turn defines the compact set $J = \overline{\mathcal{N}}(I, d)$. We prove first that if (3.5) holds for $n = 0, \ldots, m$ then (3.6) holds for $n = 0, \ldots, m$. This follows since $E_n \leq K_1 \tau^\gamma$ so that for τ sufficiently small, $\|u(t_n) - U_n\| \leq d$. Then $u(t_n) \in I$ implies that $U_n \in J$.

We now prove (3.7). By the previous lemma we have

$$E(U_n, \Delta t_n) = \alpha \|\Delta t_n^q A^p U_n\|. \tag{3.8}$$

The assumption that the lower-order method does not increase its order when applied to a linear problem implies that $\alpha \neq 0$. The timestep control therefore implies

$$\alpha \|\Delta t_n^q A^p U_n\| \leq \sigma(\tau, U_n) \tag{3.9}$$

which immediately gives an upper bound for Δt_n^q (in Lemma 3.9 it will be proved that the timestep control must be satisfied after only finitely many timestep rejections).

From (3.9) and (3.4) and using the fact that

$$\|Au\| \leq \|A^{-p+1}\| \, \|A^p u\|$$

we get, for XEPS, EPUS and EPS modes,

$$T(U_n, \Delta t_n) \leq K \Delta t_n \frac{\|A^{-p+1}\| \, \|A^p U_n\|}{\|\alpha A^p U_n\|^{\frac{s}{q}}} \sigma(\tau, U_n)^{\frac{s}{q}} \tag{3.10}$$

and since $s/q \leq 1$ (see (3.2)) and J is bounded, there is a constant $K_3(J)$ such that

$$T(U_n, \Delta t_n) \leq K_3 \Delta t_n \sigma(\tau, U_n)^{\frac{s}{q}}.$$

Similarly for XEPUS mode,

$$T(U_n, \Delta t_n) \leq K \Delta t_n^2 \frac{\|A^{-p+1}\| \, \|A^p U_n\|}{\|\alpha A^p U_n\|^{\frac{(s-1)}{q}}} \sigma(\tau, U_n)^{\frac{(s-1)}{q}}$$

and since $(s-1)/q = 1$ and $\Delta t_n \leq D$, there is again a constant $K_3(J)$ such that

$$T(U_n, \Delta t_n) \leq K_3 \Delta t_n \sigma(\tau, U_n).$$

Note that in each case the exponent γ on σ is forced by requiring K_3 uniform on J. Thus, by (2.16), there is a constant $K_2(J)$ such that (3.7) holds.

It remains to be proved that if (3.5) holds for $n = 0, \ldots, m$ then it holds for $n = m + 1$. We shall do this by proving that there exists $L(J) > 0$ such that

$$E_n \leq K_2 t_n \tau^\gamma \exp(L t_n), \qquad \forall n \ : \ t_n \leq T. \tag{3.11}$$

If (3.11) holds for a given n then so does (3.5) with $K_1(J, T) = K_2(J) T \exp(L(J) T)$, so we shall prove (3.11) by induction. Clearly (3.11) holds for $m = 0$. Writing $u_m = u(t_m)$ we have

$$\begin{array}{rcccc} E_{m+1} & = & \|S(u_m; \Delta t_m) - S^{(1)}(U_m; \Delta t_m)\| & & \\ & \leq & \|S(u_m; \Delta t_m) - S(U_m; \Delta t_m)\| & + & \|S(U_m; \Delta t_m) - S^{(1)}(U_m; \Delta t_m)\| \\ & \leq & \exp(L \Delta t_m) E_m & + & T(U_m, \Delta t_m) \end{array}$$

where L is the Lipschitz constant of f on J. Therefore, by (3.7)

$$E_{m+1} \leq \exp(L\Delta t_m)E_m + K_2\Delta t_m\tau^\gamma$$

Since $1 \leq e^x$ for all $x \geq 0$ the inductive hypothesis (3.11) gives

$$\begin{aligned} E_{m+1} &\leq K_2 t_m \tau^\gamma \exp(Lt_{m+1}) + K_2\Delta t_m\tau^\gamma \exp(Lt_{m+1}) \\ &= K_2 t_{m+1}\tau^\gamma \exp(Lt_{m+1}). \end{aligned}$$

This completes the induction. Since J is defined by B the result follows. □

Note that due to our choice of linear problem together with a $(p-1,p)$ Runge-Kutta pair with precisely p stages the local error estimate (3.8) is just a single term. Hence $E(u,t)$ is monotonic in $t>0$ for every fixed u and is only small in some neighbourhood of $t=0$. This immediately gives an upper bound on Δt_n and hence an upper bound on $T(U_n,\Delta t_n)$. If a $(p-1,p)$ pair with $\nu > p$ stages is used instead then the local error estimate becomes

$$E(U_n,\Delta t_n) = \Delta t_n^q \sum_{i=p}^{i=\nu} \alpha_i \Delta t_n^{i-p} A^i U_n \tag{3.12}$$

and $E(u,t)$ is no longer monotonic in t and the analysis fails (see §5 for further discussion of this point).

This lack of monotonicity of $E(u,t)$ also occurs in the general nonlinear problem even if $\nu = p$ — for certain $\mathcal{O}(1)$ values of the timestep, the local error estimate may be close to zero, as well as for t small, possibly resulting in an accepted step with a large local error. Thus we shall have to obtain bounds that ensure this cannot occur before obtaining results analagous to Theorem 3.3 for the nonlinear case.

Since the lower-order method (2.5) does not increase its order on linear problems and MATLAB Version 4.2 ode23 operates in XEPS mode, we have the following corollary.

Corollary 3.4 *Let $f(u) = Au$ in (1.1), where A is an invertible matrix and consider the numerical approximation over the time interval $[0,T]$ generated by the MATLAB Version 4.2 ode23 routine. Then for all compact sets B there exist constants $K_1(B,T), K_2(B,T)$ such that for all initial data $U \in B$, all $\Delta t_{\text{init}} > 0$, and all sufficiently small τ,*

$$E_n = \|u(t_n) - U_n\| \leq K_1\tau \tag{3.13}$$

and

$$T(U_n,\Delta t_n) \leq K_2\Delta t_n\tau. \tag{3.14}$$

We now return to the general framework of Theorem 3.3. We can increase the convergence rate in the linear case for XEPUS mode to $\tau^{\frac{p}{p-1}}$ (which is the convergence rate observed in most numerical experiments for trajectories which remain far from equilibria) by assuming that the exact solution remains outside some given neighbourhood of the equilibrium point at 0.

Theorem 3.5 *Let $f(u) = Au$ in (1.1), where A is an invertible matrix. Consider the numerical approximation over the time interval $[0,T]$ generated by the algorithm (2.12)–(2.21) in XEPUS mode and assume that the lower-order method does not increase its order when applied to linear problems. Assume also that $\delta = \frac{1}{2}\min_{U\in B}\min_{t\in[0,T]}\|A(S(U,t))\| > 0$. Then for all compact sets B there exist constants $K_1(B,T), K_2(B,T)$ such that for all initial data $U \in B$, all $\Delta t_{\text{init}} > 0$, and all sufficiently small τ,*

$$E_n = \|u(t_n) - U_n\| \le K_1 \left(\frac{\tau}{\delta^{\frac{1}{p}}}\right)^{\frac{p}{p-1}}, \tag{3.15}$$

$$U_n \in J \quad \text{and} \quad \|AU_n\| \ge \delta \quad \forall n : t_n \le T \tag{3.16}$$

and

$$T(U_n, \Delta t_n) \le K_2 \Delta t_n \tau^{\frac{p}{p-1}} \tag{3.17}$$

Proof The proof is very similar to the proof of Theorem 3.3. To prove that if (3.15) holds for $n = 0,\dots,m$ then (3.16) holds for $n = 0,\dots,m$ we need only additionally prove that $\|AU_n\| \ge \delta$ for sufficiently small τ which follows since

$$\begin{aligned} \|AU_n\| &\ge \|Au_n\| - \|AU_n - Au_n\| \\ &\ge 2\delta - \|A\|\|U_n - u_n\| \\ &\ge 2\delta - \|A\| K_1 \left(\frac{\tau}{\delta^{\frac{1}{p}}}\right)^{\frac{p}{p-1}} \\ &\ge \delta \end{aligned}$$

for τ sufficiently small.

Now we proceed as before and obtain the following estimate for E_{m+1} where $T(U_m, \Delta t_m)$ is estimated using (3.10), $s/q = p/(p-1)$ and $\|AU_n\| \ge \delta$,

$$E_{m+1} \le \exp(L\Delta t_m) E_m + K_2 \Delta t_m \left(\frac{\tau}{\delta^{\frac{1}{p}}}\right)^{\frac{p}{p-1}}$$

using a similar induction to complete the proof. □

Theorems 3.3 and 3.5 show that in XEPUS mode the algorithm converges at a superlinear rate (in fact $\mathcal{O}(\tau^{\frac{p}{p-1}})$) except in the neighbourhood of equilibria where the convergence rate is only linear in τ. However, we note that if the algorithm (2.12)–(2.21) were modified so that the timestep sequence instead satisfied $E(U_n, \Delta t_n) \le \sigma(\tau, U_n)\|AU_n\|^{\frac{1}{p}}$ then $\mathcal{O}(\tau^{\frac{p}{p-1}})$ convergence can be obtained, uniformly with respect to the closeness of the trajectory to the origin.

3.3 Nonlinear convergence results

We start by defining

$$
\begin{aligned}
B_1(u) &= \frac{1}{p!}\left[\beta_p^{(1)}(u,0) - \beta_p^{(2)}(u,0)\right], \\
B_2(u,t) &= \int_0^1 \frac{(1-s)^p}{p!}\left[\beta_{p+1}^{(1)}(u,ts) - \beta_{p+1}^{(2)}(u;ts)\right] ds, \\
b_1(u) &= \left\{ \begin{array}{cc} B_1(u)/\|f(u)\|, & \|f(u)\| \neq 0 \\ w, & \|f(u)\| = 0, \end{array} \right\} \\
b_2(u,t) &= \left\{ \begin{array}{cc} B_2(u,t)/\|f(u)\|, & \|f(u)\| \neq 0 \\ w, & \|f(u)\| = 0 \end{array} \right\}
\end{aligned}
$$

where $w \in \mathbb{R}^m$ is an arbitrary choice of non-zero vector since $b_1(u)$ and $b_2(u,t)$ may not be defined when $\|f(u)\| = 0$. Note also that by (3.1) with $r = p-1$ and (2.15),

$$
\begin{aligned}
E(u,t) &= t^q\|B_1(u) + tB_2(u,t)\| \\
&= t^q\|f(u)\|\,\|b_1(u) + tb_2(u,t)\|.
\end{aligned} \tag{3.18}
$$

The following lemma is proved as Lemma 2.4 of [14].

Lemma 3.6 *Let $J \subset \mathbb{R}^m$ be bounded and $f \in C^{p+1}(\mathbb{R}^m, \mathbb{R}^m)$. Then, if $dB_1(\bullet)$ denotes the Jacobian of $B_1(\bullet)$,*

$$
\sup_{u \in J}\|b_1(u)\| < \infty, \qquad \sup_{(u,t)\in J\times[0,D]}\|b_2(u,t)\| < \infty,
$$
$$
\sup_{(u,v,s)\in J\times J\times[0,1]}\|dB_1(su + (1-s)v)\| < \infty.
$$

Lemma 3.6 holds for general embedded Runge-Kutta pairs. We shall need the following stronger result for $(p-1,p)$ pairs with precisely p-stages.

Lemma 3.7 *Define*

$$
\tilde{b}_2(u,t) = \left\{ \begin{array}{cc} B_2(u,t)/\|f(u)\|^2, & \|f(u)\| \neq 0 \\ w, & \|f(u)\| = 0. \end{array} \right\} \tag{3.19}
$$

and let $J \subset \mathbb{R}^m$ be bounded and $f \in C^{p+1}(\mathbb{R}^m, \mathbb{R}^m)$. Then for a $(p-1,p)$ pair of explicit Runge-Kutta methods with precisely p stages,

$$
\sup_{(u,t)\in J\times[0,D]}\|\tilde{b}_2(u,t)\| < \infty.
$$

Proof We prove that $\beta_{p+1}^{(k)}(u,t),\ k = 1,2$, and hence $B_2(u,t)$, are $\mathcal{O}(\|f(u)\|^2)$.

From Lemma 4.2.6 in [15] there exist constants $c_i(J)$ such that

$$
\|f(\eta_i(u,t))\| \le (1 + c_i t)\|f(u)\| \quad 1 \le i \le p. \tag{3.20}
$$

Therefore we need only prove that each term in $B_2(u,t)$ is $\mathcal{O}(\|f(\eta_l)\| \; \|f(\eta_m)\|)$ for some integers $l, m \quad 1 \le l, m \le p$ which for convenience we shall write as $\mathcal{O}(\|f^2\|)$.

Consider either the Runge-Kutta method (2.13) or (2.14) and define

$$\eta_{p+1}(u,t) = S^{(k)}(u,t), \quad a_{p+1,j} = b_j^{(k)}$$

where $k = 1, 2$ respectively. Then the method may be written as

$$\eta_i(u,t) = u + t \sum_{j=1}^{i-1} a_{ij} f(\eta_j(u,t)), \quad i = 1, \dots, p+1.$$

We now prove by induction that

$$\frac{d^n}{dt^n}(\eta_i) = \mathcal{O}(\|f\|^2) \quad \forall \, 1 \le i \le n \le p+1.$$

Our inductive hypothesis is that for some $1 \le I \le p$, the following two statements hold:

$$\begin{array}{lrcll} (A_I) & \forall \, 1 \le i \le I, \; \frac{d^n}{dt^n}(\eta_i) & = & \mathcal{O}(\|f\|^2) & \forall i \le n \le p+1 \\ (B_I) & \forall \, 1 \le i \le I, \; \frac{d^n}{dt^n}(f(\eta_i)) & = & \mathcal{O}(\|f\|^2) & \forall i \le n \le p+1 \end{array}.$$

Note that these hold if $I = 1$ since $\eta_1 = u, \; f(\eta_1) = f(u)$ are independent of t. We now prove that (A_I) and (B_I) imply (A_{I+1}) and (B_{I+1}) by showing that

$$\frac{d^n}{dt^n}(\eta_{I+1}) = \mathcal{O}(\|f\|^2) \quad \forall I+1 \le n \le p+1, \tag{3.21}$$

$$\frac{d^n}{dt^n}(f(\eta_{I+1})) = \mathcal{O}(\|f\|^2) \quad \forall I+1 \le n \le p+1. \tag{3.22}$$

Note that

$$\begin{aligned} \frac{d^n}{dt^n}(\eta_{I+1}) &= \frac{d^n}{dt^n}\left(t \sum_{j=1}^{I} a_{I+1,j} f(\eta_j)) \right) \\ &= t \sum_{j=1}^{I} a_{I+1,j} \frac{d^n}{dt^n} f(\eta_j)) + n \sum_{j=1}^{I} a_{I+1,j} \frac{d^{n-1}}{dt^{n-1}} f(\eta_j)). \end{aligned}$$

From (B_I) we deduce that (3.21), and hence (A_{I+1}), holds.

Furthermore,

$$\frac{d^n}{dt^n} f(\eta_{I+1})$$

is a sum of terms of the form either

$$df(\eta_{I+1}) \frac{d^n}{dt^n}(\eta_{I+1})$$

or products of at least two derivatives of η_{I+1} with respect to t. By (A_{I+1}) the first type is of $\mathcal{O}(\|f\|^2)$ and by Lemma 4.6.4 of [15],

$$\frac{d^j}{dt^j}(\eta_i) = \mathcal{O}(\|f\|) \; \forall j \ge 1$$

and so the second type of term is also $\mathcal{O}(\|f\|^2)$. Therefore (B_{I+1}) also holds. By induction so does (A_{p+1}).

Since the above argument applies to both Runge-Kutta methods,

$$\beta_{p+1}^{(k)} = \frac{d^{p+1}}{dt^{p+1}}(S^{(k)}(u,t)) = \mathcal{O}(\|f\|^2)$$

for $k = 1, 2$. By the definition of $B_2(u,t)$ and using the fact that the methods agree upto order $p-1$ it follows that

$$\|B_2(u,t)\| \leq C \max_{0\leq\tau\leq t} \|\beta_{p+1}^{(1)}(u,\tau) - \beta_{p+1}^{(2)}(u,\tau)\| = \mathcal{O}(\|f\|^2) \tag{3.23}$$

proving the lemma. $\square$

Lemma 3.7 is crucial to our analysis of the general nonlinear case as it enables us to expand the local error as

$$E(u,t) = t^q \|f(u)\| \left\| \left\{ b_1(u) + t\|f(u)\|\tilde{b}_2(u,t) \right\} \right\| \tag{3.24}$$

which in turn will provide a better upper bound on the sequence of timesteps (see Lemmas 3.10 and 3.11) than in [14].

Lemma 3.7 can also be understood in terms of rooted trees. As was demonstrated in the proof of Lemma 3.2 the only rooted trees that correspond to elementary differentials that are $\mathcal{O}(\|f(u)\|)$ and not $\mathcal{O}(\|f(u)\|^2)$ are the branchless trees. Thus the result that no elementary differential of order $n > p$ of this form appears in $\beta_n(u,0)$, $n > p$ for a p-stage explicit Runge-Kutta method proves Lemma 3.7 when f is analytic since the remainder term in (3.1) can be removed. The above proof, by examining $\beta_{p+1}(u,t)$ appearing in the remainder term, relaxes the smoothness requirement.

Lemma 3.7 fails to hold in general for $(p-1,p)$ pairs that use $\nu > p$ stages since elementary differentials corresponding to branchless trees may then appear in $\beta_n(u,0)$ for $n = p+1, \ldots, \nu$. However, by imposing extra restrictions on the Runge-Kutta coefficients it may be possible to find higher-order embedded Runge-Kutta pairs in which these elementary differentials either are not reproduced or cancel so that the local error estimate can again be expanded as (3.24).

We shall also need the following lemma.

Lemma 3.8 *For $U_n \in J$, $\|f(U_{n+1})\| \leq \overline{K}\|f(U_n)\|$ for some $\overline{K}(J,D) > 0$.*

Proof Using (3.20),

$$\begin{aligned}
\|f(U_{n+1})\| &\leq \|f(U_{n+1}) - f(U_n)\| + \|f(U_n)\| \\
&\leq L\|U_{n+1} - U_n\| + \|f(U_n)\| \\
&\leq \left(L\Delta t_n \sum_i^p b_i(1 + c_i\Delta t_n) + 1 \right) \|f(U_n)\|
\end{aligned}$$

from (2.13) and (3.20) where L is the Lipschitz constant of f on $\bigcup_{t\in[0,D],u\in J} S^{(1)}(u,t)$. Since Δt_n is bounded above by D this completes the proof. $\square$

We define the following sets:

$$\Psi(\varepsilon) := \{u \in \mathbb{R}^m : \|b_1(u)\| \le \varepsilon\}, \quad \Gamma(\delta) := \{u \in \mathbb{R}^m : \|f(u)\| \le \delta\}$$

and, given sets $I, J \subset \mathbb{R}^m$,

$$\begin{gathered} J_{\varepsilon,\delta} = J \setminus (\Gamma(\delta) \cup \Psi(\varepsilon)), \quad I_{\varepsilon,\delta} = I \setminus (\Gamma(\delta) \cup \Psi(\varepsilon)) \\ J_\varepsilon = J \backslash \Psi(\varepsilon), \quad I_\varepsilon = I \backslash \Psi(\varepsilon). \end{gathered}$$

The following constants $C_3, \ldots, C_7$ (depending upon bounded $J \subset \mathbb{R}^m$ and D) are defined as follows:

$$C_3 = \sup\nolimits_{u \in J} \|b_1(u)\|,$$

$$C_4 = \sup\nolimits_{(u,t) \in J \times [0,D]} \|\tilde{b}_2(u,t)\|,$$

$$C_5 = \sup\nolimits_{(u,v,s) \in J \times J \times [0,1]} \|dB_1(su + (1-s)v)\|,$$

$$C_6 = Lip\{f(u), u \in J\},$$

$$C_7 = \sup\nolimits_{(u,t) \in J \times [0,D]} E(u,t)/t^q.$$

These constants are finite by Lemmas 3.6, 3.7, the smoothness of $f(u)$ and (3.18).

We now prove that if the numerical solution sequence U_n remains inside some bounded set J then at each step the timestep selection mechanism must terminate after a finite number of rejections; also there is a lower bound on Δt_n for all timesteps except the first and the last. Together these results ensure that the algorithm terminates in finite time.

Lemma 3.9 *Let $J \subset \mathbb{R}^m$ be compact and consider the sequence $(U_n, \Delta t_n)$ generated by the algorithm (2.12)–(2.21) for some fixed τ. Then for all $U_n \in J$, $\exists k = k(J,D) : E(U_n, \Delta t_n^{(k)}) \le \sigma(\tau, U_n)$. Furthermore assume that $U_n \in J \ \forall n : t_n < T$. Then for all sufficiently small τ there exists an integer $N = N(\tau) \le [T/(\theta\phi)] + 2$ such that $t_N = T$ and for $n = 1, \ldots, N-2$,*

$$\Delta t_n \ge \theta\phi \tag{3.25}$$

where $\phi = (\tau C_1/C_7)^{\frac{1}{q}}$.

Proof Let us fix $U_n \in J$. Note that if the timestep $\Delta t_n^{(j)}$ is rejected then

$$\Delta t_n^{(j+1)} \le \theta \Delta t_n^{(j)}$$

so that, eventually, $\Delta t_n^{(j)} \le \phi$ for some j. But if $t \le \phi$, then

$$E(U_n, t) \le t^q C_7 \le C_1 \tau \le \sigma(\tau, U_n).$$

Therefore at each step, the timestep selection scheme must terminate and the maximum possible number of rejected timesteps is $[(\ln\phi - \ln D)/\ln\theta)]$ where $[\bullet]$ denotes the integer part.

Now, $\forall n : 0 < t_n < T$ and $j \geq 1$,

$$\begin{aligned}
\Delta t_n^{(j)} &= \min(\theta \left(\frac{\sigma(\tau, U_n)}{E(U_n, \Delta t_n^{(j-1)})} \right)^{\frac{1}{q}} \Delta t_n^{(j-1)}, D, T - t_n) \\
&\geq \min(\theta \left(\frac{\sigma(\tau, U_n)}{C_7} \right)^{\frac{1}{q}}, D, T - t_n) \\
&\geq \min(\theta \left(\frac{\tau C_1}{C_7} \right)^{\frac{1}{q}}, D, T - t_n) \\
&= \min(\theta\phi, D, T - t_n)
\end{aligned}$$

from the definitions of C_1 and C_7. Similarly, for $j = 0$,

$$\begin{aligned}
\Delta t_n^{(0)} &= \min(\theta \left(\frac{\sigma(\tau, U_{n-1})}{E(U_{n-1}, \Delta t_{n-1})} \right)^{\frac{1}{q}} \Delta t_{n-1}, D, T - t_n) \\
&\geq \min(\theta\phi, D, T - t_n)
\end{aligned}$$

Therefore we deduce that

$$\Delta t_n \geq \min(\theta\phi, D, T - t_n) \;\; \forall n : 0 < t_n < T.$$

We now choose τ small enough so that $\theta\phi \leq D$ which proves the lower bound on the all the accepted timesteps apart from possibly the first and the last. The upper bound on the total number of timesteps $N(\tau)$ follows immediately. $\square$

We now use the timestep selection mechanism to inductively obtain bounds from above on the sequence of accepted timesteps.

Lemma 3.10 *Let $u \in J_\varepsilon$. Then all $t \in \mathbb{R}^+$ satisfying*

$$E(u, t) \leq \sigma(\tau, u) \;\text{ and }\; t \in \left(0, \frac{\varepsilon}{2C_4\|f(u)\|}\right]$$

also satisfy

$$t^q \leq \frac{2\sigma(\tau, u)}{\|f(u)\|\varepsilon}.$$

Proof Note that if $f(u) = 0$ then the result is vacuously true. Otherwise if $E(u, t) \leq \sigma(\tau, u)$ and $u \in J_\varepsilon$ then (3.18) and Lemma 3.7 give

$$t^q \left\| \left(b_1(u) + t\|f(u)\|\tilde{b}_2(u, t) \right) \right\| \leq \frac{\sigma(\tau, u)}{\|f(u)\|}.$$

But

$$\begin{aligned}
\left\| \left(b_1(u) + t\|f(u)\|\tilde{b}_2(u, t) \right) \right\| &\geq \|b_1(u)\| - t\|f(u)\|\|\tilde{b}_2(u, t)\| \\
&\geq \varepsilon - C_4 t\|f(u)\| \\
&\geq \varepsilon/2.
\end{aligned}$$

Thus $\varepsilon t^q \leq \frac{2\sigma(\tau,u)}{\|f(u)\|}$ and the result follows. $\square$

Lemma 3.11 *Assume that there exist integers $M = M(\tau)$ and $N = N(\tau)$ such that $U_n \in J_\varepsilon$ for $n = M, \ldots, N$ where $(U_n, \Delta t_n)$ are generated by (2.12)–(2.21). Then if*

$$\Delta t_M \leq \frac{\varepsilon}{2C_4\|f(U_M)\|} \tag{3.26}$$

it follows that for τ sufficiently small,

$$\Delta t_n^q \leq \frac{2\sigma(\tau, U_n)}{\|f(U_n)\|\varepsilon} \quad \forall n = M, \ldots, N. \tag{3.27}$$

Proof If $f(U_n) = 0$ for some n then $f(U_m) = 0 \ \ \forall m \geq n$ so that the result clearly holds for all subsequent steps; hence we assume $f(U_n) \neq 0 \quad \forall M \leq n \leq N$. Note that (2.20) implies that $\Delta t_n \leq \Delta t_n^{(0)}$ so it suffices to prove

$$\Delta t_n^{(0)} \leq \frac{\varepsilon}{2C_4\|f(U_n)\|} \quad \forall n = M, \ldots, N \tag{3.28}$$

and then apply Lemma 3.10. We use induction. Clearly (3.27) holds for $n = M$ by assumption. Assume that (3.27) holds for $n = p \leq N - 1$. Then

$$\Delta t_{p+1}^{(0)} \quad \leq \quad \theta \left(\frac{\sigma(\tau, U_p)}{\|f(U_p)\| \ \left\| \left(b_1(U_p) + \Delta t_p \|f(U_p)\| \tilde{b}_2(U_p, \Delta t_p) \right) \right\|} \right)^{\frac{1}{q}}.$$

Now, as in the proof of Lemma 3.10

$$\begin{aligned} \left\| \left(b_1(U_p) + \Delta t_p \|f(U_p)\| \tilde{b}_2(U_p, \Delta t_p) \right) \right\| \quad &\geq \quad \|b_1(U_p)\| - \Delta t_p \|f(U_p)\| \|\tilde{b}_2(U_p, \Delta t_p)\| \\ &\geq \quad \varepsilon - C_4 \Delta t_p \|f(U_p)\| \\ &\geq \quad \varepsilon/2. \end{aligned}$$

Therefore,

$$\begin{aligned} \Delta t_{p+1}^{(0)} \quad &\leq \quad \theta \left(\frac{2\sigma(\tau, U_p)}{\varepsilon\|f(U_p)\|} \right)^{\frac{1}{q}} \\ &\leq \quad \left(\frac{2C_2\tau}{\varepsilon\|f(U_p)\|} \right)^{\frac{1}{q}} \\ &\leq \quad \left(\frac{2C_2\overline{K}\tau}{\varepsilon\|f(U_{p+1})\|} \right)^{\frac{1}{q}} \quad \text{by Lemma 3.8} \\ &\leq \quad \frac{\varepsilon}{2C_4\|f(U_{p+1})\|} \end{aligned}$$

where the last inequality holds for sufficiently small τ since $q \geq 1$ and $\|f(u)\|$ is bounded on J_ε. □

Note that the accepted timesteps Δt_n are not bounded above by a constant (as in the analysis of [14]) but instead by an inverse power of $\|f(U_n)\|$.

The bound on the timestep sequence given by Lemma 3.11 must hold with $M = 0$ in order to obtain convergence results for the method. That is, the first accepted timestep Δt_0 must satisfy (3.26). This point is discussed further in §4.1 for the choice of initial timestep estimate Δt_{init} used by MATLAB Version 4.2 ode23.

Before proving the main convergence result we must make the following assumption.

Assumption 1 *There exist $\delta^*, \varepsilon^* > 0$ such that $\forall u \in J,\ \|f(u)\| < \delta^* \Rightarrow \|b_1(u)\| > \varepsilon^*$.*

Assumption 1 holds under very general conditions, for example those given in the following lemma.

Lemma 3.12 *Suppose that the compact set J contains finitely many equilibrium points $\overline{U}_k$, $k = 1, \ldots, N$ and on some neighbourhood of each $\overline{U}_k$, $f(u) = A_k(u - \overline{U}_k) + \mathcal{O}(\|u - \overline{U}_k\|^2)$ and $df(u) = A_k + \mathcal{O}(\|u - \overline{U}_k\|)$. Assume that each A_k is invertible and also that the lower-order Runge-Kutta method does not increase its order on linear problems. Then Assumption 1 holds.*

Proof Let us consider arbitrary $u \neq \overline{U}_k$ in a small neighbourhood of some $\overline{U}_k$ and without loss of generality set $\overline{U}_k = 0$. Then $B_1(u)$ is a linear combination of the elementary differentials of order p and the only elementary differential which is $\mathcal{O}(\|f(u)\|)$ and not $\mathcal{O}(\|f(u)\|^2)$ corresponds to the branchless rooted tree with p nodes. Thus we may write the i^{th} component of $B_1(u)$ as

$$[B_1(u)]_i = \alpha f^i_{j_1} f^{j_1}_{j_2} \ldots f^{j_{p-2}}_{j_{p-1}} f^{j_{p-1}} + \mathcal{O}(\|f(u)\|^2). \tag{3.29}$$

Note that the coefficient $\alpha \neq 0$ or else the lower-order method would increase its order on linear problems. Now, under the assumptions on f and the Jacobian df at $\overline{U}_k$, we can write

$$[B_1(u)]_i = \alpha A_{i,j_1} A_{j_1,j_2} \ldots A_{j_{p-2},j_{p-1}} A_{j_{p-1},j_p} u_{j_p} + \mathcal{O}(\|u\|^2).$$

If all the components of the first term vanish for some $u \neq 0$ then $A^p u = 0$ and A cannot be invertible. Therefore (3.29) cannot vanish and there exists a $\delta^*_k, \varepsilon^*_k$ such that Assumption 1 holds on some sufficiently small neighbourhood of each $\overline{U}_k$. Since there are only finitely many equilibria and the set J is closed, there exists δ^*, ε^* such that Assumption 1 holds. □

The condition that the lower-order method does not increase its order on linear problems is of course a very weak one — those methods that do increase their order on linear problems form a small subclass of explicit Runge-Kutta methods. However, if this condition is not met then $\alpha = 0$ and $\|B_1(u)\| = \mathcal{O}(\|f(u)\|^2)$ and so $b_1(u) = 0$ whenever $f(u) = 0$ contradicting Assumption 1 even under the general conditions of Lemma 3.12. If this condition on the lower-order method is satisfied Assumption 1 may fail to hold if equilibria in J are not isolated or the Jacobian A at some equilibrium point is singular. These cases will not occur for typical vector fields f. In the following convergence results we shall explicitly require that Assumption 1 holds but it should be remembered that this is, in the generic case, equivalent to choosing a lower-order method that does not increase its order on linear problems.

We can now prove the main convergence result.

Theorem 3.13 *Consider the numerical approximation of (1.1) over the time interval $[0, T]$ generated by the algorithm (2.12)–(2.21) in XEPS, EPS, XEPUS or EPUS mode. Assume* *that $S(B,t) \subseteq I_{2\varepsilon'}$, I bounded, for $t \in [0,T]$, that Assumption 1 holds and that*

$$\Delta t_0 \leq \frac{\varepsilon}{2C_4 \|f(U_0)\|}$$

where $\varepsilon = \min(\varepsilon^*, \varepsilon')$. Then there exist constants $K_1 = K_1(B,T), K_2 = K_2(B,T)$ such that for all sufficiently small τ and all $U \in B$,

$$E_n = \|u(t_n) - U_n\| \leq K_1 \left(\frac{\tau}{\varepsilon}\right)^{\gamma}, \tag{3.30}$$

$$U_n \in J_\varepsilon \quad \& \quad \Delta t_n^q \leq \frac{2\sigma(\tau, U_n)}{\|f(U_n)\|\varepsilon}, \tag{3.31}$$

and

$$T(U_n, \Delta t_n) \leq K_2 \Delta t_n \left(\frac{\tau}{\varepsilon}\right)^{\gamma} \tag{3.32}$$

where $\gamma = 1$ for XEPS, EPUS and XEPUS modes and $\gamma = (p-1)/p$ for EPS mode.

Proof We prove first that if (3.30) holds for $n = 0, \ldots, m$ then (3.31) holds for $n = 0, \ldots, m$. We have $E_n \leq K_1(\tau/\varepsilon)^\gamma$ so that for τ sufficiently small, $E_n \leq d$. Since $u_n \in I$ it follows that $U_n \in J$.

If $\|f(U_n)\| < \delta^*$ then $U_n \in J_\varepsilon$ by Assumption 1. Otherwise, if $\|f(U_n)\| \geq \delta^*$,

$$\begin{aligned}
\|B_1(U_n)\| &\geq \|B_1(u_n)\| - C_5 \|U_n - u_n\| \\
&= \|b_1(u_n)\| \|f(u_n)\| - C_5 E_n \\
&\geq \|b_1(u_n)\| \|f(U_n)\| - \|b_1(u_n)\| \|f(u_n) - f(U_n)\| - C_5 E_n \\
&\geq 2\varepsilon' \|f(U_n)\| - (C_3 C_6 + C_5) E_n.
\end{aligned}$$

We choose τ sufficiently small that

$$(C_3 C_6 + C_5) \left(\frac{K_1 \tau}{\varepsilon}\right)^{\gamma} \leq \delta^* \varepsilon'.$$

Then

$$(C_3 C_6 + C_5) E_n \leq \delta^* \varepsilon' \leq \|f(U_n)\| \varepsilon'$$

so that $\|B_1(U_n)\| \geq \varepsilon' \|f(U_n)\|$. Thus $\|b_1(U_n)\| \geq \varepsilon' \geq \varepsilon$ as required. Hence $U_n \in J_\varepsilon$ for $n = 0, \ldots, m$ if (3.30) holds for $n = 0, \ldots, m$. It follows that, if (3.30) holds for $n = 0, \ldots, m$ then the second statement in (3.31) holds for $n = 0, \ldots, m$ by Lemma 3.11. Thus, to prove (3.30) and (3.31), it is sufficient to prove that if (3.30) holds for $n = 0, \ldots, m$ then it holds for $n = m + 1$. We shall do this by proving that there exists $K_2(J), L(J) > 0$ such that

$$E_n \leq K_2 t_n \left(\frac{\tau}{\varepsilon}\right)^{\gamma} \exp(L t_n) \qquad \forall n \,:\, t_n \leq T. \tag{3.33}$$

If (3.33) holds for a given m then so does (3.30) with $K_1(J,T) = K_2(J) T \exp(LT)$, so we shall prove (3.33) by induction. Clearly (3.33) holds for $m = 0$ and

$$\begin{aligned}
E_{m+1} &= \|S(u_m; \Delta t_m) - S^{(1)}(U_m; \Delta t_m)\| \\
&\leq \|S(u_m; \Delta t_m) - S(U_m; \Delta t_m)\| + \|S(U_m; \Delta t_m) - S^{(1)}(U_m; \Delta t_m)\| \\
&\leq \exp(L \Delta t_m) E_m + K \Delta t_m^{s+1} \|f(U_m)\|
\end{aligned}$$

by Lemma 3.1 and where $L(J)$ is the Lipschitz constant of f on J. We have by Lemma 3.11

$$\Delta t_m^s \leq \left(\frac{2\sigma(\tau, U_m)}{\|f(U_m)\|\varepsilon}\right)^{\frac{s}{q}} \tag{3.34}$$

and as in the proof of Theorem 3.3

$$\begin{aligned} E_{m+1} &\leq \exp(L\Delta t_m)E_m + K_3\Delta t_m \left(\frac{\sigma(\tau, U_m)}{\varepsilon}\right)^{\frac{s}{q}} \\ &\leq \exp(L\Delta t_m)E_m + K_2\Delta t_m \left(\frac{\tau}{\varepsilon}\right)^{\frac{s}{q}} \end{aligned} \tag{3.35}$$

for XEPS, EPUS and EPS mode using $s/q \leq 1$, $\|f(u)\|$ bounded on J, and by (2.16). For XEPUS mode we instead have

$$\Delta t_m^{s-1} \leq \left(\frac{2\sigma(\tau, U_m)}{\|f(U_m)\|\varepsilon}\right)^{\frac{s-1}{q}} \tag{3.36}$$

and thus

$$\begin{aligned} E_{m+1} &\leq \exp(L\Delta t_m)E_m + K_3\Delta t_m^2 \left(\frac{\sigma(\tau, U_m)}{\varepsilon}\right)^{\frac{s-1}{q}} \\ &\leq \exp(L\Delta t_m)E_m + K_3 D\Delta t_m \left(\frac{\sigma(\tau, U_m)}{\varepsilon}\right)^{\frac{s-1}{q}} \\ &\leq \exp(L\Delta t_m)E_m + K_2\Delta t_m \left(\frac{\tau}{\varepsilon}\right)^{\frac{s-1}{q}} \end{aligned} \tag{3.37}$$

since $(s-1)/q = 1$. Since $1 \leq e^x$ for all $x \geq 0$ the inductive hypothesis gives, for all modes of operation,

$$\begin{aligned} E_{m+1} &\leq K_2 \left(\frac{\tau}{\varepsilon}\right)^{\gamma} t_m \exp(Lt_{m+1}) + K_2\Delta t_m \left(\frac{\tau}{\varepsilon}\right)^{\gamma} \exp(Lt_{m+1}) \\ &= K_2 \left(\frac{\tau}{\varepsilon}\right)^{\gamma} t_{m+1} \exp(Lt_{m+1}). \end{aligned}$$

This completes the induction. We now note that (3.32) has already been proved in (3.35) and (3.37). Since J is defined by B, $K_1 = K_1(B, T)$ which completes the proof. □

Note that these error estimates are independent of the closeness of the exact trajectory to equilibria. This considerable improvement over the analysis in [14] is afforded by the special properties of the class of Runge-Kutta pairs that we consider. The details of the timestep selection mechanism appear most crucially in the proofs of Lemmas 3.10 and 3.11. In [7] these Lemmas are generalised to include other algorithms, with only very weak assumptions on the timestep selection mechanism; and the same class of Runge-Kutta pairs is shown to confer superior convergence properties close to equilibria.

By allowing the distance of the exact trajectory from equilibria to appear in the error statement, we can also obtain $\mathcal{O}(\tau^{\frac{p}{p-1}})$ for the XEPUS mode, in a direct analogue of Theorem 3.5 for the linear case.

Theorem 3.14 *Consider the numerical approximation of (1.1) over the time interval* $[0, T]$ *generated by the algorithm (2.12)–(2.21) in XEPUS mode. Assume that* $S(B, t) \subset I_{2\varepsilon', 2\delta}$*,* I *bounded, for* $t \in [0, T]$*, that Assumption 1 holds and that*

$$\Delta t_0 \leq \frac{\varepsilon}{2C_4 \|f(U_0)\|}$$

where $\varepsilon = \min(\varepsilon^*, \varepsilon')$*. Then there exist constants* $K_1 = K_1(B, T), K_2 = K_2(B, T)$ *such that for all sufficiently small* τ *and* $U \in B$*,*

$$E_n = \|u(t_n) - U_n\| \leq K_1 \left(\frac{\tau}{\delta^{\frac{1}{p}} \varepsilon} \right)^{\frac{p}{p-1}}, \tag{3.38}$$

$$U_n \in J_{\varepsilon,\delta} \quad \& \quad \Delta t_n^q \leq \frac{2\sigma(\tau, U_n)}{\|f(U_n)\|\varepsilon} \tag{3.39}$$

and

$$T(U_n, \Delta t_n) \leq K_2 \Delta t_n \left(\frac{\tau}{\delta^{\frac{1}{p}} \varepsilon} \right)^{\frac{p}{p-1}} \tag{3.40}$$

Proof The proof is very similar to the previous one. To prove that if (3.38) holds for $n = 0, \ldots, m$ then (3.39) holds for $n = 0, \ldots, m$ we need only additionally prove that $U_n \in J_\delta$ for sufficiently small τ which follows since

$$\begin{aligned}
\|f(U_n)\| &\geq \|f(u_n)\| - \|f(U_n) - f(u_n)\| \\
&\geq 2\delta - C_6 \|U_n - u_n\| \\
&\geq 2\delta - C_6 K_1 \left(\frac{\tau}{\delta^{\frac{1}{p}} \varepsilon} \right)^{\frac{p}{p-1}} \\
&\geq \delta
\end{aligned}$$

for τ sufficiently small.

Now we proceed as before and obtain the following estimate for E_{m+1},

$$\begin{aligned}
E_{m+1} &\leq \exp(L\Delta t_m) E_m + K \Delta t_m^{p+1} \|f(U_m)\| \\
&\leq \exp(L\Delta t_m) E_m + K_2 \Delta t_m \left(\frac{\tau}{\delta^{\frac{1}{p}} \varepsilon} \right)^{\frac{p}{p-1}}.
\end{aligned}$$

A similar induction to before completes the proof. □

As in the linear case, we note that if the timesteps were instead required to satisfy

$$E(u, t) \leq \sigma(\tau, u) \|f(u)\|^{\frac{1}{p}}$$

then the rate of convergence, independent of the distance of the exact trajectory from equilibria, can be increased to $\mathcal{O}(\tau^{\frac{p}{p-1}})$ for XEPUS mode.

4 Exceptional cases and probabilistic arguments

Theorem 3.13 is our main convergence result. We note that there are 3 situations in which it can fail.

1. Δt_0 is not small enough for the convergence theorems to apply.
2. The exact trajectory passes through $\Psi(0)$.
3. Assumption 1 does not hold.

The first situation is particularly relevant to the MATLAB Version 4.2 ode23 routine since $\Delta t_{\text{init}} = T/128$ will not in general lie in the interval $(0, \frac{\varepsilon}{2C_4\|f(u)\|}]$. Then if the function $E(U, \bullet)$ has zeros for $(\mathcal{O}(1))$ values of t it is possible for the initial guess Δt_{init} to be wrongly accepted. We shall consider this possibility in §4.1 and state a corollary of Theorem 3.13 for the MATLAB Version 4.2 ode23 routine.

The second situation reflects the fact that the global error bounds proved here do not hold uniformly in the neighbourhood of $\Psi(0)$. In §4.2, the results of [14] are used to calculate the probability that the terms $\varepsilon^{-\gamma}$ and $(\varepsilon\delta^{\frac{1}{p}})^{-\frac{p}{p-1}}$, appearing in the error bounds of Theorems 3.13 and 3.14 respectively, do not exceed some value R for generic functions $f(u)$ and initial conditions chosen uniformly with respect to Lebesque measure from a ball in $\mathbb{R}^m$.

The third situation, provided that the lower-order method does not increase its order on linear problems, will not occur for typical vector fields and so we shall not discuss it here.

4.1 The choice of initial timestep for MATLAB Version 4.2 ode23.

Let us first fix the integration time T (and hence Δt_{init}) and the initial data U. Then, if $E(U, \Delta t_{\text{init}}) \neq 0$, by (2.20) the step $\Delta t_0^{(0)} = \Delta t_{\text{init}}$ will rejected for sufficiently small τ and furthermore the conditions of Lemma 3.11 will be satisfied with $M = 0$ since

$$\Delta t_0 \leq \Delta t_0^{(1)} \leq \frac{\varepsilon}{2C_4\|f(U_0)\|}.$$

Therefore, if $E(U, \Delta t_{\text{init}}) \geq \kappa > 0 \; \forall U \in B$, then Theorem 3.13 holds for all sufficiently small τ. Thus we state the following corollary for MATLAB Version 4.2 ode23.

Corollary 4.1 *Consider the numerical approximation of (1.1) over the time interval* $[0, T]$ *using MATLAB Version 4.2 ode23. Assume that* $S(B, t) \in I_{2\varepsilon'}$*, I bounded, for* $t \in [0, T]$ *and that Assumption 1 holds with* $\delta^*, \varepsilon^* > 0$*. Assume also that* $E(U, \Delta t_{\text{init}}) \geq \kappa > 0 \; \forall U \in B$*. Then, for* $\varepsilon = \min(\varepsilon^*, \varepsilon')$*,*

$$\Delta t_0 \leq \frac{\varepsilon}{2C_4\|f(U_0)\|} \tag{4.1}$$

for all sufficiently small τ*. Furthermore there exists a constant* $K_1(J, T)$ *such that for all sufficiently small* τ *and* $U \in B$*,*

$$E_n = \|u(t_n) - U_n\| \leq K_1 \left(\frac{\tau}{\varepsilon}\right)$$

and

$$U_n \in J_\varepsilon \quad \& \quad \Delta t_n^p \leq \frac{2\tau \max(1, \|U_n\|_\infty)}{\|f(U_n)\|\varepsilon}.$$

We now consider the zeros of the local error estimate $E(U,t)$ and treat U and Δt_{init} in turn as random variables.

Note first that for fixed T (and hence fixed Δt_{init}), $S^{(1)}(\bullet, \Delta t_{\text{init}}) - S^{(2)}(\bullet, \Delta t_{\text{init}}) : \mathbb{R}^m \to \mathbb{R}^m$ and thus for typical vector fields f the set of initial data U for which $E(U, \Delta t_{\text{init}}) = 0$ consists of isolated points.

Now, for a given fixed U, we treat the integration time T as a random variable uniformly distributed on some interval $[T^-, T^+]$ so that, in turn, Δt_{init} is uniformly distributed on the interval $[T^-/128, T^+/128]$. Let us suppose that the dimension of the problem $m = 1$ and there exists a unique $\Delta t^* \in [T^-/128, T^+/128]$ such that $E(U, \Delta t^*) = 0$. Then, for sufficiently small τ, $E(U, \Delta t_{\text{init}}) \geq \sigma(\tau, U)$ and (4.1) holds except for Δt_{init} in some small interval containing Δt^*. In the generic case that $\frac{dE(U,t)}{dt}$ evaluated at Δt^* is nonzero the size of this interval, and hence the probability of spuriously accepting Δt_{init}, is $\mathcal{O}(\tau)$ as $\tau \to 0$.

In higher dimensions the argument is similar, but we note also that, for given initial data U, there will typically be no $\Delta t^* \geq \frac{\varepsilon}{2C_4\|f(U)\|}$ such that $E(U, \Delta t^*) = 0$. For such a Δt^* to exist would requires that all the components of $E(U, \Delta t^*)$ to vanish simultaneously. A rigorous probabilistic argument for the higher-dimensional case appears to be very complicated so we satisfy ourselves with the statement that the probability of Δt_{init} being accepted as $\tau \to 0$ is no worse than for the one-dimensional case.

Our analysis suggests two alternative strategies for selecting Δt_{init}. Firstly, we observe that as $\tau \to 0$ it seems reasonable to also require $\Delta t_{\text{init}} \to 0$. Then, for all sufficiently small τ the first accepted timestep will satisfy (4.1). A choice of Δt_{init}, consistent with Lemmas 3.10 and 3.11, would be $\Delta t_{\text{init}} = C\tau^{\frac{1}{q}}$ for some $C > 0$. A second, more pragmatic, approach is simply to choose Δt_{init} close to machine precision. Then Δt_{init} should be accepted as Δt_0 (if it is rejected then either τ is also close to machine precision or the ODE is extremely stiff and in either case the integration should probably not be allowed to continue!). The proof of Lemma 3.9 shows that, independent of the size of Δt_0, $\Delta t_1 \geq \min(\theta\phi, D, T - t_1)$ and so only a small amount of computational effort is wasted on unnecessarily small timesteps. Of course, this argument does not hold if the algorithm includes a maximum stepsize ratio α enforcing $\frac{\Delta t_{n+1}}{\Delta t_n} \leq \alpha$.

4.2 Behaviour near $\Psi(0)$

In [14] it was argued that, for a generic function $f(u)$, the set $\Psi(0)$ where b_1 disappears will consist of isolated points. Therefore, if the dimension of the problem m is greater than 1 then an exact trajectory $u(t)$, with random initial conditions chosen from a ball, will pass through the set $\Psi(0)$ with probability zero. Let us define the following quantities

$$\delta = \delta(U,T) := \inf_{0 \leq t \leq T} \|f(S(U,t))\|,$$

$$\varepsilon = \varepsilon(U,T) := \inf_{0 \leq t \leq T} \|b_1(S(U,t))\|.$$

Then, in the statement of Theorem 3.13, it is of interest to know with what probability an exact trajectory, chosen by picking an initial condition U randomly with respect to Lebesque measure from a ball in $\mathbb{R}^m$, satisfies

$$\varepsilon^{-\gamma} \leq R$$

so that the global error is bounded by $K_1 R\tau^{\gamma}$. The analysis in [14] shows that, for all sufficiently large R and dimension $m > 1$,

$$\text{Prob}\{\varepsilon^{-\gamma} \leq R\} \geq 1 - \frac{C}{R^l} \text{ where } l = \frac{m-1}{\gamma}$$

and C is a constant.

Similarly, for XEPUS mode in Theorem 3.14 we require

$$\left(\varepsilon\delta^{\frac{1}{p}}\right)^{-\frac{p}{p-1}} \leq R$$

for the global error to be bounded by $K_1 R\tau^{\frac{p}{p-1}}$. For sufficiently large R,

$$\text{Prob}\left\{\left(\varepsilon\delta^{\frac{1}{p}}\right)^{-\frac{p}{p-1}} \leq R\right\} \geq 1 - \frac{C}{R^l} \text{ where } l = \frac{m(m-1)}{mp+m-1}$$

and C is a constant.

The situation where the exact trajectory passes through $\Psi(0)$ in one dimension was studied numerically in [4] on a variety of test problems. It was observed, for a slightly different algorithm, that in the neighbourhood of $\Psi(0)$ there is a local reduction in the convergence rate and a corresponding degradation in the global convergence rate.

5 MATLAB Version 5.0 ode23

We now briefly turn our attention to the new ode23 routine supplied in MATLAB Version 5.0. This algorithm is far more sophisticated than that in MATLAB Version 4.2 and allows the user far greater control over the integration via a number of different parameters such as the initial timestep, maximum stepsize and choice of absolute and relative tolerances. The algorithm also performs interpolation and produces output at user-specified times, if so desired, and performs event detection. However, all the above are either not relevant to the timestep selection mechanism or have already been incorporated into the class of algorithms (2.12)–(2.21) analysed above.

However, there are two major differences in the timestep selection mechanism from MATLAB Version 4.2 ode23. Firstly there is a maximum timestep ratio of 5 (i.e. $\Delta t_{n+1}/\Delta t_n \leq 5$). Secondly, after the first stepsize rejection at step n, a new timestep is selected according to the asymptotic approximation (2.20) but if subsequent stepsize rejections occur at the same n then the timestep is repeatedly halved until the local error estimate satisfies the local control. Both of these differences are included in the analyses of [14, 7] and so the convergence results presented there (which are *not* uniform with respect to initial data on compact neighbourhoods of equilibria) apply to MATLAB Version 5.0 ode23 (with minor modifications in the case of [14]). (A third change is the introduction of a minimum timestep based upon the precision of the floating-point arithmetic being used. If the error-control criterion cannot be satisfied with a timestep larger than this minimum then the integration is halted. Since we are only concerned with convergence results in real arithmetic we shall not consider this further.)

The crucial difference, as far as convergence properties are concerned, is that the routine uses a different embedded Runge-Kutta pair, namely the Bogacki–Shampine (2,3) pair [10, 9, 2]. This is designed to be operated in extrapolation mode and is a FSAL method (First Same As Last) so that while the higher-order method has 3 stages, the lower-order method uses the first stage of the next step and thus in reality has 4 stages. This means that the convergence results proved here do not apply to this pair, even if used in conjunction with the timestep selection mechanism from MATLAB Version 4.2 ode23. To summarise, the best available results for MATLAB Version 5.0 ode23 are of the same type as those appearing in [14] with error constants that are not uniform in neighbourhoods of equilibria.

Because the convergence results proved in §3, compared with those in [14], are stronger only on neighbourhoods of equilibria, it is sufficient, and indeed revealing, to illustrate the difference in convergence results that can be achieved for the different versions of ode23 by studying a linear ODE. Consider the two-dimensional problem

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

This has a saddle point at the origin and exact solution $(x_0 e^t, y_0 e^{-t})$ given initial data (x_0, y_0). From the Runge-Kutta coefficients of the Bogacki–Shampine (2,3) pair and (3.12) we find that the local error estimate for MATLAB Version 5.0 ode23 is

$$E(U_n, \Delta t_n) = \frac{1}{48} \|\Delta t_n^3 A^3 (I + \Delta t_n A) U_n\| \tag{5.1}$$

where I is the 2×2 identity matrix. Note that if $\Delta t_n = 1$ then $(I + \Delta t_n A)$ in (5.1) becomes singular with nullspace given by the line $x = 0$. Thus for *any* fixed tolerance and arbitrary y_0 we can force MATLAB Version 5.0 ode23 to accept an initial timestep $\Delta t_0 = 1$ by choosing x_0 sufficiently small ($\ll \tau$). We can make the situation even worse by also arranging that the maximum timestep equals 1 so that a sequence of timesteps $\Delta t = 1$ are accepted, the timesteps only reducing when x_n becomes $\mathcal{O}(\tau)$.

The first two columns in Table 1 show the numerical results for MATLAB Version 5.0 ode23 applied to the above example over the time interval $[0, 10]$ with the relative and absolute error tolerances set to RelTol=AbsTol=10^{-3} and the initial timestep InitialStep=1. The default maximum timestep, MaxStep, is chosen to be $\frac{1}{10}$ of the integration time and is therefore also 1. The initial data is $(x_0, y_0) = (10^{-5}, 100)$. The initial timestep estimate InitialStep=1 is accepted because x_0 is sufficiently small causing an $\mathcal{O}(1)$ error. Then, due to the choice of maximum timestep, $\Delta t_n^{(0)} = 1$ is repeatedly accepted until $x = \mathcal{O}(\tau)$ at $n = 9$ where the error control recovers. These numerical results should be contrasted with those for the Version 4.2 algorithm given in the last two columns of Table 1. The initial timestep is automatically set to 10/128 and the maximum timestep is 10/16. The tolerance τ is chosen to be 10^{-3} so that the tolerance is identical to that used in the Version 5.0 integration. For this algorithm the local error is given by (see §3.2)

$$E(U_n, \Delta t_n) = \frac{1}{6} \|\Delta t_n^3 A^3 U_n\| \tag{5.2}$$

MATLAB Version 5.0 ode23		MATLAB Version 4.2 ode23	
Time	Error	Time	Error
$t_0 = 0$	$E_0 = 0$	$t_0 = 0$	$E_0 = 0$
$t_1 = 1$	$E_1 = 3.454611$	$t_7 = 1.059370$	$E_7 = 7.1254 \times 10^{-3}$
$t_2 = 2$	$E_2 = 2.422417$	$t_{13} = 2.040615$	$E_{13} = 5.3198 \times 10^{-3}$
$t_3 = 3$	$E_3 = 1.275003$	$t_{19} = 3.021860$	$E_{19} = 2.9868 \times 10^{-3}$
$t_4 = 4$	$E_4 = 0.596996$	$t_{25} = 4.003105$	$E_{25} = 1.4916 \times 10^{-3}$
$t_5 = 5$	$E_5 = 0.262272$	$t_{31} = 4.987268$	$E_{31} = 6.9828 \times 10^{-4}$
$t_6 = 6$	$E_6 = 0.110701$	$t_{36} = 5.979402$	$E_{36} = 3.5937 \times 10^{-4}$
$t_7 = 7$	$E_7 = 0.045463$	$t_{40} = 7.063932$	$E_{40} = 2.2255 \times 10^{-4}$
$t_8 = 8$	$E_8 = 0.018305$	$t_{43} = 8.186425$	$E_{43} = 1.6944 \times 10^{-4}$
$t_9 = 8.783268$	$E_9 = 0.009749$	$t_{44} = 8.656907$	$E_{44} = 2.9399 \times 10^{-4}$
$t_{10} = 9.566536$	$E_{10} = 0.022364$	$t_{47} = 9.775934$	$E_{47} = 1.2331 \times 10^{-3}$
$t_{11} = 10$	$E_{11} = 0.034693$	$t_{48} = 10$	$E_{48} = 1.5620 \times 10^{-3}$

Table 1: Numerical results for MATLAB Versions 5.0 and 4.2 ode23. The errors are measured in the ∞-norm. To enable a direct comparison, only those outputs from MATLAB Version 4.2 ode23 that closely match the output times of the Version 5.0 algorithm are given.

and, in accordance with Corollary 3.4, the algorithm works effectively and will do so for *any* choice of initial timestep.

The failure of the Version 5.0 algorithm in the above linear example was constructed by a careful choice of Δt_{init}. But the example is also representative of what may occur at intermediate timesteps during the integration of a nonlinear problem for any fixed τ. At some time during the integration, for initial data sufficiently close to the stable manifold of the equilibrium point, a computed trajectory may enter the neighbourhood of the equilibrium point with the timestep sequence increasing to $\mathcal{O}(1)$. Once this occurs, because the local error estimate (5.1) can vanish on linear problems for certain non-zero values of Δt_n, the error can no longer be bounded independently of the distance of the exact trajectory from equlibria. This possibility cannot be ruled out by the above analysis although it will be unlikely to occur for any randomly chosen set of initial data. So, although the changes in the timestep selection mechanism of Version 5.0 certainly improve the efficiency of the code for the majority of problems, the strength of the convergence results that can be proved in the neighbourhood of equilibria is limited by the Runge-Kutta pair used.

6 Conclusions

We have identified a class of algorithms, based upon the timestep selection mechanism used in the MATLAB Version 4.2 ode23 routine, for which convergence (as $\tau \to 0$) holds uniformly on neighbourhoods of equilibria. Crucially, these algorithms use explicit Runge-Kutta pairs of orders p and $p-1$ with precisely p stages to estimate the local error and so include MATLAB Version 4.2 ode23.

In fact the analysis can be extended to cover many other algorithms, under weak assumptions on the timestep selection mechanism [7]. Uniform convergence close to equilibria occurs for the same class of Runge-Kutta pairs described above. In §5 we briefly discussed the more sophisticated ode23 routine supplied with MATLAB Version 5.0. While the new routine has many benefits and is undoubtedly far more efficient for most problems, it nevertheless has demonstrably poorer convergence properties close to equilibria due to the embedded Runge-Kutta pair employed. One possible remedy, that can be applied very generally to algorithms using Runge-Kutta pairs of all orders, is to switch to a $(p-1,p)$ Runge-Kutta pair with precisely p stages whenever the numerical solution enters the neighbourhood of an equilibrium point (i.e. when $\|f(u)\|$ becomes sufficiently small). Then, even if this alternative Runge-Kutta pair is of lower-order, $\mathcal{O}(\tau^\gamma)$ global convergence can be maintained, uniform with respect to distance from equilibria, by using the alternative Runge-Kutta pair in an appropriate operating mode. The loss in overall efficiency should be small while improving the convergence properties of the algorithm near equilibria.

For algorithms that converge uniformly on neighbourhoods of equilibria, convergence may not occur, or occur at a reduced rate, only in the neighbourhood of points where the leading term of the local error estimate, scaled by $\|f(u)\|$, vanishes. (For linear problems defined by an invertible matrix, such points do not exist and so convergence occurs uniformly on all compact sets of initial data). In [8] modified adaptive algorithms are introduced which eliminate this mode of failure for generic vector-fields.

We conclude by observing that the convergence results we have derived have been stated for embedded Runge-Kutta pairs of orders p and $p-1$ with precisely p stages and such pairs only exist for $p \leq 4$. However the results may also apply to certain higher-order Runge-Kutta pairs. Consider a $(p-1,p)$ pair with $\nu > p$ stages. All we required for the results of §3 to hold was that the lower-order method does not increase its order on linear problems and that the higher-order terms of the local error estimate expansion are all $\mathcal{O}(\|f\|^2)$. If $\nu > p$ then this second condition is no longer satisfied automatically. But this can be achieved by imposing $\nu-p$ extra conditions on the coefficients of the embedded Runge-Kutta methods to ensure that the $\mathcal{O}(\|f\|)$ terms in $b_2(u,t)$ vanish. Since the number of extra conditions increases linearly with ν and the number of free Runge-Kutta coefficients increases quadratically it seems at least possible, if not likely, that Runge-Kutta pairs with $p > 4$ exist for which the convergence results presented here also hold. For a (4,5) embedded pair with 6 stages it can easily be shown that the single extra condition to be satisfied is $b_6^{(1)} = b_6^{(2)}$. However the pair used in MATLAB Version 4.2 ode45 (which uses an algorithm identical to MATLAB Version 4.2 ode23 in every other respect) does not satisfy this extra condition and so the improved convergence results do not apply. Indeed MATLAB Version 4.2 ode45 can be made to fail in the same way as the example given in §5 for MATLAB Version 5.0 ode23. A more detailed analysis of algorithms using such higher-order Runge-Kutta pairs in the neighbourhood of equilibria would be valuable.

References

[1] MATLAB. The Math Works Inc., Natick, MA.

[2] P. Bogacki and L.F. Shampine. A 3(2) pair of Runge-Kutta formulas. *Appl. Math. Lett.*, 2:1–9, 1989.

[3] J.C. Butcher. *The numerical analysis of ordinary differential equations.* Wiley, New York, 1992.

[4] M. Calvo, D.J. Higham, J.I. Montijano, and L. Rández. Stepsize selection for tolerance proportionality in explicit Runge-Kutta codes. *Advances in Computational Mathematics*, 7:361–382, 1997.

[5] G. Hall. Equilibrium states of Runge-Kutta schemes. *ACM Trans. Math. Software*, 11:289–301, 1985.

[6] D.J. Higham and A.M. Stuart. Analysis of the dynamics of local error control via a piecewise continuous residual. *BIT*, 38:44–57, 1998.

[7] H. Lamba. Dynamical systems and adaptive time-stepping ODE solvers. SCCM Technical report, Stanford University, 1998.

[8] H. Lamba and A.M. Stuart. Convergence proofs for numerical IVP software. In *Dynamics of Algorithms*, volume 118 of *IMA Volumes in Mathematics and its Applications*, pages 107–127, 1999.

[9] L.F. Shampine. *Numerical Solution of Ordinary Differential Equations.* Chapman and Hall, New York, 1994.

[10] L.F. Shampine and M.W. Reichelt. The MATLAB ODE suite. *SIAM J. Sci. Comp.*, 18:1–22, 1997.

[11] H.J. Stetter. Considerations concerning a theory for ODE solvers. Numerical Treatment of Differential Equations. Lecture Notes in Mathematics 631, Springer-Verlag, Berlin, 1976.

[12] H.J. Stetter. Tolerance proportionality in ODE codes. Proc. second Conf. on Numerical Treatment of Ordinary Differential Equations, Humboldt University, 1980.

[13] D. Stoffer and K. Nipp. Invariant curves for variable step size integrators. *BIT*, 31:169–180, 1991.

[14] A.M. Stuart. Probabilistic and deterministic convergence proofs for software for initial value problems. *Numerical Algorithms*, 14:227–260, 1997.

[15] A.M. Stuart and A.R. Humphries. *Dynamical systems and numerical analysis.* CUP, 1996.