Parallel start for explicit parallel two-step peer methods

Bernhard A. Schmitt, Rüdiger Weiner

Abstract

Explicit parallel two-step peer methods use s stages with essentially identical properties. They are quite efficient in solving standard nonstiff initial value problems and may obtain a parallel speed-up near s on s processors for expensive problems. The two-step structure requires $s-1$ initial approximations which have been computed by one-step methods in earlier versions. We now present a self-contained starting procedure using parallel Euler steps in the initial interval. Low order error terms introduced by this step are eliminated by special coefficient sets increasing the order to s after $s-2$ time steps. An estimate for the initial stepsize is discussed, as well. Parallel OpenMP experiments with realistic problems demonstrate the efficiency compared to standard codes.

Keywords. Explicit peer methods, nonstiff ODE systems, parallel start.

1 Introduction

Explicit parallel peer methods approximate the solution of the initial value problem

$$y'(t) = f(t, y(t)), \quad y(t_{st}) = y_0 \in \mathbb{R}^n \tag{1}$$

by a time stepping scheme using s stages $Y_{mi} \cong y(t_{mi})$ per time step associated with off-step points $t_{mi} = t_m + h_m c_i$, $i = 1, \ldots, s$ of some time grid $t_0, t_1, \ldots$. In contrast to Runge-Kutta methods all these stages are peers, having the same properties like order s, for instance. Parallel execution of the time step with stepsize $h_m = t_{m+1} - t_m$ is possible since the actual stages Y_{mi}, $i = 1, \ldots, s$, depend only on the previous ones. The scheme has the form

$$Y_{mi} = \sum_{j=1}^{s} b_{ij} Y_{m-1,j} + h_m \sum_{j=1}^{s} a_{ij} f\big(t_{m-1,j}, Y_{m-1,j}\big), \quad i = 1, \ldots, s, \tag{2}$$

with coefficients a_{ij}, b_{ij} and nodes c_i. A more compact formulation is obtained by stacking the stages Y_{mi} into $Y_m := (Y_{m1}^{\mathsf{T}}, \ldots, Y_{ms}^{\mathsf{T}})^{\mathsf{T}} \in \mathbb{R}^{sn}$ and, accordingly, $F(Y_m) := \big(f(t_{m1}, Y_{m1})^{\mathsf{T}}, \ldots, f(t_{ms}, Y_{ms})^{\mathsf{T}}\big)^{\mathsf{T}}$. Then, (2) corresponds to

$$Y_m = (B_m \otimes I) Y_{m-1} + h_m (A_m \otimes I) F\big(Y_{m-1}\big) \tag{3}$$

with coefficient matrices $B_m = (b_{ij})_{i,j=1}^{s}$, $A_m = (a_{ij})_{ij,=1}^{s}$. The index m indicates that (some of) these coefficients may depend on the time step. This class

of peer methods has been introduced recently in its non-parallel form using derivatives $h_m \sum_{j<i} r_{ij} f(t_{mj}, Y_{mj})$ from the actual time step in [9] and discussed further in [6, 10]. (3) closely resembles earlier versions of General Linear Methods as in the first edition of Butcher's textbook [1]. However, Butcher was looking for a general form covering most known methods and did not specify the meaning of the variables Y_{mi} in the form we do here. Our class of peer methods yields high order approximations $Y_{mi} - y(t_{mi}) = \mathcal{O}(h_m^s)$, $i = 1, \ldots, s$, uniformly in all stages. So, dense output is available cheaply. Earlier work on two step peer methods concentrated on semi-implicit versions for stiff equations, see [5, 4, 7]. Other types of two-step methods similar to Runge-Kutta methods were developed by Jackiewicz, Tracogna [3] and others and Wright [11]. The problem of starting a two-step Runge-Kutta method with lower stage orders was addressed by Verner [8].

Due to the two-step structure with s stages all peer methods require s initial values Y_{0i} at the beginning. So far these starting values have been computed by Runge-Kutta methods like DOPRI5 [2] which also provides an estimate for the initial stepsize h_0. The purpose of the present paper is to derive a new starting procedure which does not rely on other schemes (and codes) and is parallel at the same time. The required parallelism leaves as only option a parallel Euler step of low order. Then, we introduce a scheme that eliminates low order terms by using peer methods with adapted coefficient sets for a small number of time steps after the start. The elimination strategy is developed with the aid of a polynomial representation of the peer step for linear test equations. However, we also show that this strategy works for nonlinear problems, too. Finally, we address the numerical computation of the error eliminating coefficients, present a heuristic for the choice of the initial stepsize h_0 and some realistic parallel numerical experiments in OpenMP.

Replacing the starting procedure has only a marginal effect on computing time, in general. However, a second motivation for studying a self-contained start for peer method is its relation to order control. Since the basic idea of the parallel two-step structure is that s is essentially the number of processors used, a change of order should not affect the stage number as it is common for sequential methods. Since our starting strategy increases the orders from 2 to s in a few steps we expect that similar strategies may be developed for increasing the order from a value below s in order control.

The paper continues with a short review of stability and error properties of explicit peer methods in Section 2 and shortly describes the parallel Euler starting step in Section 3. The structure of the low order errors introduced by this start and their elimination is investigated in Section 4 for linear problems. Section 5 covers nonlinear problems and the coefficients for error elimination are computed in Section 6. With an additional heuristic for the starting stepsize h_0 the whole starting procedure is described in Section 7. The parallel performance of this new peer implementation is compared in Section 8 with the old Runge-Kutta start and with DOPRI5 on a parallel machine.

2 Basic properties

For the scalar test equation $y' = \lambda y$ the scheme (3) is a simple matrix multiplication $Y_m = M_m(z)Y_{m-1}$ with the stability matrix

$$M_m(z) = B_m + zA_m, \quad z := h_m\lambda. \tag{4}$$

The uniform boundedness of solutions in the trivial test case $\lambda = 0$ (zero stability) requires the boundedness of all matrix products $B_mB_{m-1}\cdots B_k$ and may lead to severe theoretical difficulties for general matrices. We avoid these problems by choosing B fixed with *optimal zero stability* requiring

$$B_m \equiv B, \quad \det(xI - B) = x^s - x^{s-1}. \tag{5}$$

The eigenvalue one is necessary due to preconsistency (see below), but all others are chosen to be zero. However, we will have to modify this specification for the starting steps.

Order conditions are linear in A, B (cf. [9]) and depend on the stepsize ratio

$$\sigma_m := \frac{h_m}{h_{m-1}}, \quad m \geq 1, \tag{6}$$

due to the two-step structure. These stepsize ratios will be bounded above by $\sigma_m \leq \bar{\sigma}$ but not below. In fact, with the conditions

$$\mathbf{AB}(q): \quad (1+\sigma_m c_i)^k - \sum_{j=1}^{s} b_{ij}c_j^k - k\sigma_m \sum_{j=1}^{s} a_{ij}c_j^{k-1},\ i = 1,\ldots,s,\ k = 0,\ldots,q-1, \tag{7}$$

the local error in step m is of order $O(h_{m-1}^{q-1})$, i.e.

$$\begin{aligned} h_m\Delta_{mi} &:= y(t_{mi}) - \sum_{j=1}^{s} b_{ij}y(t_{m-1,j}) - h_m\sum_{j=1}^{s} a_{ij}y'(t_{m-1,j}) \\ &= \mathcal{O}(h_{m-1}^q) \end{aligned} \tag{8}$$

under $\mathbf{AB}(q)$. The first condition $\mathbf{AB}(1)$ is always assumed since it corresponds to preconsistency

$$B\mathbb{1} = \mathbb{1}, \quad \mathbb{1} = (1,\ldots,1)^{\mathsf{T}} \tag{9}$$

and requires that one is an eigenvalue of B leading to the restriction $\varrho(B) \geq 1$ for the spectral radius of B. This eigenvalue condition was already incorporated in (5). Evidently, additional order conditions $\mathbf{AB}(q)$, $q > 1$ require step dependend parameters. Since zero stability is easier accomplished for constant B, see (5), it is convenient to solve the order conditions for a σ-dependent coefficient A_m. For our purposes it has some advantages to display order conditions for a Nordsieck-type formulation in terms of Taylor coefficients of Y_m. These conditions are formulated for transformed matrices $\tilde{A}_m = V^{-1}A_mV$, $\tilde{B} = V^{-1}BV$ where V is the Vandermonde matrix $V = \left(c_i^{j-1}\right)_{i,j=1}^{s}$ associated with the nodes

c_i. Then, (9) means $\tilde{B}e_1 = e_1$ with the first unit vector e_1 and order s, i.e. $\mathbf{AB}(s+1)$, leads to the following matrix representation

$$\sigma_m \tilde{A}_m = (I + \sigma_m F) S_m P D^{-1} - \tilde{B} F D^{-1}, \tag{10}$$

with the diagonal matrices $S_m = \mathrm{diag}(1, \sigma_m, \ldots, \sigma_m^{s-1})$, $D = \mathrm{diag}(1, 2, \ldots, s)$ and with

$$P = \left(\binom{j-1}{i-1} \right)_{i,j=1}^{s}, \quad F = F_0 - \phi e_s^{\mathsf{T}} := \begin{pmatrix} 0 & 0 & \ldots & 0 \\ 1 & 0 & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \end{pmatrix} - \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_{s-1} \end{pmatrix} e_s^{\mathsf{T}}.$$

The Pascal matrix P contains the binomial coefficients and F is the Frobenius companion matrix of the node polynomial $\varphi(t) = \prod_{i=1}^{s}(t - c_i) = \sum_{j=0}^{s} \phi_j t^j$. The form (10) is equivalent with (12) in [6] since $\sigma_m F S_m (D^{-1}P - PD^{-1}) = SPD^{-1} - e_1 \mathbb{1}^{\mathsf{T}} D^{-1}$.

Now, a peer method of order s is fully determined by choosing the nodes c_i and the fixed matrix B with the restrictions (5), (9). Such methods have been constructed in [9, 6] with the additional aims of good stability, small error constants and moderate coefficient magnitudes. An additional topic was an order increase to $s+1$ by a certain superconvergence property [10]. However, in all implementations so far, the estimate for the initial stepsize and the required starting values Y_{0i} have been computed by Runge-Kutta methods.

3 Parallel start

If the main features of the peer method (2) are to be preserved in the computation of starting values Y_{0i}, $i = 0, \ldots, s$, parallelism and explicit step, there is hardly an alternative to a parallel Euler step

$$Y_{0i} := y_0 + c_i h_0 f(y_0), \quad i = 1, \ldots, s. \tag{11}$$

In principle, only $s-1$ approximations have to be computed in (11) by fixing the first grid point t_0 such that $t_{st} = t_0 + h_0 \min_i c_i$ holds. This also prevents f-evaluations to the left of t_{st}. However, we will use (11) with $t_0 = t_{st}$ in order to simplify the error analysis. Of course, the error introduced by the Euler step is of order $\mathcal{O}(h_0^2)$ only and it is the main purpose of this paper to design methods to eliminate this large initial error again in few time steps.

The precise form of the error introduced by (11) is

$$h_0 \Delta_{0i} = y(t_{0i}) - y(t_0) - c_i h_0 y'(t_0) = \sum_{k=2}^{q-1} \frac{h_0^k}{k!} y^{(k)}(t_0) c_i^k + \mathcal{O}(h_0^q)$$

or in compact form with $\mathbf{c}^k := (c_i^k)_{i=1}^s$

$$h_0 \Delta_0 = \sum_{k=2}^{q-1} h_0^k \mathbf{c}^k \otimes w_k + \mathcal{O}(h_0^q), \quad w_k := \frac{y^{(k)}(t_0)}{k!}. \tag{12}$$

We consider first the propagation of the errors $E_m := Y_m - \mathbf{y}_m$, $\mathbf{y}_m^{\mathsf{T}} := \big(y(t_{mi})^{\mathsf{T}}\big)_{i=1}^{s}$ in the first few steps. Restricting the discussion to autonomous problems from now on we replace function increments through

$$f\big(y(t_{mi}) + E_{mi}\big) - f\big(y(t_{mi})\big) = J_{mi}E_{mi} \tag{13}$$

where J_{mi} is a mean value of derivatives f_y. Hence, $F(\mathbf{y}_m + E_m) - F(\mathbf{y}_m) = J_m E_m$, where J_m is a block diagonal matrix of the form $J_m = I \otimes f_y(t_m, y(t_m)) + \mathcal{O}(h_m)$. In the short start phase we will allow varying coefficients B_m. So, the first three errors are

$$\begin{aligned}
E_0 &= -h_0\Delta_0 \\
E_1 &= -h_1\Delta_1 - h_0((B_1 \otimes I)\Delta_0 + h_1(A_1 \otimes I)J_0\Delta_0) \\
E_2 &= -h_2\Delta_2 - h_1((B_2 \otimes I)\Delta_1 + h_2(A_2 \otimes I)J_1\Delta_1) \\
&\quad -h_0((B_2B_1 \otimes I)\Delta_0 + h_1(B_2A_1 \otimes I)J_0\Delta_0) \\
&\quad -h_0h_2(A_2 \otimes I)J_1((B_1 \otimes I)\Delta_0 + h_1(A_1 \otimes I)J_0\Delta_0).
\end{aligned}$$

We have written down this in length to motivate some decisions concerning error control. Since old errors gain one order by the multiplication with $h_m A_m$ but not in the product with B_m it may be expected to gain one order per step, yielding $E_m = \mathcal{O}(h^{m+2})$, $m \le s-2$, by choosing B_m appropriately, but only one order since B_m and A_m are coupled through order conditions. Then, one also could use initially peer methods of lower, increasing order according to $h_m\Delta_m = \mathcal{O}(h^{m+2})$, $1 \le m \le s-2$. In this case however, it would be extremely difficult to track the propagation of the different error terms $\mathbf{c}^k$ from (12) through the first intervals. Hence we decided to always use peer methods of full order s after step zero. Full order s through matrices A_m given by (10) also simplifies the elimination of terms like $B_2A_1\mathbf{c}^k$ as in the representation of E_2. In order to avoid tediuos algebraic computations we develop our strategy in a polynomial formulation for the test equation $y' = \lambda y$ and show later that it also works for general equations.

4 Polynomial error elimination

The analysis of error propagation can be simplified by considering a polynomial $p(x) = \sum_{k=1}^{s} Z_k x^{k-1}$ interpolating the stage values in stretched coordinates, i.e.

$$p(c_j) = Y_{m-1,j}, \quad j = 1, \ldots, s. \tag{14}$$

Accordingly, $r(x)$ will denote the polynomial of degree $s-1$ interpolating $r(c_j) = Y_{mj}$, $j = 1, \ldots, s$. Throughout the next sections we will use $x \in \mathbb{R}$ as a dimensionless variable relativ to a given time step, e.g., $p(x) \cong y(t_{m-1} + h_{m-1}x)$. This polynomial representation yields a second description of the peer step and we will switch between algebraic and polynomial formulation and use whichever is more convenient. The multiplication $Y_{m-1} \mapsto B_m Y_{m-1}$ may be written as an operator $\mathcal{B}_m$ acting on polynomials defined by

$$(\mathcal{B}_m p)(x) = \sum_{i,j=1}^{s} b_{ij} L_i(x) p(c_j),$$

with the Lagrangian basis polynomials L_i of degree $s-1$. However, its precise form will play no role with the exception of the reproduction of constant functions $\mathcal{B}_m 1 = 1$ due to (9). We note, that $\mathcal{B}_m \varphi = 0$ by definition. The multiplication with the matrix A_m can be expressed by simple operations for high order $\mathbf{AB}(s+1)$.

Lemma 1 *Let $\tilde{A}_m$ be given by (10) and let p be the interpolating polynomial (14). Then the interpolating polynomial r of $Y_m = M_m(z)Y_{m-1}$ after one step of the method (2) applied to $y' = \lambda y$ is given by*

$$r = \mathcal{B}_m\Big(p - z\int_0^{\cdot} p(\xi)\,d\xi\Big) + z\int_0^{1+\sigma_m \cdot} p(\xi)\,d\xi - z\frac{\sigma_m^s}{s!}p^{(s-1)}\varphi \tag{15}$$

with $z = h_{m-1}\lambda$.

Proof We represent the multiplication by the different matrices in (10) as operations on polynomials $p(x) = \sum_{k=1}^s Z_k x^{k-1}$. The index m is dropped here. Shifting of Taylor coefficients and scaling by D^{-1} corresponds to integration

$$\begin{aligned}
\sum_{k,j=1}^s x^{k-1}(FD^{-1})_{kj}Z_j &= \sum_{k=1}^{s-1} x^k \frac{1}{k}Z_k - \sum_{k=1}^s x^{k-1}\phi_{k-1}\frac{1}{s}Z_s \\
&= \sum_{k=1}^s x^k \frac{1}{k}Z_k - \sum_{k=0}^s x^k \phi_k \frac{1}{s}Z_s \\
&= \int_0^x p(\xi)d\xi - \varphi(x)\frac{p^{(s-1)}}{s!},
\end{aligned}$$

with an additional error term depending on $Z_s = p^{(s-1)}/(s-1)!$. The scaled Pascal matrix SP means extrapolation, so

$$\begin{aligned}
\sum_{k,j=1}^s x^{k-1}(SPD^{-1})_{kj}Z_j &= \sum_{k,j=1}^s (\sigma x)^{k-1}\binom{j-1}{k-1}\frac{1}{j}Z_j \\
&= \sum_{k=1}^s (1+\sigma x)^{k-1}\frac{1}{k}Z_k.
\end{aligned}$$

Combining with

$$\sum_{k,j=1}^s x^{k-1}(\sigma FSPD^{-1})_{kj}Z_j = \sigma x\sum_{k=1}^s (1+\sigma x)^{k-1}\frac{1}{k}Z_k - \frac{\sigma^s Z_s}{s}\sum_{k=0}^s \phi_k x^k,$$

reveals the meaning of the multiplication by $\sigma\tilde{A}$ from (10):

$$\begin{aligned}
\sigma\sum_{k,j=1}^s x^{k-1}\tilde{a}_{kj}Z_j &= \sum_{k=1}^s (1+\sigma x)^k \frac{1}{k}Z_k - \frac{\sigma^s}{s}Z_s\varphi(x) \\
&\quad -\Big(\mathcal{B}\big(\int_0^{\cdot} p(\xi)d\xi - \frac{Z_s}{s}\varphi\big)\Big)(x) \\
&= \int_0^{1+\sigma x} p(\xi)d\xi - \Big(\mathcal{B}\int_0^{\cdot} p(\xi)d\xi\Big)(x) - \frac{\sigma^s}{s!}p^{(s-1)}\varphi,
\end{aligned} \tag{16}$$

since $\mathcal{B}\varphi = 0$. Hence, the full multiplication by $\tilde{M} = \tilde{B} + h_m\lambda\tilde{A} = \tilde{B} + z\sigma_m\tilde{A}$ corresponds to (15). $\square$

Remarks 1. The error term in (15) multiplied by $p^{(s-1)}$ reduces the degree of the integrals on the right hand side to $s-1 = \deg r$ again. In fact, both $\int_0^x p(\xi)\,d\xi - \varphi(x)p^{(s-1)}/s!$ and $\int_0^{1+\sigma_m x} p(\xi)\,d\xi - \varphi(x)\sigma_m^s p^{(s-1)}/s!$ have degree $s-1$ and are reproduced exactly by interpolation in the nodes c_i, $i = 1,\dots,s$.

2. We note that integration constants cancel out in (15) due to preconsistency (9). So, $\int_\alpha^{1+\sigma_m \cdot} p(x)\,dx - \mathcal{B}_m \int_\alpha^{\cdot} p(x)\,dx$ is independent of $\alpha \in \mathbb{R}$.

Going back to error elimination, we first consider only the test equation $y' = \lambda y$. Since the disturbing error terms in (12) are polynomials of lower degree, the correction term with $p^{(s-1)}$ in (15) vanishes. In fact, writing Δ_0 as a polynomial with $\Delta_{0i} = \Delta_0(t_0 + h_0 c_i)$, $i = 1,\dots,s$, and the monomials $\mu_k(x) := x^k$, we have

$$h_0\Delta_0(t_0 + h_0 x) = \sum_{k=2}^{s-1} h_0^k \mu_k(x) w_k + \mathcal{O}(h_0^s) \tag{17}$$

for $q = s$. So, using the same notation for the errors E_m, with $h_1\Delta_1 = \mathcal{O}(h_1^{s+1})$ we get by (15) up to $\mathcal{O}(h^s)$-terms

$$\begin{aligned} E_1(t_1 + h_1 x) &= -\sum_{k=2}^{s-1} h_0^k w_k \Big(\mathcal{B}_1\big(\mu_k - h_0\lambda \int_0^{\cdot} \mu_k\,d\xi\big) + h_0\lambda \int_0^{1+\sigma_1 x} \mu_k(\xi)\,d\xi \Big) \\ &= -\sum_{k=2}^{s-1} h_0^k w_k \Big(\mathcal{B}_1\big(\mu_k - \frac{h_0\lambda}{k+1}\mu_{k+1}\big) + \frac{h_0\lambda}{k+1}\mu_{k+1}(1+\sigma_1 x) \Big). \end{aligned}$$

No error terms $\mu_k^{(s-1)}$ appear, they vanish for $k < s-1$ and are of order $\mathcal{O}(h^s)$ for $k = s-1$. This first step already reveals the principles by which the initial errors $\mu_k(x)$ appear in subsequent steps: one contribution is evaluated in the original coordinates $x = (t-t_0)/h_0$ and mapped by $\mathcal{B}_1$ and the second contribution is just an integration evaluated in the next interval $1 + \sigma_1 x = (t-t_0)/h_0$ where $t = t_1 + h_1 x$. We note that upon integration terms gain one factor h_0, here. Since it is hard to model the integration of the terms $\mathcal{B}_1\mu_k$ in later time steps for general B_m, we simply eliminate all such terms having order below h^s by the strategy:

$$\mathcal{B}_m\mu_k\Big(\frac{t_{m-1} - t_0 + h_{m-1}\cdot}{h_0}\Big) = 0, \quad 2 \le m+1 \le k \le s-1. \tag{18}$$

The full elimination up to degree $s-1$ will also be beneficial in the nonlinear case. The conditions (18) may be scaled differently, for instance as $\mathcal{B}_m\mu_k(\tau_{m-1} + x) = 0$ with

$$\tau_{m-1} := (t_{m-1} - t_0)/h_{m-1}, \quad m \ge 1. \tag{19}$$

Together with (9) this gives the following algebraic conditions on B_m:

$$\begin{aligned} B_1\big(\mathbb{1}, \mathbf{c}^2, \mathbf{c}^3, \dots, \mathbf{c}^{s-1}\big) &= \big(\mathbb{1}, 0, \dots, 0\big), \\ B_2\big(\mathbb{1}, (\tau_1\mathbb{1} + \mathbf{c})^3, \dots, (\tau_1\mathbb{1} + \mathbf{c})^{s-1}\big) &= \big(\mathbb{1}, 0, \dots, 0\big), \\ B_3\big(\mathbb{1}, (\tau_2\mathbb{1} + \mathbf{c})^4, \dots, (\tau_2\mathbb{1} + \mathbf{c})^{s-1}\big) &= \big(\mathbb{1}, 0, \dots, 0\big), \\ &\dots \\ B_{s-2}\big(\mathbb{1}, (\tau_{s-3}\mathbb{1} + \mathbf{c})^{s-1}\big) &= \big(\mathbb{1}, 0\big). \end{aligned} \tag{20}$$

In the first step (20) almost completely defines the rank-two matrix $B_1 = (\mathbb{1}e_1^\mathsf{T} + be_2^\mathsf{T})V^{-1}$ with arbitrary $b \in \mathbb{R}^n$. With each further step the rank of B_m may grow. The concrete choice of B_m will be described later. We will show now, that the error elimination strategy (18) indeed increases the order of the errors by one at each step. In order to balance the different errors to some extent it makes sense to use a small starting stepsize h_0 and increase h_m in the following steps with higher order. This situation is considered in the formulation of the following theorem which gives the precise error structure for the test equation.

Theorem 2 *Let the peer method be applied to the equation $y' = \lambda y$ with a parallel initial Euler step (11) and stepsizes $h_0 \le h_1 \le \ldots \le h_{s-2}$. If the method uses coefficients B_m according to (18) and A_m according to (10), then the errors $E_{mi} = Y_{mi} - y(t_{mi})$, $i = 1, \ldots, s$, $0 \le m \le s-2$ are given by*

$$E_{mi} = -\lambda^m \sum_{k=m+2}^{s-1} (t_m - t_0 + h_m c_i)^k \frac{1}{k!} y^{(k-m)}(t_0) + \mathcal{O}(h_m^s). \tag{21}$$

Proof Since the map $Y \mapsto M(z)Y$ is linear and its polynomial version may be described by (15), we track the different error terms $h_0^k x^k = h_0^k \mu_k(x)$ in (17) inductively with the aid of (15) and drop terms eliminated by (18). Since $E_0 = -h_0\Delta_0$ has the asserted form we apply (15) for $m \ge 1$ with a single term $p(x) = \lambda^{m-1}(t_{m-1} - t_0 + h_{m-1}x)^k$, $k < s-1$. Hence,

$$\begin{aligned} r &= \mathcal{B}_m\Big(\lambda^{m-1}(t_{m-1} - t_0 + h_{m-1}\cdot)^k - \frac{\lambda^m}{k+1}(t_{m-1} - t_0 + h_{m-1}\xi)^{k+1}\Big|_0^\cdot\Big) \\ &\quad + \frac{\lambda^m}{k+1}(t_{m-1} - t_0 + h_{m-1}\xi)^{k+1}\Big|_0^{1+\sigma_m\cdot} \end{aligned}$$

Now $\mathcal{B}_m$ eliminates the first two terms due to (18) with the exception of the integration constant at $\xi = 0$ which is reproduced, $\mathcal{B}_m(t_{m-1} - t_0)^{k+1} = (t_{m-1} - t_0)^{k+1}$ and cancels the constant of the second integral. So, only $(t_{m-1} - t_0 + h_{m-1}(1 + \sigma_m x))^{k+1} = (t_m - t_0 + h_m x)^{k+1}$ remains and yields

$$-r(x)\frac{1}{k!}y^{(k-m+1)}(t_0) = -\frac{\lambda_m}{(k+1)!}(t_m - t_0 + h_m x)^{k+1} y^{(k+1-m)}(t_0)$$

which proves the assertion after an index shift for $k < s-1$. For $k = s-1$ the correction term with $p^{(s-1)} = (s-1)!(h_{m-1}\lambda)^{s-1}$ in (15) does not vanish. However, all terms except $\mathcal{B}_m p = 0$ are multiplied by $z = h_{m-1}\lambda$ and are of order $\mathcal{O}(h_m^s)$. $\square$

5 Nonlinear equations

For nonlinear problems the polynomial peer step (15) has to be modified since $g(x) := f\big(p(x)\big)$ is not a poynomial, in general. The correct formulation is the interpolation of g at the nodes $x = c_i$, $i = 1, \ldots, s$, and (16) is applied to this interpolating polynomial $P_{s-1}g$. Since this interpolation operator P_{s-1} reproduces polynomials up to degree $s-1$, (16) applies to polynomial error

expansions up to this degree, for instance to the sum in the representation (12) of the initial error E_0,

$$e_0(x) = -\sum_{k=2}^{s-1}(h_0 x)^k w_k + \mathcal{O}(h_0^s).$$

Accordingly, in time step m we consider $e_{m-1}(x) = p(x) - y(t_{m-1} - t_0 + h_{m-1}x)$ with p from (14) and replace the linearization (13) by a full Taylor expansion. For the initial steps with $m \le s-2$ and $h_0 \le \ldots \le h_m$ we use $y(t) - y_0 = O(h_m)$ and $e_{m-1} = O(h_m^2)$. Then, with $q = \lfloor s/2 \rfloor$ holds

$$\begin{aligned} & f\big(y(t) + e_{m-1}(x)\big) - f\big(y(t)\big) \\ &= \sum_{j=1}^{q} \frac{1}{j!} f^{(j)}(y_0)\big((y(t) + e_{m-1}(x) - y_0)^j - (y(t) - y_0)^j\big) + \mathcal{O}(h_m^s) \\ &= \sum_{j=1}^{q} \frac{1}{j!} f^{(j)}(y_0) \sum_{k=1}^{j} \binom{j}{k} \big(y(t) - y_0\big)^{j-k} e_{m-1}^k(x) + \mathcal{O}(h_m^s). \end{aligned} \tag{22}$$

Of course, $f^{(j)}(y_0)$ is a j-linear symmetric map and the powers in the last formula mean a corresponding repetition of the argument. Now, $y(t) - y_0 = \sum_{k=1}^{s-1}(t-t_0)^k w_k + O(h_m^s)$ has an expansion in powers of $t - t_0 = t_{m-1} - t_0 + h_{m-1}x$ and we will show that this holds for e_{m-1}, too. Compared to the model case of Theorem 2 nonlinearity adds additional terms of higher degree only which are cancelled by the full elimination strategy (18).

Theorem 3 *Let the peer method be applied to the equation (1) with a parallel initial Euler step (11) and stepsizes $h_0 \le h_1 \le \ldots \le h_{s-2}$. If the method uses coefficients B_m according to (18) and A_m according to (10), then its errors satisfy*

$$E_{mi} = Y_{mi} - y(t_{mi}) = \mathcal{O}(h_m^{m+2}), \quad i = 1, \ldots, s,\ 0 \le m \le s-2.$$

Proof As a generalization of (21) we will prove inductively that the errors have the form

$$e_m(x) = \sum_{k=m+2}^{s-1} (t_m - t_0 + h_m x)^k u_{mk} + \mathcal{O}(h_m^s) \tag{23}$$

with $u_{mk} \in \mathbb{R}^n$. For $m = 0$ it is known from (12). Using the error expansion (23) with index $m-1$ in the Taylor formula (22) of the difference function $g(x) = f\big(y(t) + e_{m-1}(x)\big) - f\big(y(t)\big)$, $t = t_{m-1} + h_{m-1}x$, we see that it also has an expansion in these polynomials

$$f\big(y(t) + e_{m-1}(x)\big) - f\big(y(t)\big) = \sum_{k=m+1}^{s-1} (t - t_0)^k w_{m-1,k} + \mathcal{O}(h_m^s),$$

with $w_{m-1,k} \in \mathbb{R}^n$. Applying (16) after interpolation with $\bar{g} = P_{s-1}g$ we get

$$e_m = \mathcal{B}_m\big(e_{m-1} - h_{m-1}\int_0^{\cdot} \bar{g}(\xi)d\xi\big) + h_{m-1}\int_0^{1+\sigma_m \cdot} \bar{g}(\xi)d\xi + \mathcal{O}(h_m^s).$$

Here, we have used again the fact that $h_{m-1}\bar{g}^{(s-1)} = \mathcal{O}(h_m^s)$. Since the expansion polynomials of both e_{m-1} and g start with degree $m+1$ and have the form $(t_{m-1} - t_0 + h_{m-1}x)^k$, $m+1 \le k \le s-1$, they are eliminated in the multiplication with $\mathcal{B}_m$ by (18) up to order $\mathcal{O}(h_m^s)$. The remaining terms are

$$h_{m-1}\int_0^{1+\sigma_m x}(t_{m-1} - t_0 + h_{m-1}\xi)^k d\xi = \frac{1}{k+1}(t_m - t_0 + h_m x)^{k+1}$$

and yield (23) with $u_{mk} = \frac{1}{k+1}w_{m-1,k}$. $\square$

Since the starting procedure eliminates all low order terms of a standard Taylor expansion from a certain degree upwards it is clear that the result also holds for non-autonomous problems.

6 Computing starting coefficients

The implementation of the starting strategy requires the choice of matrices B_m according to (20). Beside the starting phase the coefficient matrix B satisfying (5) is chosen in order to provide rather large stability regions for the peer method. However, due to the low order of the Euler step it should be advisible to use small starting stepsizes and so, the stability region is a minor issue while rounding errors still may count. For small h_m rounding errors are essentially proportional to $\|B_m\|$ and we may seek minimal norm solutions of (20). Considering the Frobenius norm here leads to a simple result.

Lemma 4 *The solution B_m of the condition (20), $1 \le m \le s-2$ with minimal Frobenius norm has rank one, $B_m = \mathbb{1}v_m^\mathsf{T}$, with $v_m^\mathsf{T} = e_1^\mathsf{T}R^{-1}Q^\mathsf{T}$ and the QR decomposition*

$$QR = \big(\mathbb{1}, (\tau_{m-1}\mathbb{1} + \mathbf{c})^{m+1}, \ldots, (\tau_{m-1}\mathbb{1} + \mathbf{c})^{s-1}\big), \quad Q \in \mathbb{R}^{s\times(s-m)}.$$

Proof The matrix $T := (\mathbb{1}, (\tau_{m-1}\mathbb{1} + \mathbf{c})^{m+1}, \ldots, (\tau_{m-1}\mathbb{1} + \mathbf{c})^{s-1}) = QR$ (cf. (19)) has full rank $s-m$ since it is part of a Vandermonde matrix. Hence, the unique minimal norm solution of the underdetermined system $v^\mathsf{T}T = e_1^\mathsf{T}$ can be computed with the QR decomposition: $T^\mathsf{T}v = R^\mathsf{T}Q^\mathsf{T}v = e_1 \iff Q^\mathsf{T}v = (R^\mathsf{T})^{-1}e_1$. Since the orthogonal complement of the kernel of Q^T is the range of the orthogonal matrix Q, the minimal norm solution is $v_m = Q(R^\mathsf{T})^{-1}e_1$. Since both the system $B_mT = \mathbb{1}e_1^\mathsf{T}$ and the Frobenius norm decouple for the different rows of B_m the solution for all rows is the same, $B_m = \mathbb{1}v_m^\mathsf{T}$. $\square$

Remarks 1) Although varying matrices B_m are used only for the first $s-2$ steps, one might still worry about large products $B_1 \ldots B_m$, $m \ge 1$. However, the choice of rank-one matrices according to the lemma gives an easy answer, since with $B_k = \mathbb{1}v_k^\mathsf{T}$ and $v_k^\mathsf{T}\mathbb{1} = 1$ follows

$$B_k \cdots B_m = \mathbb{1}v_m^\mathsf{T}, \quad 1 \le k \le m \le s-2.$$

2) The norm $\|B_m\|_2 = \sqrt{s}\|v_m\|_2$ still depends on the condition number of the system in Lemma 4. It is plain that this system is badly conditioned for large τ_{m-1} since all columns are nearly multiples of $\mathbb{1}$. However, in this situation h_{m-1}

is much smaller than h_0 which does not look like a sensible starting stepsize sequence. On the contrary, we will consider geometrically increasing sequences $h_m = \sigma_{start}^m h_0$, $\sigma_{start} > 1$, where

$$\tau_m = \frac{1 + \sigma_{start} + \ldots + \sigma_{start}^{m-1}}{\sigma_{start}^m} = \frac{1 - \sigma_{start}^{-m}}{\sigma_{start} - 1} \in [0, \frac{1}{\sigma_{start} - 1}), \quad m \geq 0.$$

This consideration was the reason for rescaling the conditions (18) in the form (20).
3) We note that the characteristic polynomial of explicit peer methods having a rank-one matrix B has special properties investigated in [6].

In principle, the vector v_m from Lemma 4 may be computed with effort $O(s^3)$ which is no big overhead for large problem dimensions n. Instead, we found that it is possible to use fixed starting sequences $h_m = \sigma_{start}^m h_0$ with rather large σ_{start}. In fact, we used $\sigma_{start} = 2$ for $s \leq 6$ leading to $\tau_m < 1$ and $\sigma_{start} = 3/2$ for $s = 8$ where $\tau_m < 2$. In this case, all vectors v_m can be computed once and stored in the code. Hence, the starting method has only the nodes c_i in common with the peer method used after the start.

Since the numerical experiments in Section 8 will be concerned with large problems and automatic stepsize control, the elimination strategy is demonstrated here with a small numerical example using constant stepsizes after the start. The nonlinear, non-autonomous initial value problem $y' = -ty^2$, $y(-1) = 2/3$ with the solution $y(t) = 2/(2+t^2)$ is solved on the interval $[-1, 1]$. The diagram in Figure 6 shows the error plots of runs for the six-stage method epp6 from Table 8 with an initial parallel Euler step. Then a number $i \leq s-2$ of starting steps is used with B_m from Lemma 4 and $h_m = 2^m h_0$, $1 \leq m \leq i$. After the start the computation continued with constant stepsizes, $h_m = h_i$, $m > i$. Varying i from zero to four leads to the different lines showing the expected error slopes 2 to 6 due to error elimination very nicely. The broken line belongs to a run with exact starting values $Y_{0i} = y(t_{0i})$, $i = 1, \ldots, s$, and has order 7 due to a superconvergence of this method for constant stepsizes, [10].

7 Estimating the initial step size

Starting the peer method also requires a guess for an appropriate initial stepsize h_0. Since our starting procedure eliminates the low order errors from the Euler step, the estimate for h_0 should be based on the full order peer method. The local error of the peer method is estimated in practice by comparing Y_{ms} with an embedded approximation of order $s-1$. In fact, this estimate essentially is an approximation for $h_m^s y^{(s)}(t_m)$ by a difference quotient $\nabla^{s-1} F(Y_m)$, where ∇^{s-1} corresponds to the last row $e_s^{\mathsf{T}} V^{-1}$ of the Vandermonde matrix. In the initial point t_0 the exact derivative has the form

$$y^{(s)}(t_0) = f^{(s-1)}(y_0) f_0 \cdots f_0 + \ldots + (f_0')^{s-1} f_0, \quad f_0 := f(y_0), \qquad (24)$$

where most of the elementary differentials have been omitted. Standard procedures for guessing the initial stepsize try to approximate this expression

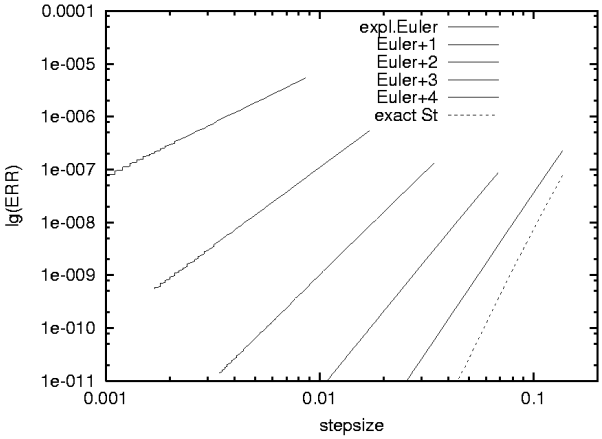

Figure 1: Varying the number of initial elimination steps

cheaply, [2]. In the beginning, where only f_0 is available, the crude estimate $\|y^{(s)}\| \cong \|f_0\|^{s-1}$ for the first differential in (24) is commonly used. Now, with a first guess h_0 based on this estimate the Euler step (11) can be performed and corresponding parallel f-evaluations $F_{0i} = f(Y_{0i})$, $i = 1, \ldots, s$ which are part of the next peer step (3). With these function values, a better approximation of (24) is at hand. Since all vectors Y_{0i} lie on one single line the difference quotient

$$\nabla^{s-1} F_0 = C h_0^{s-1} f_0^{(s-1)} f_0 \cdots f_0 + \mathcal{O}(h_0^s) \tag{25}$$

still approximates only the first elementary differential in (24), but with higher accuracy. With a first difference $\nabla F_0 := F(Y_{0s}) - f_0 \cong h_0 f_0' f_0$ also the last differential in (24) may be approximated by

$$h_0^{s-1} \|(f_0')^{s-1} f_0\| \cong \frac{\|\nabla F_0\|^{s-1}}{\|f_0\|^{s-2}}. \tag{26}$$

So with the supplemented approximation for (24) a second estimate h_0' may be obtained which replaces the first guess if $h_0' < h_0$. In the latter case however, the initial Euler step (11) has to be repeated.

This casual description lacks some practical details. First, error control usually mixes absolute and relative error criteria by using the norm

$$\|u\|_{tol} := \left(\frac{1}{n} \sum_{j=1}^{n} \left(\frac{u_i}{atol + rtol|y_i(t_0)|} \right)^2 \right)^{1/2}, \quad u \in \mathbb{R}^n, \tag{27}$$

relative to $y(t_0)$ here, with user-specified tolerances. This norm has to be combined with the usual norm $\|u\|_n := (\frac{1}{n} \sum_{j=1}^{n} u_j^2)^{1/2}$ in approximations like (26). Secondly, we will use the starting procedure of Section 6 with geometrically increasing stepsize sequence $h_m = \sigma_{start}^m h_0$. So, we will in fact guess the stepsize h_{s-2} with the estimate of $y^{(s)}(t_0)$ and reduce h_0 by the factor σ_{start}^{2-s} for the Euler step. Finally, we would like the first cheap guess based on $\|f_0\|$ alone to

be appropriate in many cases and use a small factor for guessing h_0 in order to avoid repetition of the Euler step.

Combining these steps leads to the following starting procedure with stepsize estimate:

1. Choose $h_0 := \bar{h}_0 \sigma_{start}^{2-s}$ with
$$\bar{h}_0 := \frac{1}{10} \frac{C_0}{\left(\|f_0\|_{tol} (1 + \|f_0\|_n^2)^{s/2-1} \right)^{1/s}}. \tag{28}$$
2. Perform parallel Euler step (11) with function evaluations $f(Y_{0i})$.
3. Compute $h_0' := \bar{h}_0' \sigma_{start}^{2-s}$, where according to (25), (26)
$$\bar{h}_0' := \frac{C_0 h_0}{\left(h_0 \|\nabla^{s-1} F(Y_0)\|_{tol} + \|\nabla F_0\|_{tol}^{s-1} \|f_0\|_n^{2-s} \right)^{1/s}}. \tag{29}$$
4. For $h_0' < h_0$ repeat steps 2 and 3 with $h_0 := h_0'$ once.
5. Perform $s-2$ parallel peer steps with stepsizes $h_m = \sigma_{start}^m h_0$, $m = 1, \ldots, s-2$, matrices $B_m = \mathbb{1} v_m^\mathsf{T}$ from Lemma 4 and A_m according to (10).
6. Continue using the original peer method with fixed B and standard stepsize control.

8 Numerical tests

The starting procedure was incorporated in a modified version of the FORTAN90 code used in [6] replacing the Runge-Kutta start. For ease of reading we recall some implementation details.

The computation of the coefficient matrix A_m corresponding to (10) in each time step with the stepsize ratio σ_m is implemented with the decomposition $\sigma A_m = A' + (CV)(\sigma_m S_m)(D^{-1}PV^{-1})$ in (10). The cost is $O(s^2)$ FLOPs per processor and time step and is not much overhead if the problem dimension n is large. The three fixed matrices CV, $A' = (\mathbb{1}\mathbb{1}^\mathsf{T} - BCV)D^{-1}$ and $D^{-1}PV^{-1}$ are precomputed. The last row $e_s^\mathsf{T} D^{-1} PV^{-1}$ of the last matrix corresponds to a difference quotient of order $s-1$ and so,

$$ee := e_s^\mathsf{T} D^{-1} PV^{-1} F(Y_m) \cong \frac{\sigma^s}{s} e_s^\mathsf{T} PV^{-1} \big(y'(t_{m,j})\big)_{j=1}^s \cong \frac{h^s \sigma^s}{s!} y^{(s)}(t_m)$$

can be used as error estimate in the current step. In fact, for the computation of ee the new function evaluations $F(Y_{mi})$ are performed which are lost in case of a stepsize rejection. This is different to the version used in [6], where ee was computed from the previous values $F(Y_{m-1,i})$. The error criterion mixes absolute and relative criteria in a standard way by requiring $\|ee\|_{tol} \leq 1$, cf. (27). Computations were performed on one four processor node of the MARC

Name	s	r	$\bar{\sigma}$	σ_{start}	C_0
epp4	4	0.741	1.6	2	0.3
epp6	6	0.579	1.5	2	1.0
epp8	8	0.466	1.4	1.5	0.5

Table 1: Properties of the explicit parallel peer methods

Opteron cluster at Marburg. Hence the full parallel potential of the 6- and 8-stage method could not be exploited.

The test problems are chosen with right-hand sides having different evaluation costs in order to show possible differences in parallel performance. Two well-known test problems are modified for larger dimensions in order to produce a visible computer load.

- BRUS2h, the Brusselator with the parameters from [2] but larger 200×200 grid. The diffusion constant $\nu = 10^{-5}$ is smaller in order to obtain a mildly stiff problem. The dimension is $n = 80000$.
- PLEI1t, the Pleiades problem from [2] for 7 stars in $\mathbb{R}^2$. 1000 identical copies give dimension $n = 28000$.
- MBOD4h, a celestial multi-body problem with 400 objects in $\mathbb{R}^3$ moving only under the forces of gravity from an inititial disk-shaped distribution. The dimension is $n = 2400$.

The computational cost of the f-evaluation for the Brusselator is quite low, $\cong 10n$, but the multi-body problem is rather expensive since $O(n^2)$ forces have to be computed. A reference solution for PLEI1t is known, those for BRUS2h and MBOD4h were computed by DOPRI5 with $TOL = 10^{-14}$. Each problem is solved with tolerances $rtol = atol = TOL \in \{10^{-3}, 10^{-4}, \ldots, 10^{-12}\}$.

Three explicit parallel peer methods with stage numbers $s \in \{4, 6, 8\}$ are used to solve this problem both with Runge-Kutta start and the parallel start from Section 3 and 6. Table 1 contains some properties of these methods, where r is the stability abscissa with $\varrho(M(z)) < 1$, $z \in (-r, 0] \subseteq \mathbb{R}$, stepsize increases are limited by $\sigma_m \leq \bar{\sigma}$, σ_{start} is the fixed stepsize ratio for the start and C_0 the constant in the estimates (28) and (29). The choice of C_0 was optimized by a few test runs on all three test problems. We note that only the 4- and 6-stage methods are identical with those from [6].

Each combination of methods and problems was solved for a series of tolerances with $atol = rtol = TOL \in \{10^{-2}, 10^{-3}, \ldots, 10^{-12}\}$. ERR is the final error in the scaled Euclidean norm $\left(\frac{1}{n}\sum_{j=1}^{n} e_i^2\right)^{1/2}$.

As the first topic we investigate if the parallel start developed here can replace the Runge-Kutta start including the estimate of the initial stepsize. No big savings in computing time are to be expected from the parallel start since it takes only a small fraction of the overall computation. The first three diagrams in Figure 2–4 show computing times for each of the three methods applied to all three problems. Runge-Kutta start is marked with filled symbols and parallel start with open ones. Computing times for both starting procedures are almost

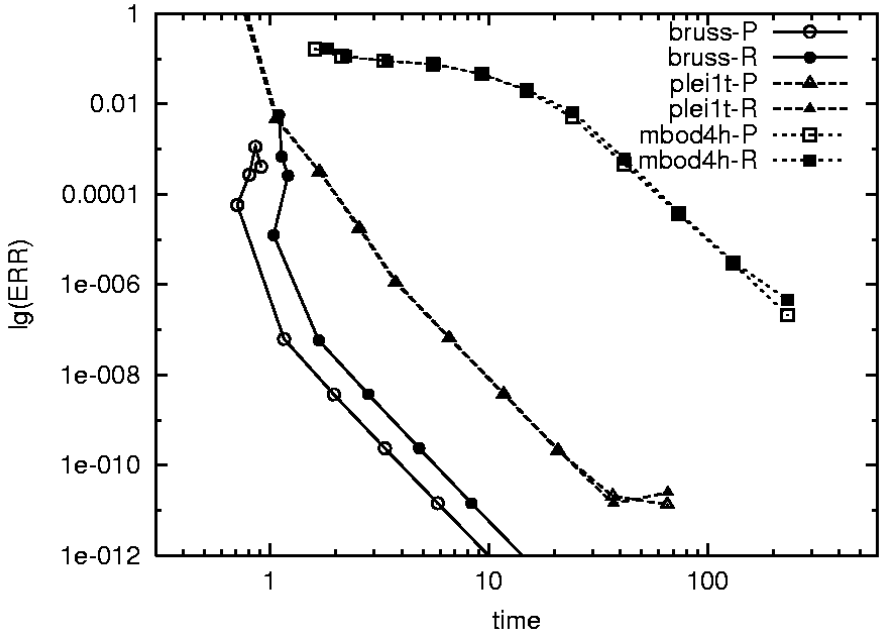

Figure 2: Parallel and Runge-Kutta start for 4-stage method

identical showing that the parallel Euler start with subsequent error elimination works efficiently. Run time differences are probably rather due to variations in the starting stepsize. In order to show that explicit parallel peer methods are also competitive we compare with the sequential Runge-Kutta code DOPRI5 having 6 (+1) stages and order 5, marked by crosses in the diagrams of Figure 5–7. For the Brusselator the parallel 4-stage method outperforms DOPRI for almost all tolerances, while the 6-stage method leads for small final errors $ERR \leq 10^{-8}$. For the many copies of the Pleiades problem the picture is similar, however, the run times of the 4-stage peer method and DOPRI get nearer for sharp tolerances. Still, the solutions of method epp4 are more accurate. Here, the 8-stage peer method is the most efficient one for $ERR \leq 10^{-5}$. The error scale for the multi-body problem in Figures 7 and 8 is different from the others since this ODE is quite sensitive to perturbations, all observed errors ERR are magnitudes larger than TOL. Here with increasing order the different peer methods outperform each other and DOPRI5 at different tolerances. Again, the peer methods epp4 and epp6 produce much more accurate solutions than DOPRI5 for the same tolerance. The problem MBOD4h is also the most interesting one with respect to parallel performance, since it has a rather expensive right hand side due to the many forces which have to be computed. For this reason Figure 8 also shows the run times of the three peer methods on one single processor marked by filled symbols. For this expensive problem the F-evaluations dominate the cost and the 4- and 8-stage methods obtain an almost optimal speed-up near 4. We note that the speed-up of the 6 and 8-stage methods may even be larger if more than 4 processors are available.

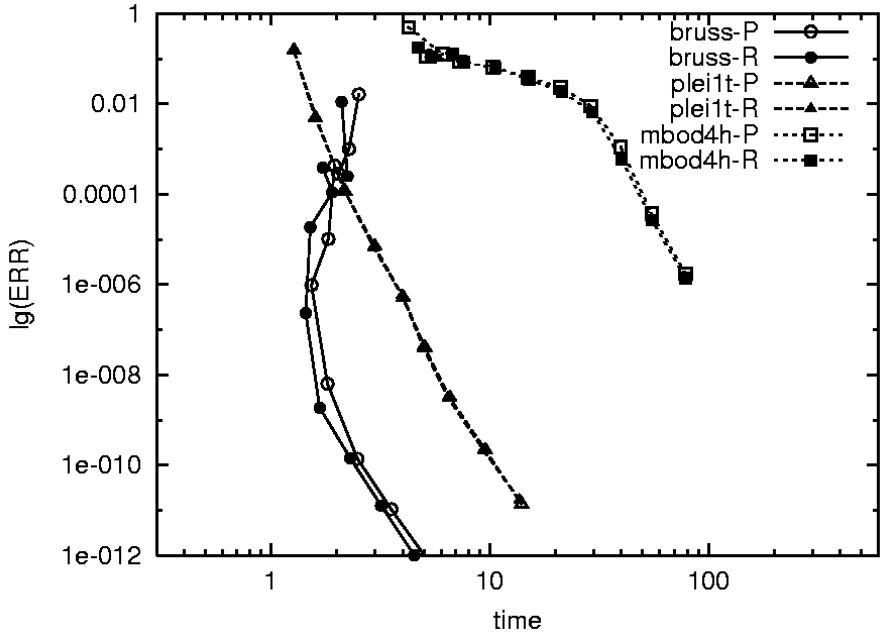

Figure 3: Parallel and Runge-Kutta start for 6-stage method

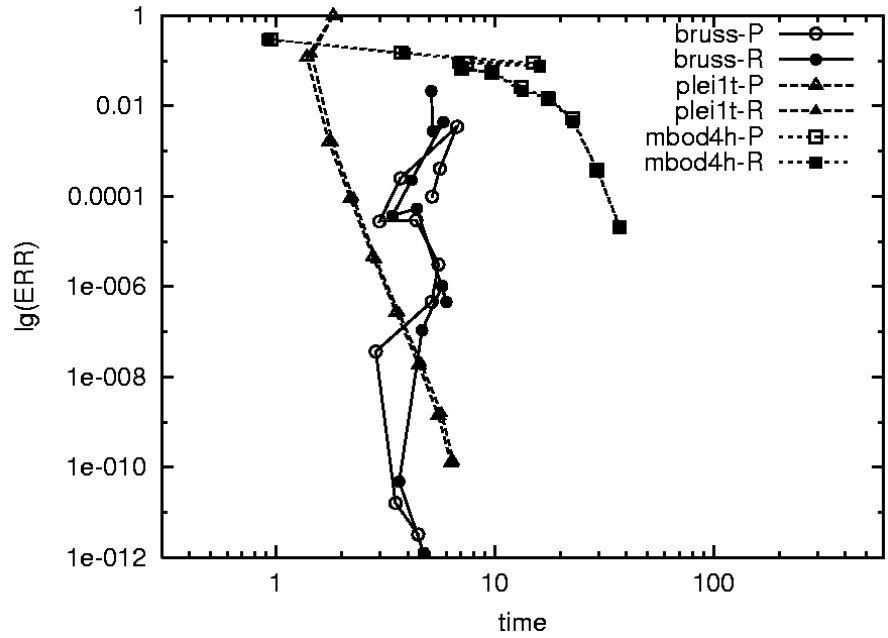

Figure 4: Parallel and Runge-Kutta start for 8-stage method

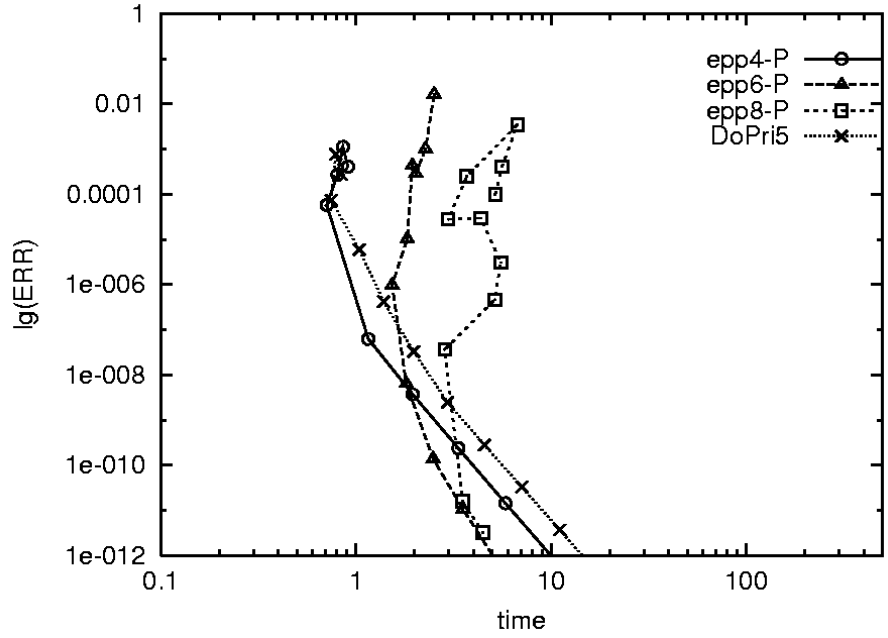

Figure 5: Run times for problem BRUS2h

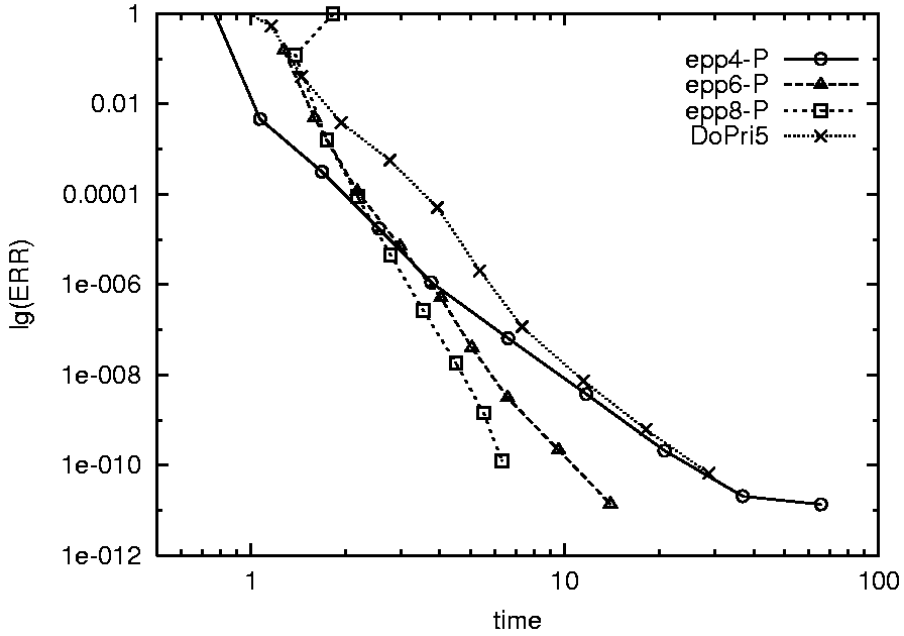

Figure 6: Run times for problem PLEI1t

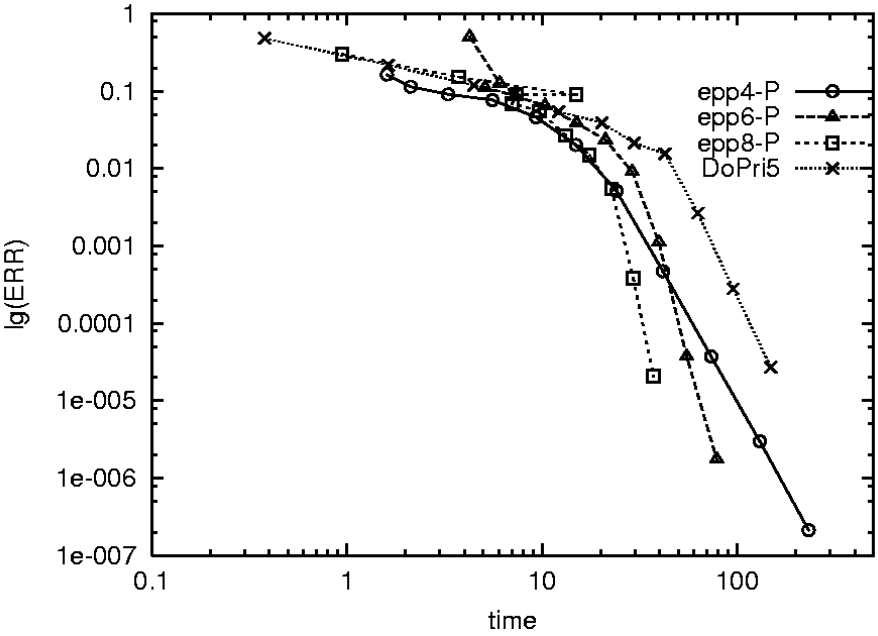

Figure 7: Peer methods and DOPRI5 for problem MBOD4h

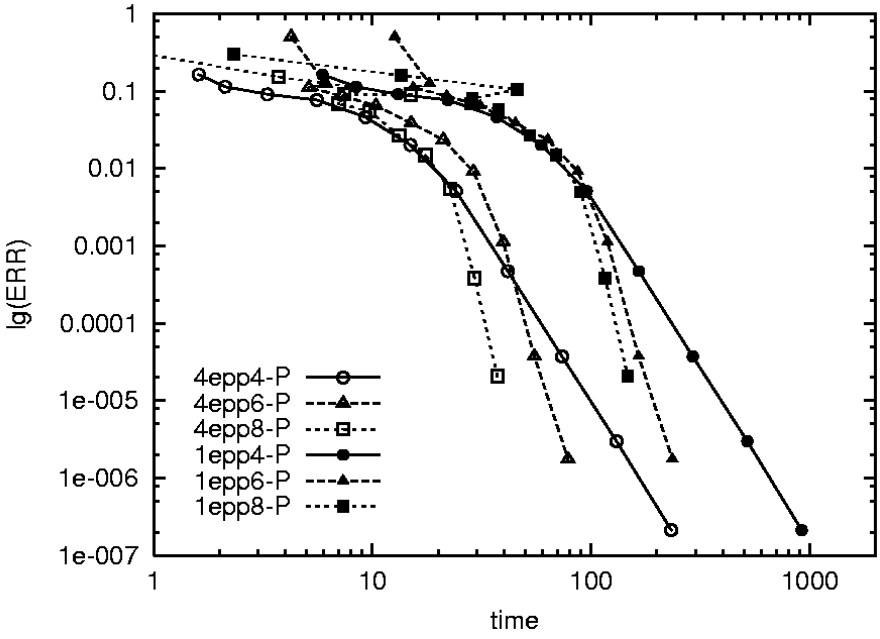

Figure 8: Peer methods for problem MBOD4h on 1 and 4 processors

9 Conclusions

We presented a parallel starting method for the class of explicit parallel peer methods which uses a parallel Euler start of low order. By variation of some method coefficients in a few steps afterwards the order is efficiently raised to s again being also the number of stages. This error elimination strategy was developed from a polynomial representation of the method being valid only for linear autonomous problems. Still it was proven to work also for nonlinear problems both in theory an practice. The order increase after the Euler start by using special parameter sets is a test case for general order control of explicit parallel peer methods using a fixed number of processors. Realistic numerical experiments demonstrate the parallel efficiency of explicit peer methods compared to standard ODE codes.

References

[1] J.C. Butcher, *The numerical analysis of ordinary differential equations (Runge-Kutta and general linear methods)*, John Wiley & Sons, 1987.

[2] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations, I. Nonstiff Problems*, second revised edition, Springer, Berlin, 1993.

[3] Z. Jackiewicz and S. Tracogna, *A general class of Two-Step Runge-Kutta methods for ordinary differential equations*, SIAM J. Numer. Anal. 32, 1390–1427 (1995).

[4] H. Podhaisky, R. Weiner, and B.A. Schmitt, *Rosenbrock-type 'Peer' two-step methods* , Appl. Numer. Math. 53, 409–420 (2005).

[5] B.A. Schmitt, and R. Weiner, *Parallel two-step W-methods with peer variables*, SIAM J. Numer. Anal. 42, 265-282 (2004).

[6] B. A. Schmitt, R. Weiner, S. Jebens, *Parameter optimization for explicit parallel peer two-step methods*, Appl. Numer. Math. (2008), in press DOI:10.1016/j.apnum.2008.03.013

[7] B.A. Schmitt, R. Weiner, and H. Podhaisky, *Multi-implicit peer two-step W-methods for parallel time integration*, BIT Numerical Mathematics 45, 197–217 (2005).

[8] J.H. Verner, *Improved starting methods for two-step Runge-Kutta methods of stage-order $p-3$*, Appl. Numer. Math. 56 82006).

[9] R. Weiner, K. Biermann, B.A. Schmitt, and H. Podhaisky, *Explicit two-step peer methods*, Comput. Math. Appl. 55, 609-619 (2008).

[10] R. Weiner, B.A. Schmitt, H. Podhaisky, and S. Jebens, *Superconvergent explicit two-step peer methods*, J. Comp. Appl. Math. (2008), in press DOI:10.1016/j.cam.2008.02.014

[11] W.M. Wright, *General Linear Methods with Inherent Runge-Kutta Stability*, PhD Thesis, The University of Auckland (2003).