Parameter optimization for explicit parallel peer two-step methods

Bernhard A. Schmitt, Rüdiger Weiner, Stefan Jebens

Dedicated to Professor Karl Strehmel for his outstanding commitment to the NUMDIFF seminar

Abstract

Due to a two-step structure certain explicit peer methods with s stages have a natural parallel implementation on s processors. By the peer property all stages have essentially identical properties and we construct a class of zero-stable methods with order $p = s$ in all stages. Two approaches are discussed for choosing the free parameters. In a certain subclass the stability polynomial depends only linearly on a new set of parameters and by employing taylored root locus bounds a linear program can be formulated and solved exactly for stable and accurate methods. The second approach uses Monte-Carlo simulation in a wider class of methods. The two approaches are compared in realistic numerical tests on a parallel computer.

Keywords Explicit two-step peer methods, parallel methods for ODEs, linear programming.

1 Introduction

Many standard time stepping schemes for the initial value problem

$$y' = f(t, y),\ t \in [t_0, t_e], \quad y(t_0) = y_0 \in \mathbb{R}^n, \tag{1}$$

compute one extraordinary solution approximation per time step with good stability and accuracy properties. The class of peer methods instead uses s different approximations in each time step having essentially identical properties. Some work has

been done on parallel peer two-step methods for stiff problems [9, 10, 11] and on sequential methods [8], but here we consider explicit parallel peer methods introduced in [12]. Other types of two-step methods similar to Runge-Kutta methods have been considered by Jackiewicz, Tracogna [7] and others and recently by Wright [13]. There exist many approaches for the construction of parallel methods for stiff and non-stiff ODEs, some of which are described in the textbook [2]. However, with the exception of the stiff solver [1] no parallel software seems to be widely available. Recently, a fresh motivation for developping parallel methods has come from the advent of multi-core processors for desktop computers. For such machines peer methods allow easy parallelization across the method with black-box implementation of the right-hand side f.

In each time step associated with the interval $[t_m, t_m + h_m]$ explicit peer methods compute s new stages Y_{mi}, $i = 1, \ldots, s$, as approximations for the solution $y(t_{mi})$ at points $t_{mi} = t_m + h_m c_i$ and these depend on the stages $Y_{m-1,i}$, $i = 1, \ldots, s$, of the previous time step only:

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

In fact, this is already the parallel version without the contribution from actual stages. Storing the stage vectors Y_{mi} and also $f(t_{mi}, Y_{mi})$ as *rows* of $s \times n$ matrices $Y_m = (Y_{mi}^{\mathsf{T}})_{i=1}^{s}$, $F_m = f(Y_m) = (f(Y_{mi})^{\mathsf{T}})_{i=1}^{s} \in \mathbb{R}^{s \times n}$, a more compact formulation is

$$Y_m = BY_{m-1} + h_m A f(Y_{m-1}) = \Big(B, A\Big) \begin{pmatrix} Y_{m-1} \\ h_m F_{m-1} \end{pmatrix} \tag{3}$$

with coefficient matrices $A = (a_{ij})_{i,j=1}^{s}$, $B = (b_{ij})_{i,j=1}^{s}$. This form shows that one step of the explicit peer method can be executed by s parallel function evaluations for $F_{m-1} = (f(Y_{m-1,i})^{\mathsf{T}})_{i=1}^{s}$ followed by one single parallel matrix multiplication, i.e. one call of the standard `gemm` routine of the highly optimized parallel BLAS library.

At the scalar test equation $y' = \lambda y$ the scheme (3) reduces to the recursion $Y_m = M(z)Y_{m-1}$. Hence, with $z = h\lambda$ the stability matrix of the scheme turns out to be

$$M(z) = B + zA, \quad z \in \mathbb{C}. \tag{4}$$

An important part of this paper discusses the characteristic polynomial of this matrix which we will simply call the *stability polynomial* of the method (2). The notation is

$$\mu(\lambda, z) = \det\left(\lambda I - M(z)\right) = \sum_{i=0}^{s} \mu_i(z) \lambda^i. \tag{5}$$

After this introduction we formulate in Section 2 the order conditions for high order s which establishes a close relation between the parameter matrices A and B in (3). The main result is presented in Section 3 where a new representation of the polynomial (5) is found for a subclass of peer methods by a certain parameter substitution. By a careful analysis of the Frobenius reduction of A it is shown that μ is linear in these new parameters. This fact is exploited in Section 4 where a linear program is formulated for constructing methods with large stability intervals and small error constants for stage numbers 3 to 8. Eigenvalue computations are avoided by the use of optimized root locus bounds leading to linear constraints. As an alternative approach Monte-Carlo simulation in a more general class of peer methods is considered in Section 5. Finally, methods from both approaches and the standard code DOPRI5 are compared in Section 6 on a parallel machine with timings from several demanding test problems.

2 High order explicit parallel peer methods

Order conditions for peer methods are derived in a standard way by considering the residual

$$h_m \Delta_{mi} = y(t_{mi}) - \sum_{j=1}^{s} b_{ij} y(t_{m-1,j}) - h_m \sum_{j=1}^{s} a_{ij} y'(t_{m-1,j})$$

at the solution y of the ODE. Since values from two different time intervals appear here the order conditions depend on the stepsize ratio

$$\sigma_m := h_m / h_{m-1} \tag{6}$$

of the two interval lengths. By Taylor expansion of Δ_{mi} the following sufficient conditions for local order $q-1$ are derived [12],

$$\begin{aligned} &\mathbf{AB}(q): \quad AB_i(k) = 0, \quad i = 1, \ldots, s, \ k = 1, \ldots, q, \\ &AB_i(k) := \ c_i^{k-1} - \sum_{j=1}^{s} b_{ij} (\tfrac{c_j - 1}{\sigma})^{k-1} - (k-1) \sum_{j=1}^{s} a_{ij} (\tfrac{c_j - 1}{\sigma})^{k-2}. \end{aligned} \tag{7}$$

So, the number of order conditions is sq. Concentrating on methods with order at least $s-1$ these conditions may be written in compact matrix form containing both coefficient matrices A and B [12]. In order to avoid theoretical difficulties with the boundedness of matrix families, see [11], it is convenient to choose B fixed for all time steps and to solve the order conditions for A. Introducing the diagonal matrices $C = \mathrm{diag}(c_1, \ldots, c_s)$, $D = \mathrm{diag}(1, 2, \ldots, s)$, $S = \mathrm{diag}(1, \sigma, \ldots, \sigma^{s-1})$, the

Vandermonde matrix $V = (c_j^{i-1})_{i,j=1}^s$ and the Pascal matrix $P = (\binom{i-1}{j-1})_{i,j=1}^s$ this gives

$$\begin{gathered} B\mathbb{1} = \mathbb{1}, \quad \mathbb{1} = (1,\dots,1)^{\mathsf{T}} \\ A = \Big(CVSD^{-1}P + \tfrac{1}{\sigma}\mathbb{1}\mathbb{1}^{\mathsf{T}}D^{-1} - \tfrac{1}{\sigma}BCVD^{-1} - \tfrac{1}{s}\sigma^{s-1}\beta e_s^{\mathsf{T}} \Big) V^{-1} \end{gathered} \tag{8}$$

for explicit parallel methods (2) of order $\geq s-1$. The choice $\beta = 0$ satisfies $\mathbf{AB}(s+1)$ and leads to a method of order s. This representation is essentially formula (8,9) from [12] since $(C-I)V_1D^{-1}V_1^{-1} = (CV - \mathbb{1}\mathbb{1}^{\mathsf{T}})D^{-1}V^{-1}$ where $V_1 = ((c_j-1)^{i-1})_{i,j=1}^s = VP^{-1}$. It follows from the identity

$$\sum_{k=1}^{j}(c_i-1)^k \frac{1}{k}\binom{j-1}{k-1} = \sum_{k=1}^{j}(c_i-1)^k \binom{j}{k}\frac{1}{j} = c_i^j\frac{1}{j}.$$

3 Linear parameters in the stability polynomial

As in [9] we introduce the transformed matrices $\tilde{A} = V^{-1}AV$, $\tilde{B} = V^{-1}BV$. Applying the same transformation to the diagonal matrix C of the nodes leads to the Frobenius matrix

$$V^{-1}CV = F = F_0 - \phi e_s^{\mathsf{T}},\ F_0 = \begin{pmatrix} 0 & & \cdots & 0 \\ 1 & 0 & & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{pmatrix},\ \phi = \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{s-1} \end{pmatrix}. \tag{9}$$

Its characteristic polynomial is the node polynomial $\phi(t) = \sum_{k=0}^s \phi_k t^k = \prod_{i=1}^s (t-c_i)$. With (9) the transformed version of condition (8) becomes

$$\begin{aligned} \tilde{B}e_1 &= e_1, \\ \tilde{A} &= FSD^{-1}P + \tfrac{1}{\sigma}e_1\mathbb{1}^{\mathsf{T}}D^{-1} - \tfrac{1}{\sigma}\tilde{B}FD^{-1} - \tfrac{1}{s}\sigma^{s-1}V^{-1}\beta e_s^{\mathsf{T}} \\ &= F_0SD^{-1}P + \tfrac{1}{\sigma}e_1\mathbb{1}^{\mathsf{T}}D^{-1} - \tfrac{1}{\sigma}\tilde{B}F_0D^{-1} - \tfrac{1}{s}\sigma^{s-1}\tilde{\beta}e_s^{\mathsf{T}} \end{aligned} \tag{10}$$

where $\tilde{\beta} := \phi + \sigma^{-s}\tilde{B}\phi + V^{-1}\beta$ and e_i is the i-th unit vector. The second form of $\tilde{A}$ introduces the parameter $\tilde{\beta}$ which is free for order $s-1$ and is fixed by $\tilde{\beta} = \phi + \sigma^{-s}\tilde{B}\phi$ for order s. So, only for order s stability properties are related to the choice of the nodes (c_i) through ϕ. We now make the important observation that the part of (10) being independent of B has Hessenberg form since the matrices P and $SD^{-1}P$ are upper triangular.

Before proceeding we collect a few relations concerning the Pascal matrix P. A shorthand notation for the Taylor theorem is $f(\cdot + h) = \exp(h\partial_t)f(\cdot)$. For polynomials p the inverse Vandermonde matrix V^{-1} transforms function values into Taylor

coefficients. For polynomial coefficients the matrix DF_0^{T} implements the derivative and the binomial theorem the shift $p(\cdot) \mapsto p(\cdot + 1)$. The algebraic version of the Taylor theorem is given in the following lemma which can be verified by direct computation.

Lemma 1 *Let $\tilde{E} := DF_0^{\mathsf{T}}$. Then the Pascal matrix satisfies $P = \exp(\tilde{E})$ and commutes with $\tilde{E}$. In fact the elements of its powers are $e_i^{\mathsf{T}} \tilde{E}^k e_j = \frac{(i+k-1)!}{(i-1)!} \delta_{i,j-k}$, $e_i^{\mathsf{T}} P^k e_j = \binom{j-1}{i-1} k^{j-i}$, $k \in \mathbb{Z}$.*

With the notation of this lemma parts of formula (10) can be greatly simplified. Removing the last column of $\tilde{A}$ containing $\tilde{\beta}$ by multiplication with the shift matrix F_0^{T} yields the extremely simple relation

$$\tilde{B} + \sigma \tilde{A} \tilde{E} = SP. \tag{11}$$

In fact, multiplying (10) with $\sigma \tilde{E}$ gives

$$\begin{aligned} \sigma \tilde{A} \tilde{E} &= S F_0 D^{-1} P D F_0^{\mathsf{T}} + e_1 \mathbb{1}^{\mathsf{T}} F_0^{\mathsf{T}} - \tilde{B} F F_0^{\mathsf{T}} \\ &= S(P - e_1 \mathbb{1}^{\mathsf{T}}) + e_1 \mathbb{1}^{\mathsf{T}} - \tilde{B} = SP - \tilde{B}. \end{aligned}$$

Here, the identities $F F_0^{\mathsf{T}} = I - e_1 e_1^{\mathsf{T}}$, $\mathbb{1}^{\mathsf{T}} F_0^{\mathsf{T}} = \mathbb{1}^{\mathsf{T}} - e_1^{\mathsf{T}}$ have been used and $F_0 D^{-1} P D F_0^{\mathsf{T}} = P - e_1 \mathbb{1}^{\mathsf{T}}$ which follows from $D^{-1} P D = \left(\frac{1}{i} \binom{j-1}{i-1} j \right) = \left(\binom{j}{i} \right)$.

Zero stability is the least stability requirement for a method and for the scheme (2) this means that B has to be power bounded. By the first condition in (10) B has one eigenvalue 1 and for optimal damping we require that all other eigenvalues be zero. In this paper we consider two such choices for B. The first one is to maintain Hessenberg structure of $\tilde{A}$ and use a fixed upper triangular matrix

$$\tilde{B} = \begin{pmatrix} 1 & \tilde{b}_{12} & \tilde{b}_{13} & \cdots & \tilde{b}_{1s} \\ & 0 & \tilde{b}_{23} & \cdots & \tilde{b}_{2s} \\ & & \ddots & & \vdots \\ & & & 0 & \tilde{b}_{s-1,s} \\ & & & & 0 \end{pmatrix}. \tag{12}$$

We immediately observe that $\tilde{B} F D^{-1}$ is upper triangular, too, and we obtain the following simple lemma.

Lemma 2 *With a matrix $\tilde{B}$ of the shape (12) the matrix $\tilde{A}$ in (10) has Hessenberg form.*

This structure carries over to the transformed stability matrix in (4). It also has Hessenberg form and the subdiagonal elements of $\tilde{A}$ and $\tilde{M}$ are $\tilde{a}_{i+1,i} = \sigma^{i-1}/i, \quad \tilde{m}_{i+1,i} = \tilde{a}_{i+1,i}z, \quad i = 1, \ldots, s-1$. Of course, the characteristic polynomial μ of $\tilde{M}$ depends nonlinearly on the matrix entries of $\tilde{B}$. However we will show now that it is possible to use the characteristic coefficients of A,

$$\det(\lambda I - A) = \sum_{i=0}^{s} \alpha_i \lambda^i, \tag{13}$$

as new parameters in order to eliminate this nonlinearity from μ for a certain subclass of methods. For this purpose the characteristic polynomial (13) is computed by the well-known transformation of the Hessenberg matrix $\tilde{A}$ to Frobenius companion form.

In the rest of this section only the case $\sigma = 1$ is discussed. We use the reduction to the form (9) and consider $F_0 - \alpha e_s^{\mathsf{T}} = U\tilde{A}U^{-1}$, $\alpha^{\mathsf{T}} = (\alpha_0, \ldots, \alpha_{s-1})$, as the Frobenius companion of $\tilde{A}$. The matrix U is upper triangular with diagonal elements $u_{ii} = (i-1)!$, $i = 1, \ldots, s$. We prefer this version above the direct computation of U^{-1} since U will have a simpler form here. The other elements of U eliminate the entries of $\tilde{A}$ above the subdiagonal. The system $U\tilde{A} - F_0U + \alpha e_s^{\mathsf{T}}U = 0$ can be solved columnwise by

$$u_{i,j+1} = j\Big(u_{i-1,j} - \sum_{k=i}^{j} u_{ik}\tilde{a}_{kj}\Big), \ 1 \le i \le j \le s-1, \tag{14}$$

where $u_{i-1,j}$ is missing for $i = 1$, of course. At the end the last column of $-U\tilde{A}U^{-1}$ contains the coefficients α. We note that U does not depend on the last column of $\tilde{A}$ and so, by (10), not on ϕ through $\tilde{\beta}$. By construction the transformation by U simplifies the structure of $\tilde{A}$. But more important is the very surprising result that it also simplifies the stability matrix M and its characteristic polynomial $\mu(\lambda, z)$. In fact, the transformed stability matrix has the form

$$U\tilde{M}(z)U^{-1} = U\tilde{B}U^{-1} + zU\tilde{A}U^{-1} = U\tilde{B}U^{-1} + z(F_0 - \alpha e_s^{\mathsf{T}}). \tag{15}$$

Due to the triangular structure of U missing lower diagonals of $\tilde{B}$ in (12) are also missing in $U\tilde{B}U^{-1}$ and lead to fewer contributions in the characteristic polynomial $\mu(\lambda, z)$.

For a triangular matrix $\tilde{B}$ as in (12) the transformation U can be characterized by a regular linear system. Multiyplying (14) with the shift F_0^{T} gives $U\tilde{A}F_0^{\mathsf{T}} - F_0UF_0^{\mathsf{T}} = 0$. The first column of this expression is zero, but we can place here

the initial condition $Ue_1 = e_1$ for the recursion in (14) leading to the linear system $U(e_1e_1^\mathsf{T} + \tilde{A}F_0^\mathsf{T}) - F_0UF_0^\mathsf{T} = e_1e_1^\mathsf{T}$. With the representation (11) and $\sigma = 1$ this becomes

$$U\Big(e_1e_1^\mathsf{T} + (P - \tilde{B})\hat{D}^+\Big) - F_0UF_0^\mathsf{T} = e_1e_1^\mathsf{T},$$

where $D^+ := \mathrm{diag}(0, 1, \frac{1}{2}, \ldots, \frac{1}{s-1})$. Multiplying from the right by the diagonal matrix $\mathrm{diag}(1, 1, 2, \ldots, s-1)$ gives $(e_1e_1^\mathsf{T} + (P - \tilde{B})\hat{D}^+)\mathrm{diag}(1, 1, 2, \ldots, s-1) = e_1e_1^\mathsf{T} + (P - \tilde{B})$ and $F_0^\mathsf{T}\mathrm{diag}(1, 1, 2, \ldots, s-1) = DF_0^\mathsf{T}$. A final multiplication with P^{-1} yields the following defining system for the upper triangular transformation U of $\tilde{A}$ to Frobenius form

$$U\Big(I - (\tilde{B} - e_1e_1^\mathsf{T})P^{-1}\Big) - F_0UDF_0^\mathsf{T}P^{-1} = e_1e_1^\mathsf{T}P^{-1}. \tag{16}$$

In principle, $e_1e_1^\mathsf{T}P^{-1}$ cancels out on both sides. However, we prefer the given form since $(\tilde{B} - e_1e_1^\mathsf{T})P^{-1}$ is nilpotent. For a rank-one matrix $\tilde{B}$ the solution of (16) can be computed explicitly.

Lemma 3 *Define $\tilde{A}$ by (10) with the rank-one matrix $\tilde{B} = e_1b^\mathsf{T}$ having the first row $b^\mathsf{T} = (1, \tilde{b}_{12}, \ldots, \tilde{b}_{1s})$. Then, the solution of (16) is given by the sum*

$$U = \sum_{k=0}^{s-1} e_{k+1}b^\mathsf{T}(DF_0^\mathsf{T})^kP^{-k-1}. \tag{17}$$

Proof Due to the condition $Ue_1 = e_1$ the system (16) simplifies here to

$$U - F_0UDF_0^\mathsf{T}P^{-1} = e_1b^\mathsf{T}P^{-1}. \tag{18}$$

Since F_0^s vanishes the linear map $X \mapsto F_0XDF_0^\mathsf{T}P^{-1}$ is nilpotent. Hence, the solution is unique and given by the finite Neumann series

$$U = \sum_{k=0}^{s-1} F_0^ke_1b^\mathsf{T}P^{-1}(DF_0^\mathsf{T}P^{-1})^k.$$

The assertion follows from $F_0e_k = e_{k+1}$, $k = 1, \ldots, s-1$ and the commutation property from Lemma 1. $\square$

Note in (17) that U is linear in the components of b. The assumptions of the last lemma lead to an extremely simple form of the transformed stability matrix (15) and yield the following result.

Theorem 4 *Let $\tilde{B}$ be the rank-one matrix $\tilde{B} = e_1b^\mathsf{T}$, $b^\mathsf{T} = (1, \tilde{b}_{12}, \ldots, \tilde{b}_{1s})$, and let $\tilde{A}$ be given by (10) with $\beta = 0$. Then, the characteristic polynomial $\mu(\lambda, z)$ of the*

stability matrix (4) has as only parameters the coefficients $\alpha_0, \ldots, \alpha_{s-1}$ of (13) and is a linear function of them, i.e.

$$\mu(\lambda, z) = \sum_{i=0}^{s} \lambda^i \mu_i(z) = \sum_{i=0}^{s} \lambda^i \Big(\mu_{i0}(z) + \sum_{k=1}^{s} \mu_{ik}(z) \alpha_{k-1} \Big), \tag{19}$$

with fixed polynomials μ_{ik}.

Proof The proof follows from the fact that the transformed matrix $U\tilde{B}U^{-1}$ in (15) is constant and no longer depends on the parameters b and ϕ, i.e.

$$U\tilde{B}U^{-1} = Ue_1 b^{\mathsf{T}} U^{-1} = e_1 d^{\mathsf{T}}, \tag{20}$$

with a constant vector d. Hence $U\tilde{M}(z)U^{-1}$ essentially has doubly companion form (cf. [3]) with constant entries d^{T} in the first row and $-z\alpha$ in the last column. From this follows the linear dependence on α in (19).

The difficult part is the proof of identity (20) where one has to show that $b^{\mathsf{T}} = d^{\mathsf{T}}U$ holds for arbitray b, $b^{\mathsf{T}}e_1 = 1$. The explicit form (17) leads to

$$d^{\mathsf{T}}U = \sum_{k=0}^{s-1} d_{k+1} b^{\mathsf{T}} \tilde{E}^k P^{-k-1} = b^{\mathsf{T}} P^{-1} \sum_{k=0}^{s-1} d_{k+1} \tilde{E}^k P^{-k}. \tag{21}$$

and we are done if we can find d such that $\sum_{k=0}^{s-1} d_{k+1} \tilde{E}^k P^{-k} = P$. Now, all powers $(\tilde{E}P^{-1})^k$ satisfy the following k-independent shift property

$$e_i \tilde{E}^k P^{-k} e_j = \frac{j-1}{i-1}\, e_{i-1} \tilde{E}^k P^{-k} e_{j-1},\ 1 < i \le j \le s,\ k \ge 0,$$

due to Lemma 1 since

$$e_i \tilde{E}^k P^{-k} e_j = \binom{j-1}{i-1} \frac{(j-i)!}{(j-i-k)!} (-k)^{j-i-k}.$$

The shift property carries over (for arbitrary d) to the whole sum yielding

$$e_i^{\mathsf{T}} P^{-1} \Big(\sum_{k=0}^{s-1} d_{k+1} \tilde{E}^k P^{-k} \Big) e_j = \binom{j-1}{i-1} e_1^{\mathsf{T}} \sum_{k=0}^{s-1} d_{k+1} \tilde{E}^k P^{-k-1} e_{j-i+1}, \tag{22}$$

$j \ge i$. The simpler matrix $U_0 = \sum_k e_{k+1} e_1^{\mathsf{T}} \tilde{E}^k P^{-k-1}$ is the solution of (18) with the fixed right hand side $e_1 e_1^{\mathsf{T}} P^{-1}$ instead of $e_1 b^{\mathsf{T}} P^{-1}$. Now, by choosing $d^{\mathsf{T}} := e_1^{\mathsf{T}} U_0^{-1}$ independent of b we are able to verify (20). By definition holds

$$d^{\mathsf{T}} U_0 = e_1^{\mathsf{T}} \sum_{k=0}^{s-1} d_{k+1} \tilde{E}^k P^{-k-1} = e_1^{\mathsf{T}}.$$

With (22) this carries over to the other components in (21) proving (20):

$$\begin{aligned} d^{\mathsf{T}}Ue_j &= \sum_{i=1}^{s} b_i e_i^{\mathsf{T}} \sum_{k=0}^{s-1} d_{k+1}\tilde{E}^k P^{-k-1} e_j = \sum_{i=1}^{s} b_i \binom{j-1}{i-1} d^{\mathsf{T}} U_0 e_{j-i+1} \\ &= \sum_{i=1}^{s} b_i \delta_{1,j-i+1} = b_j. \qquad \square \end{aligned}$$

Remark The coefficients d used in the proof seem to have the explicit form $d_{k+1} = \frac{(k+1)^{k-1}}{k!}$ and we think that they are the Taylor coeffients of the inverse function $(x \mapsto \log(x)/x)^{-1}$ at $x = 1$ (cf. Lemma 1) which means that $\sum_{k=0}^{\infty} d_k(\frac{\log x}{x})^k \equiv x$. The theorem shows that μ is a linear function of the new parameters α_i. This will be the basis for employing linear optimization in the search for suitable peer methods in Section 4. To be sure, these parameters α_i depend on the $2s-1$ original parameters $\tilde{b}_{1j}$ and ϕ_i and even products $\tilde{b}_{1j}\phi_i$ appear. But replacing the elements $\tilde{b}_{12}, \ldots, \tilde{b}_{1s}$ by the new parameters $\alpha_{s-1}, \ldots, \alpha_1$ leads to the simple form of μ in Theorem 4. In fact, the coefficient α_0 is itself a linear combination of the other parameters α_j, $j \geq 1$ and is superfluous in the statement of the theorem.

The linear representation (19) can be easily computed with an algebraic manipulation program even in more general situations. It is easily seen from (14) that the elements of the Frobenius transformation U depend linearly on the elements $\tilde{b}_{1j}$, $2 \leq j \leq s$, which only appear in the first row $\tilde{a}_{1j} = \frac{1}{j}(1 - b_{1,j+1})$, $1 \leq j \leq s-1$, of $\tilde{A}$. In fact, by (14) holds $u_{1,j+1} = -j(\tilde{a}_{1j} + u_{12}\tilde{a}_{2j} + \ldots)$ in the first row of U. In the second row we have $u_{2,j+1} = j(u_{1j} - \sum_{k=2} u_{2k}\tilde{a}_{kj})$. So, here and in all other rows the elements $\tilde{a}_{1j}$ do not show up again. Considering now the last row of the Frobenius companion of $\tilde{A}$, $\alpha = (F_0U - U\tilde{A})e_s/u_{ss}$, one can observe the explicit relations

$$\alpha_{s-j} = \frac{1}{j!}\,\tilde{b}_{1,j+1} + \ldots, \quad j = 1, \ldots, s-1, \tag{23}$$

which can be used to eliminate $\tilde{b}_{1,j+1}$ from the characteristic polynomial $\mu(\lambda, z)$. This property is only based on the Hessenberg structure of $\tilde{M}$, i.e. on the choice (12). In the situation of Theorem 4 this substitution leads to the linear dependence of $\mu(\lambda, z)$ on $\alpha_1, \ldots, \alpha_{s-1}$. However, we found that even for a larger class of matrices (12) with three more elements $\tilde{b}_{2,s-1}$, $\tilde{b}_{2s}$, $\tilde{b}_{3s}$ linear dependence in μ can be achieved. In fact, $\tilde{b}_{2s}$ can be substituted by α_0, but $\tilde{b}_{2,s-1}$, $\tilde{b}_{3s}$ lead to product terms in the matrix (15). Surprisingly, product terms like $\tilde{b}_{2,s-1}\alpha_{s-1}$ cancel out again in the computation of the characteristic polynomial μ. We have verified this property for stage numbers $s = 3, \ldots, 8$ by direct computation and state it as the following conjecture.

Conjecture *With a matrix*

$$\tilde{B} = \begin{pmatrix} 1 & \tilde{b}_{12} & \cdots & \cdots & \tilde{b}_{1,s-1} & \tilde{b}_{1s} \\ & 0 & \cdots & 0 & \tilde{b}_{2,s-1} & \tilde{b}_{2s} \\ & & 0 & \cdots & 0 & \tilde{b}_{3s} \\ & & & 0 & & 0 \\ & & & & \ddots & \vdots \\ & & & & & 0 \end{pmatrix} \tag{24}$$

and $\tilde{A}$ from (10) the characteristic polynomial (5) of the stability matrix can be written as an affine linear combination of polynomials in λ and z,

$$\mu(\lambda, z) = \sum_{i=0}^{s} \lambda^i \mu_i(z) = \sum_{i=0}^{s} \lambda^i \Big(\mu_{i0}(z) + \sum_{k=1}^{q} \mu_{ik}(z) \eta_k \Big), \tag{25}$$

with $q = s+2$ and the parameter vector $\eta^{\mathsf{T}} = (\eta_1, \ldots, \eta_q) = (\alpha_0, \ldots, \alpha_{s-1}, \tilde{b}_{2,s-1}, \tilde{b}_{3s})$.

Remark With (24) and free nodes $c_1, \ldots, c_{s-1}$ (we use $c_s = 1$ for convenience) the explicit peer method of order s has $2s + 1$ free parameters while the stability polynomial μ depends on $s + 2$ only. So, the described parameter substitution identifies subclasses of methods having the same stability polynomial. From the practical point of view it also allows to split the search for peer methods in two steps, first choosing the parameters η in (25) with respect to stability properties and looking then for good nodes with other criteria.

4 Method search by linear programming

Efficient time integration schemes have to combine good stability and accuracy properties. In order to exploit the linear structure of (25) in a search for good schemes we were looking for a formulation where linear optimization methods could be employed to compute 'optimal' methods exactly. In the following we develop a linear programming formulation where stability requirements lead to linear constraints and an accuracy property is used as objective function. The problem is simplified by considering only real stability intervals with

$$(-r, 0) \subseteq \{ z \in \mathbb{R} : \varrho(M(z)) < 1 \}. \tag{26}$$

Of course, the spectral radius of M depends on the parameters of the stability polynomial μ in a complicated way. However, with suitable root locus *bounds* it is possible to guarantee absolute stability by using only linear constraints. A simple general bound is due to Cauchy [6].

Lemma 5 *With real coefficients $\gamma_0, \ldots, \gamma_{s-1} \geq 0$ (not all zero) let $\hat{\xi}$ be the positive root of the polynomial*

$$\gamma(\xi) = \xi^s - \sum_{i=0}^{s-1} \gamma_i \xi^i,$$

and let the polynomial $p(\lambda) = \sum_{i=0}^{s} p_i \lambda^i$, $p_s = 1$, have complex coefficients $|p_i| \leq \gamma_i$, $i = 0, \ldots, s-1$. Then, any root $\hat{\lambda} \in \mathbb{C}$ of p satisfies $|\hat{\lambda}| \leq \hat{\xi}$.

Proof The assumption $|\hat{\lambda}| > \hat{\xi}$ immediatly leads to a contradiction. □

With this lemma we get a sufficient condition for (26) by requiring

$$|\mu_i(z)| \leq \gamma_i(z),\ i = 0, \ldots, s-1,\ \sum_{i=0}^{s-1} \gamma_i(z) \xi^{i-s} \leq 1,\ z \in [-r, \xi - 1],$$

with some ξ slightly smaller than one (e.g. $\xi = 0.99$). In view of the linear dependence (25) and for real z this is a set of linear two-sided inequalities, indeed. Even for complex z the conditions $|\mu_i| \leq \gamma_i$ might be enforced by a larger polygonal set of linear constraints. For practical a-priori estimates the coefficients γ_i in Lemma 5 have been chosen in many different ways in the literature. In the present context, however, there is no need to specify these coefficients. They can simply be incorporated in the linear program as additional unknowns where the simplex algorithm will choose them in an optimal way. Summarizing this discussion and using (25) the stability condition (26) is enforced by the set of linear inequalities

$$\begin{array}{c} -\gamma_i(z) \leq \mu_{i0}(z) + \sum_{k=1}^{q} \mu_{ik}(z) \eta_k \leq \gamma_i(z),\ i = 0, \ldots, s-1, \\ \sum_{i=0}^{s-1} \gamma_i(z) \xi^{i-s} = 1, \quad z \in [-r, \xi - 1]. \end{array} \tag{27}$$

The conditions $\gamma_i(z) \geq 0$ are redundant. In practice the real interval $[-r, \xi - 1]$ is approximated by a suitably fine grid. This is justified by the continuous dependence of polynomial roots. We note that the system (27) need not be feasible if r is too large. However, the largest possible abscissa r can be found conveniently with $O(\epsilon)$ accuracy by adding restrictions (27) at $z = -\epsilon, -2\epsilon, \ldots$ one by one and updating the solution with the dual simplex method.

Since many solutions to the system of inequalities (27) may exist for some r, it is possible to consider accuracy properties as additional objectives. Candidates here are the leading error coefficients $\mathbf{AB}(s+2)$, e.g. $AB_s(s+2)$ related to the stage $Y_{ms} \cong y(t_m + h_m)$. According to (7), however, these depend on the nodes c_i and by using this objective function we would loose the advantage of having a smaller set of parameters η which was formulated in the last remark of Section 3. Instead the

error in the approximation of the exponential function was considered through the coefficient

$$efc := \lim_{z\to 0} z^{-s-1}\mu(e^z, z). \tag{28}$$

Due to (25) this coefficient has an affine linear representation $efc = d_0 + \sum_{k=1}^{q} d_k\eta_k$ which can be easily computed by algebra. Combining the objective function $ef = |efc|$ with the root locus bounds the following linear programm for parameter search is obtained,

$$ef = \min! \quad \text{subject to } -ef \le efc \le ef \text{ and (27)}. \tag{29}$$

It can be solved exactly by the (dual) simplex method as described above. Before proceeding we note a relation to long-term error propagation which strengthens the importance of the error coefficient efc. In [11] it was shown that the leading error term in the global error is $v^{\mathsf{T}}AB(s+2)$ where v is the left eigenvector of B with $v^{\mathsf{T}}B = v^{\mathsf{T}}$, $v^{\mathsf{T}}\mathbb{1} = 1$.

Lemma 6 *Let $B^k = \mathbb{1}v^{\mathsf{T}}$, $k \ge s-1$, and A be given by (8) with $\beta = 0$. Then, $efc = v^{\mathsf{T}}AB(s+2)$.*

Proof Since one is a simple eigenvalue of B the stability matrix $M(z)$ has a simple eigenvalue $\lambda_1(z)$ with $\lambda_1(0) = 1$ in a neighbourhood of $z = 0$. Since the other eigenvalues $\lambda_i(z)$, $i \ge 2$, branch off from zero we have from (28) that $efc = \lim_{z\to 0} z^{-s-1}(e^z - \lambda_1(z))$. Summing the order conditions (7) with factors $z^k/k!$ shows that $\exp(z\mathbf{c}) = (e^{zc_i})_{i=1}^{s}$ is an approximate eigenvector of $M(z)$,

$$(e^z I - B - zA)\exp(z\mathbf{c}) = w, \quad w = \frac{z^{s+1}}{(s+1)!}AB(s+2) + O(z^{s+2}),$$

with small residual w. With a vector $u = u(z)$ satisfying $u^{\mathsf{T}}\exp(z\mathbf{c}) \equiv 1$ near $z = 0$, $z \in \mathbb{R}$, it is also an exact eigenvector of the matrix $M(z) + wu^{\mathsf{T}}$ since $(e^z I - M(z) - wu^{\mathsf{T}})\exp(z\mathbf{c}) = 0$. The left eigenvector of $M(z) = B + zA$ belonging to $\lambda_1(z)$ satisfies $y = v + O(z)$. Now, standard error estimates ([4],7.2.2) applied to $M(z) + wu^{\mathsf{T}}$ show that the first eigenvalue of $M(z)$ behaves like

$$\lambda_1(z) = e^z + \frac{y^{\mathsf{T}}w}{y^{\mathsf{T}}\exp(z\mathbf{c})} + O(\|w\|^2) = e^z + v^{\mathsf{T}}w + O(z^{s+2}). \quad \square$$

The linear program (29) with the sufficient root locus bound of Lemma 5 has been solved for s-stage methods based on (24) with $3 \le s \le 8$. The results are contained in (31). There is a distinction between the maximal interval $[r, 0]$ where the linear program (29) was feasible and the largest stability interval

$$[r^*, 0] \subset \{z \in \mathbb{R} : \varrho(M(z)) \le 1\}, \tag{30}$$

computed independently by the QR-method for the same set of method parameters. The small difference $r^* - r$ shows that the optimized root locus bounds are quite sharp. Furthermore, using such bounds instead of exact eigenvalues leads to complex stability regions that stretch out into the complex plane around the real axis. Since the error coefficients efc in the third row are rather large the lower part of table (31) shows the maximal abscissae r_0 where (29) was feasable with error constant $efc = 0$. Again, r_0^* is the abscissa according to (30).

s	3	4	5	6	7	8
r	1.35	1.35	1.10	0.95	0.85	0.80
r^*	1.355	1.350	1.100	0.985	0.875	0.805
efc	0.403	0.710	0	2.215	4.15	11.31
r_0	0.95	1.00	1.10	0.85	0.75	0.70
r_0^*	1.280	1.235	1.100	0.895	0.755	0.725

(31)

These methods have fairly large stability intervals and nice long-term error behaviour but, unfortunately, extremely large matrix entries in B for $s > 4$ since the elements $\tilde{b}_{2,s-1}, \tilde{b}_{3s}$ have very small factors in (25). For instance, the six-stage method with abscissa $r^* = 0.985$ in (31) has the matrix entry $\tilde{b}_{25} \cong 141$ and for the 8-stage method with $r^* = 0.805$ the corresponding element is $\tilde{b}_{27} \cong 17678$. Of course, size constraints on all parameters may be easily incorporated in the linear program (29) but they reduce the size of the resulting stability interval dramatically.

It was observed that the substitution of the parameters $\tilde{b}_{1,j+1}$ by α_{s-j}, cf. (23), still worked for a general matrix (12). Afterwards the representation of the characteristic polynomial $\mu(\lambda, z)$ was still linear in α, which means that only products of α_j with other parameters appear (for $s \leq 8$). With this property it is possible to use the linear program as an inner solution method of a Monte-Carlo simulation or a line search with one additional parameter $\tilde{b}_{ij}$, $i \geq 2$, $1 \leq j - i \leq s - 3$. However, this approach had its limits. For six-stage methods line searches with different parameters lead to an impractical method only with $\max_{i,j} |b_{ij}| \cong 68$ for $\tilde{b}_{35} = 4.5$. And for the interesting case $s = 8$ the Maple output for μ had a text size of 226 KByte and it was no longer possible to convert it into computer code for the simplex method. So we turned to a more pragmatic approach with more general parameter matrices B.

5 More general coefficients

The upper triangular structure (12) of $\tilde{B}$ discussed so far led to a reliable approach for parameter optimization but, unfortunately, to rather unpractical parameter sets. Therefore we also considered a general ansatz for matrices B having optimal zero stability given by

$$B = \mathbb{1}v^{\mathsf{T}} + QWQ^{-1} = Q(\mathbb{1}e_s^T + W)Q^{-1}, \quad v^{\mathsf{T}} = (\widetilde{v}^{\mathsf{T}}, 1 - \widetilde{v}^{\mathsf{T}}\mathbb{1}), \tag{32}$$

where

$$W = \begin{pmatrix} \widetilde{W} & -\widetilde{W}\mathbb{1} \\ 0^{\mathsf{T}} & 0 \end{pmatrix}, \quad Q = \begin{pmatrix} I - \mathbb{1}\tilde{v} & \mathbb{1} \\ -\tilde{v}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} \widetilde{Q} & -\widetilde{Q}\mathbb{1} \\ 0 & 1 \end{pmatrix}$$

with a strictly upper triangular matrix $\widetilde{W}$ and regular $\widetilde{Q}$. Due to $Q\mathbb{1} = \mathbb{1}$, $W\mathbb{1} = 0$, $v^{\mathsf{T}}Q = e_s^{\mathsf{T}}$ the matrix B has the right eigenvector $\mathbb{1}$ and the left eigenvector v with the consequence that $B^k = \mathbb{1}v^{\mathsf{T}}$ for $k \geq s - 1$. Of course, the ansatz (32) has more degrees of freedom than B itself but leads to well-defined properties of B.

The parameter space of (32) including the nodes $c_1, \ldots, c_{s-1}$ has been searched bei Monte-Carlo simulation for methods of order s with the following criteria. First, the leading term in the long-time error is small by the condition $v^{\mathsf{T}}AB(s+2) \cong 0$. The second criterion was a reasonably large stability region. Here, the expensive test of the spectral radius was confined to the real axis and some simple curves in the complex plane. Since methods with large entries in B or A may produce large rounding errors as a third objective a small size of

$$vmax = \max_{i,j} |(D^{-1}PV^{-1})_{i,j}| \tag{33}$$

was considered since the matrix $D^{-1}PV^{-1}$ is an important factor in A, see (8). We note, that only small overall changes of the initial guess $v = e_s$, $W = 0$, $Q = I$ in (32) where used in the Monte-Carlo search. In the numerical tests all methods have been found by this search with the exception of the four-stage method `epp4l` constructed with the linear program (29). The stability abscissae and error constants of all test methods for $\sigma = 1$ are given in the following table.

method	s	r	$\lvert AB_s(s+2)\rvert$	$\lvert v^T AB(s+2)\rvert$
epp4l	4	1.176	18.3	10^{-6}
epp4y	4	0.741	0	0
epp6	6	0.579	0	0
epp8	8	0.548	0.356	0

Note that method `epp4l` from Section 4 indeed has a much larger stability interval than method `epp4y`.

6 Parallel implementation and tests

Numerical tests have been performed with a Fortran90 implementation in double precision using OpenMP parallelization on a 4-processor node of the MARC Opteron cluster at Marburg. This means that the peer methods with more than four stages may not exhibit their full parallel potential. The time step of the peer method (3) consists of two distinct parts which have been designed for easy parallelization on s processors. First, the s function evaluations $f(Y_{m-1,1}), \ldots, f(Y_{m-1,s})$ have perfect parallelism in practice with an OpenMP `DO PARALLEL` directive. Similarly, the second part with s independent linear combinations for Y_m looks like an easy parallel task. Unfortunately, here the parallel performance (of several different implementations) is quite poor for reasons unknown. Even with the computation of the product $(B, A) \cdot (Y_{m-1}^{\mathsf{T}}, h_m F_{m-1}^{\mathsf{T}})^{\mathsf{T}}$ by one single call of the BLAS GEMM routine from the optimized Intel MKL library the speed up in the matrix multiplication did not exceed two. So, only for quite expensive right-hand sides f an optimal speed-up near s may be observed in practice.

Here are more details of the implementation. The approximations Y_{0j} in the first time interval are computed with the one-step method DOPRI5 using slightly sharper tolerances. In each time step the matrix A_m (8) has to be computed with the actual stepsize ratio σ. This is done with the decomposition $\sigma A = A' + (CV)(\sigma S)(D^{-1}PV^{-1})$ in (8) with $O(s^2)$ FLOPs per processor and time step which 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. Moreover, the last row $e_s^{\mathsf{T}} D^{-1}PV^{-1}$ of the last one corresponds to a difference quotient of order $s-1$ and so,

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

can be used as error estimate in the current step. The error criterion was

$$\frac{1}{n}\sum_{i=1}^{n} \frac{ee_i^2}{(atol + rtol|Y_{m-1,s,i}|)^2} \le 1. \tag{34}$$

It was a surprise to note that this estimate is available directly after the computation of F_{m-1} and *before* the step to Y_m in (3) is performed. Hence, the stepsize ratio σ_m satisfying the error criterion (34) may be computed explicitly. Then, with $h_m = \sigma_m h_{m-1}$ the large matrix multiplication yielding Y_m can be performed without any risk of step rejections.

Motivated by the remarks on parallel performance we consider the following test problems with right-hand sides having different evaluation costs. Two are well-known test problems modified for larger dimensions in order to produce a visible computer load.

- BRUS2h, the Brusselator with the parameters from [5] but a larger 200×200 grid. The diffusion constant $\nu = 10^{-5}$ was chosen smaller in order to obtain a mildly stiff problem. The dimension is $n = 80000$.
- PLEI1h, the Pleiades problem from [5] for 7 stars in $\mathbb{R}^2$. 100 identical copies give dimension $n = 2800$.
- MBOD4h, a celestial multi-body problem with 400 objects in $\mathbb{R}^3$ moving only under the forces of gravity from an initital disk-shaped distribution. The dimension is $n = 2400$.

A reference solution for PLEI1h is known, those for BRUS2h and MBOD4h were computed by DOPRI5 with $TOL = 10^{-14}$. Each problem is solved with $rtol = atol = TOL \in \{10^{-3}, 10^{-4}, \ldots, 10^{-12}\}$. While the computational cost of the f-evaluation for the Brusselator is quite low, $\cong 10n$, the multi-body problem is very expensive since $O(n^2)$ forces have to be computed. This difference has a great effect on the parallel performance which is demonstrated in Figure 1 showing speed-ups for the four-stage method `epp4y` on 1, 2 and 4 processors.

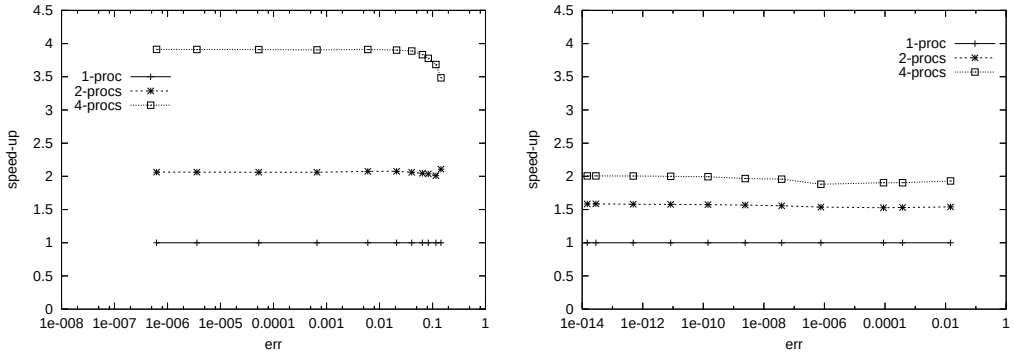

Figure 1: Speed-up for MBOD4h (left) and BRUS2h (right)

It is seen that the performance at the cheap Brusselator suffers from the limited speed-up in the matrix-matrix multiplication in (3). In the multibody problem MBOD4h, however, this multiplication is only a small part of the overall cost leading

to an almost optimal speed-up of 3.91 for sharp tolerances. For the PLEI1h problem a speed-up of 3.2 was observed for this 4-stage method. The detailed results for all peer methods on four processors and DOPRI5 on one are contained in Figure 2. The slopes of the different curves nicely correspond to the different orders of the peer methods. The method `epp4l` from the optimization in Section 4 draws no advantage from its larger stability interval. DOPRI5 catches up with the other 4-stage method `epp4y` for sharper tolerances but is outperformed by the 6- and 8-stage methods. We note that these two peer methods are expected to be even faster on 6 or 8 processors which were not available to us with OpenMP.

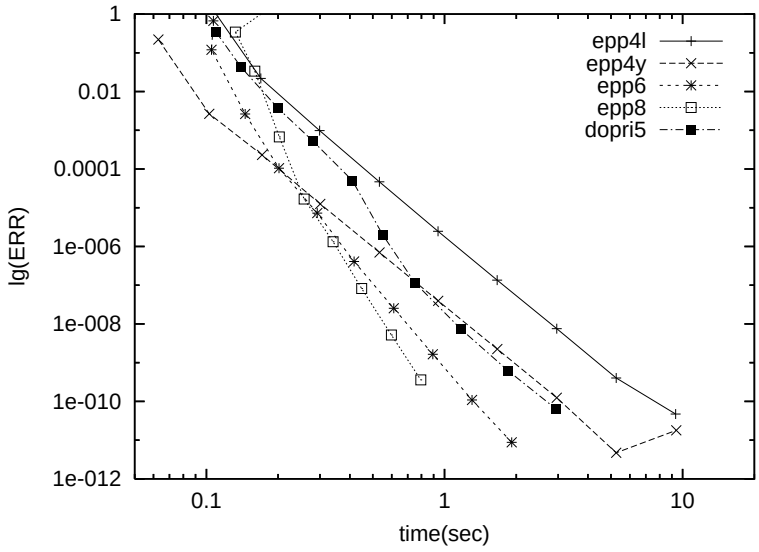

Figure 2: Method efficiency at problem PLEI1h

The results for the other problems confirm this picture. Due to a greater sensitivity of the problem the obtained errors in MBOD4h are much larger than the prescribed tolerances and we omitted the results with $TOL = 10^{-3}$ in Figure 3. Here DOPRI5 also shows the weakest performance, comparable to `epp4l`.

7 Conclusions

Two approaches have been discussed for searching for explicit parallel peer methods having optimal zero stability and order s. In a reduced part of the parameter space a linear representation of the stability polynomial has been obtained. Based on this

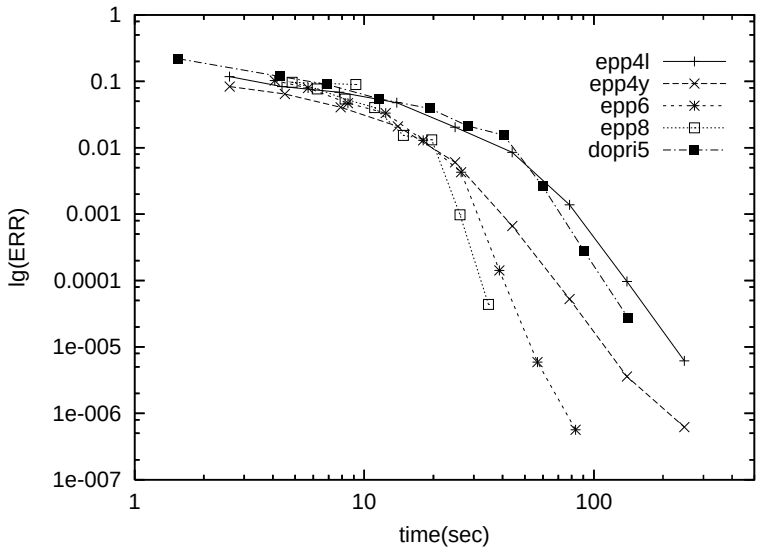

Figure 3: Method efficiency at problem MBOD4h

form a linear program has been formulated and solved for methods with large stability interval and small long-term error. Unfortunately, these methods have poor practical properties and as an alternative a Monte-Carlo search based on a rather general ansatz was performed, too. The methods found by the second approach are very competative with DOPRI5 on a parallel computer in a set of demanding test problems. Despite its limited practical impact the approach with the linear representation is an interesting theoretical result since it contains much less parameters and identifies subclasses of explicit methods with identical stability properties. Also, some post-optimization of very stable methods from this approach may produce better methods in the future since our Monte-Carlo simulation is restricted to small changes from a good initial guess.

References

[1] C. Bendtsen, *A parallel stiff ODE solver based on MIRKs*, Adv. Comput. Math. 7 (1997), 27–36.

[2] K. Burrage, *Parallel and sequential methods for ordinary differential equations*, Clarendon Press, Oxford, 1995.

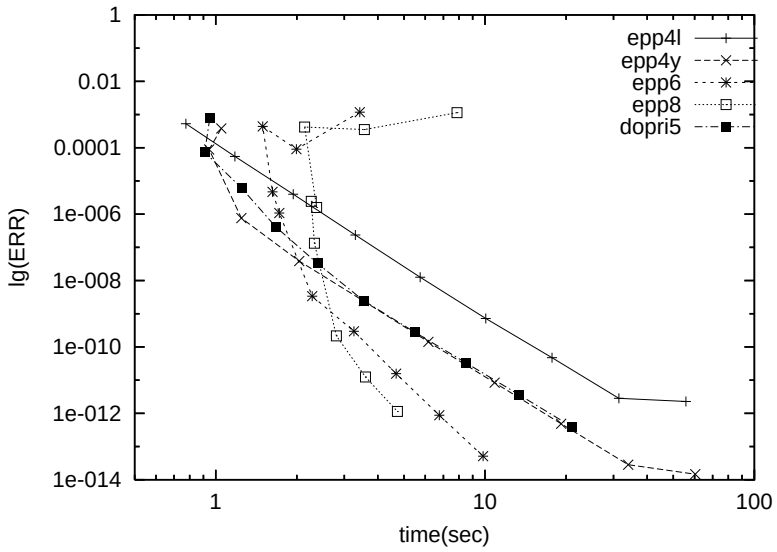

Figure 4: Method efficiency at problem BRUS2h

[3] J.C. Butcher, *Order and effective order*, Appl. Numer. Math. 28 (1998), 179-191.

[4] G.H. Golub, C.F. Van Loan, *Matrix computations*, 2nd ed., The Johns Hopkins University Press, Baltimore and London, 1989.

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

[6] A. Householder, *The numerical treatment of a single nonlinear equation*, McGraw-Hil, 1970.

[7] 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).

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

[9] B.A. Schmitt, R. Weiner, *Parallel Two-Step W-Methods with Peer Variables*, SIAM J. Numer. Anal. 42 (2004), 265-282.

[10] B.A. Schmitt, R. Weiner, and K. Erdmann, *Implicit Parallel Peer Methods for stiff initial value problems*, APNUM 53, 457–470 (2005).

[11] 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).

[12] R. Weiner, K. Biermann, B.A. Schmitt and H. Podhaisky, *Explicit two-step peer methods*, Report of the Institute of Numerical Mathematics, University of Halle, No. 10 (2005).

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