BIT
1998, Vol. 38, No. 2, pp. 372–390

006-3835/98/3802-0372 $12.00

HIGH ORDER EXPLICIT METHODS FOR PARABOLIC EQUATIONS *

ALEXEI A. MEDOVIKOV †

Institute of Numerical Mathematics, Gubkina street 8, Moscow 117333 Russia. email: nucrect@inm.ras.ru

Abstract.

This paper discusses explicit embedded integration methods with large stability domains of order 3 and 4. The high order produces accurate results, the large stability domains allow some reasonable stiffness, the explicitness enables the method to treat very large problems, often space discretization of parabolic PDEs, and the embedded formulas permit an efficient stepsize control. The construction of these methods is achieved in two steps: firstly we compute stability polynomials of a given order with optimal stability domains, i.e., possessing a Chebyshev alternation; secondly we realize a corresponding explicit Runge–Kutta method with the help of the theory of composition methods.

AMS subject classification: 65L20, 65M20.

Key words: Explicit Runge–Kutta methods, stiff ordinary differential equations, parabolic equations, approximation by polynomials.

1 Introduction.

The aim of this paper is to construct explicit Runge–Kutta methods with a large stability interval along the real axis. Such methods are frequently called Chebyshev (П.Л. Чебышев) methods and are usually used with orders one and two. In this paper we shall derive such methods of orders three and four. Typical applications of such methods are mildly stiff ODEs arising from a semidiscretization of a parabolic PDE.

We write the system of ordinary differential equations as

$$\frac{dy}{dt} = f(t,y), \qquad y|_{t=t_0} = y_0, \tag{1.1}$$

and solve it by an explicit Runge–Kutta method,

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{i-1} a_{ij} f(t_0 + c_j h, Y_j), \\ y_1 &= y_0 + h\sum_{j=1}^{s} b_j f(t_0 + c_j h, Y_j). \end{aligned} \tag{1.2}$$

*Received November 1996. Revised December 1997. Communicated by Syvert P. Nørsett.
†This work was supported by the Russian Fund Fundamental Researches and the Swiss National Science Foundation 20-43.314.95.

The coefficients of this method are usually displayed as

$$
\begin{array}{c|cccc}
0 & 0 & 0 & \dots & 0 & 0 \\
c_2 & a_{21} & 0 & \dots & 0 & 0 \\
c_3 & a_{31} & a_{32} & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \\
c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & 0 \\
\hline
1 & b_1 & b_2 & \dots & b_{s-1} & b_s
\end{array}
\qquad \text{(1.3)}
$$

, or $\begin{array}{c|c} \mathbf{c} & A \\ \hline 1 & \mathbf{b}^t \end{array}$,

where $\mathbf{c}^t = (0, c_2, c_3, \cdots, c_s)$, $\mathbf{b}^t = (b_1, \dots, b_s)$.

For the stability analysis we apply method (1.2) to the test problem

$$y' = -\lambda y, \qquad y\Big|_{t=t_0} = y_0, \tag{1.4}$$

for which the numerical result becomes

$$y_1 = R_s(\lambda h) y_0 = \left(1 + \sum_{i=0}^{s-1} \mathbf{b}^t A^i \mathbf{e} (-\lambda h)^{i+1}\right) y_0, \tag{1.5}$$

where $\mathbf{e}^t = (1, \dots, 1)$ and $R_s(z)$, $z = \lambda h$, is the stability polynomial (of degree s). The region $U = \{z : |R_s(z)| \le 1\}$ is called the ***stability region***. We are specially interested in the largest interval on the real axis $I = [0, l]$ which is contained in U and call it the ***real stability interval***. We will construct explicit methods whose real stability interval is as large as possible.

A first step is the computation of the stability polynomial, which, for an s-stage explicit Runge–Kutta method of order p, must be of the form

$$R_s(z) = 1 - z + \dots + (-1)^p z^p / p! + \sum_{i=p+1}^{s} d_i z^i. \tag{1.6}$$

This follows from formula (1.5) and the order conditions (4.5) below. The optimal polynomial with maximal stability interval, however, has an undesirable property: namely, at several places its absolute value is exactly equal to one. For practical calculations it is desirable to introduce more stability by choosing a value of deviating from zero $\eta < 1$ and by constructing polynomials such that

$$\max_{t \in [\delta, l]} |R_s(t)| \le \eta, \tag{1.7}$$

with l as large as possible. The value δ is necessary because of $R_s(0) = 1$, (1.7) is not possible close to $t = 0$. We choose the values of η depending on the degree s as $0.96 \le \eta \le 0.985$ according with an experimental table of Lebedev (see [8]). First Chebyshev methods were proposed by Saul'ev (1960), Saul'ev's postgraduate student Yuan Chzao Din (1958), Franklin (1959), and Guillou

and Lago (1961), to solve semi-discrete parabolic PDEs. The optimal stability polynomials (1.6) for order $p = 1$ are shifted Chebyshev polynomials

$$R_s(z) = \frac{1}{T_s(w_0)} T_s(w_0 - w_1 z), \quad w_0 = 1 + \frac{1-\eta}{s^2}, \quad w_1 = \frac{T_s(w_0)}{T_s'(w_0)},$$

(see Figure 1.1). Formulas for the optimal stability polynomials of order $p = 2$ were derived by Lebedev in [9, 7].

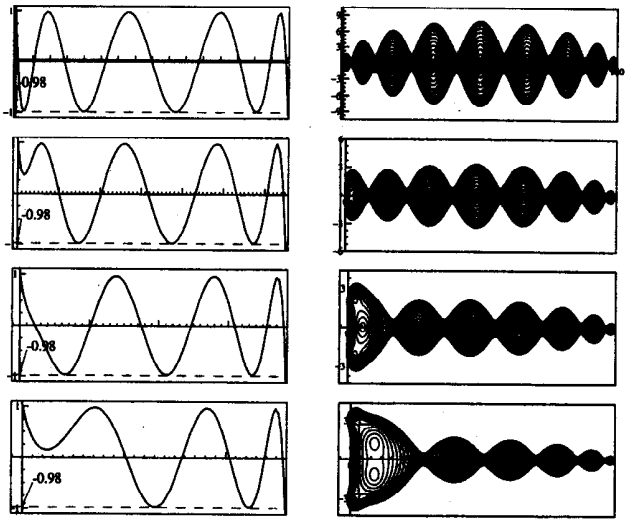

Figure 1.1: First (Chebyshev), second (Zolotarev, Lebedev), third and fourth order polynomials and their stability regions.

The second step is the realization of the stability polynomial as a Runge–Kutta method. Van der Houwen and Sommeijer [6] have proposed an elegant idea based on the three-term recursion formula for Chebyshev polynomials, which can also be applied to second order methods (see the review of Verwer [21] and the code RKC by Sommeijer). Another idea, mentioned by Saul'ev (1960) and Guillou and Lago (1961), is to represent the RK method as the composition of explicit Euler steps. Lebedev [11] has extended this idea and uses the scheme

$$\begin{aligned}
Y_0 &= y_0, \\
U_{i+\frac{1}{2}} &= Y_i + h\alpha_{i+1} f(t_i, Y_i), && t_{i+\frac{1}{2}} = t_i + h\alpha_{i+1}, \\
U_{i+1} &= U_{i+\frac{1}{2}} + h\alpha_{i+1} f(t_{i+\frac{1}{2}}, U_{i+\frac{1}{2}}), && t_{i+1} = t_{i+\frac{1}{2}} + h\alpha_{i+1}, \\
Y_{i+1} &= U_{i+1} - h\nu_{i+1}\alpha_{i+1}(f(t_{i+\frac{1}{2}}, U_{i+\frac{1}{2}}) - f(t_i, Y_i)) \\
&= U_{i+1} - \nu_{i+1}(U_{i+1} - 2U_{i+\frac{1}{2}} + Y_i), && i = 0, \ldots, s-1, \\
y_1 &= \begin{cases} Y_s, & \text{if } s \text{ even}, \\ Y_{s-1} + h\alpha_s f(t_{s-1}, Y_{s-1}), & \text{if } s \text{ odd}, \end{cases}
\end{aligned} \tag{1.8}$$

with stability polynomials of the form

$$R_s(z) = \begin{cases} \prod_{i=1}^{s/2} [(1-\alpha_i z)^2 - \nu_i \alpha_i^2 z^2], & s \text{ even} \\ (1-\alpha_s z) \prod_{i=1}^{[s/2]} [(1-\alpha_i z)^2 - \nu_i \alpha_i^2 z^2], & s \text{ odd.} \end{cases}$$

In Section 2 of this paper we calculate the roots of the optimal stability polynomials of orders $p = 3, 4$ with the help of program BERN. This program is described briefly in Section 3 and the articles [14, 10]. Then we construct in Section 4 third and fourth order Runge–Kutta methods with the help of the composition method approach. Section 5 constructs embedded composition methods for error control.

These methods have been implemented into the computer codes DUMKA, DUMKA3 and DUMKA4. Section 6 presents some numerical results. It turns out that the best applications for these programs are high-dimensional and not extremely stiff problems, usually semi-discrete PDEs with eigenvalues of the Jacobian matrix $f'(t,y) = \partial f(t,y)/\partial y$ in a narrow strip along the real axis of the complex plane. For non-stiff problems the methods perform as usual 3rd and 4th order explicit embedded integration schemes.

A problem, not addressed to in this paper, is the internal stability of the constructed method. This problem is treated by Lebedev [11, 8] by a proper ordering of the roots and parameters α_i (see, e.g., the second edition of [3], p. 33). Since we have found that most of the roots of the normalized optimal stability polynomial (2.1) of order $p \geq 2$ are close to the roots of the normalized Chebyshev stability polynomial of the same degree (see, e.g., the last $s-p$ roots in Table 2.4), we have applied a slight modification of Lebedev's algorithm for Chebyshev stability functions.

2 Construction of stability polynomials.

Our problem is to find, for a given s, a polynomial of the form (1.6) such that the corresponding real stability interval is, in the direction of positive z, as large as possible. We introduce the coordinate change $z = lt$ so that (1.6) becomes

$$F_s(t) = 1 - lt + \cdots + (-1)^p (lt)^p/p! + \sum_{i=p+1}^{s} (d_i l^i) t^i \tag{2.1}$$

and require that

$$\max_{t \in [0,1]} |F_s(t)| \leq 1, \tag{2.2}$$

with l as large as possible. The optimal polynomial, among all polynomials of order p, possesses a Chebyshev alternation (called *equal ripple property* by Riha [16]):

THEOREM 2.1: *For $s > p$ condition (2.2) is satisfied with maximal l if there exist $s+1-p$ points*

$$t_i : 0 < t_1 < t_2 \cdots < t_{s+1-p} \leq 1$$

such that:

$$F_s(t_i) = (-1)^{p+i-1}, \qquad i = 1, \ldots, s+1-p. \tag{2.3}$$

The existence of such a polynomial with Chebyshev alternation is then assured by Riha [16] (see Figure 1.1).

In the situation corresponding to problem (1.7), with additional damping, we obtain the requirement

$$\max_{t\in[\epsilon,1]} |F_s(t)| \le \eta, \tag{2.4}$$

with l as large as possible and with $\epsilon \approx (1 - \sqrt{2\eta - 1})/l$. Here we will construct polynomials which satisfy $s + 1 - p$ points

$$t_i : \epsilon < t_1 < t_2 \cdots < t_{s+1-p} \le 1$$

such that

$$F_s(t_i) = (-1)^{p+i-1}\eta, \quad i = 1, \ldots, s+1-p. \tag{2.5}$$

The first idea to construct such polynomials is the use of a Remez type procedure (see [4, 12, 15]). But the Jacobian in this algorithm very quickly becomes ill-conditioned and it is impossible to construct higher degree polynomials with this method.

An alternative method, which works for all degrees, has first been developed for second order polynomials by Kovalenko [1] and by Medovikov and Lebedev [7, 11]. This method is based on the following fact: For a given weight function $q(x)$, defined for some interval $a \le x \le b$, there exists an excellent algorithm for computing the polynomial "of least deviation from zero"

$$P_n(x) = \prod_{i=1}^{n} (x - \beta_i),$$

such that

$$\inf_{\beta_i} \max_{x\in[a,b]} |P_n(x)q(x)| = \max_{x\in[a,b]} |P_n(x)q(x)|.$$

This algorithm is described in more detail in Section 3 below and has been incorporated in the code BERN [7, 11]. In order to use this algorithm for our problem, we write the polynomial $F_s(t)$ of (2.4) as the product

$$F_s(t) = w_p(t)G_{s-p}(t),$$

where $w_p(t)$, of the form

$$w_p(t) = 1 - r_1 t + r_2 t^2 + \cdots + (-1)^p r_p t^p, \tag{2.6}$$

is considered as (firstly unknown) weight function. Then

$$G_{s-p}(t) = \prod_{i=p+1}^{s} (1 - t/\gamma_i)$$

is the minimal solution on the interval $[\epsilon^*, 1]$ obtained from BERN in dependence of this weight function, where $0 < \epsilon < \epsilon^* < 1$ (see Figure 2.1).

[1] It is a Russian tradition to place deceased colleagues into a framebox.

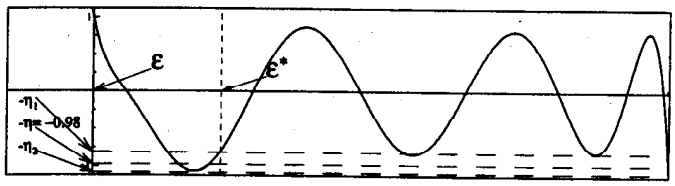

Figure 2.1: Computation of a third order polynomial with Algorithm 2.2.

We then adapt iteratively $w_p(t)$ and ϵ^* in order to satisfy

$$\eta_1 = \eta_2 = \eta, \tag{2.7}$$

where

$$\eta_1 = \max_{t\in[\epsilon^*,1]} |F_s(t)|, \tag{2.8}$$

and the value

$$\eta_2 = \max_{t\in]0,\epsilon^*]} |F_s(t)| \tag{2.9}$$

is the local maximum of the function $|F_s(t)|$ in the half-open interval $]0,\epsilon^*]$. This value is either equal to $F_s(\epsilon^*)$ or equal to $F_s(t_0)$, where $F_s'(t_0) = 0$ and $t_0 \in]0,\epsilon^*]$.

We also must satisfy the order property of the polynomial

$$p: \quad F_s^{(p)}(t)|_{t=0} = (-l)^p \tag{2.10}$$

or, in terms of the parameters of the polynomials $w_p(t)$ and $G_{s-p}(t)$,

$$\begin{aligned}
p=1: \quad r_1 &= l - \sum_{p<i}^{s} 1/\gamma_i, \\
p=2: \quad r_2 &= l^2/2! - r_1 \sum_{p<i}^{s} 1/\gamma_i - \sum_{p<i<k}^{s} \frac{1}{\gamma_i\gamma_k}, \\
p=3: \quad r_3 &= l^3/3! - r_1 \sum_{p<i<k}^{s} \frac{1}{\gamma_i\gamma_k} - r_2 \sum_{p<i}^{s} \frac{1}{\gamma_i} - \sum_{p<i<k<l}^{s} \frac{1}{\gamma_i\gamma_k\gamma_l}, \\
p=4: \quad r_4 &= l^4/4! - r_1 \sum_{p<i<k<l}^{s} \frac{1}{\gamma_i\gamma_k\gamma_l} - r_2 \sum_{p<i<k}^{s} \frac{1}{\gamma_i\gamma_k} \\
&\quad - r_3 \sum_{p<i}^{s} \frac{1}{\gamma_i} - \sum_{p<i<k<l<m}^{s} \frac{1}{\gamma_i\gamma_k\gamma_l\gamma_m}.
\end{aligned} \tag{2.11}$$

The parameters γ_i are the roots of the least deviation from zero polynomial $G_{s-p}(t)$ with the weight function $w_p(t)$, consequently the values γ_i, $i = p+1,\ldots,s$ are functions of the coefficients r_i, $i = 1,\ldots,p$ and ϵ^*. So, we have p

equations and $p+2$ variables in (2.11): $r_1, \ldots, r_p, l, \epsilon^*$. The values η_1 and η_2 are also functions of $r_1, \ldots, r_p, l, \epsilon^*$. Next we add to system (2.11) the equations which follow from (2.7),

$$\eta_1(r_1, \ldots, r_p, l, \epsilon^*) = \eta, \tag{2.12}$$

$$\eta_2(r_1, \ldots, r_p, l, \epsilon^*) = \eta_1(r_1, \ldots, r_p, l, \epsilon^*). \tag{2.13}$$

The solutions of (2.11), (2.12) and (2.13) give us polynomials which satisfy (2.4) and (2.5). Inequality (2.4) is valid because we have $|F_s(\epsilon)| < \eta$ if $\epsilon = (1 - \sqrt{2\eta - 1})/l$ (due to $1 - lt \le F_s(t) \le 1 - lt + l^2t^2/2$ in a small vicinity of zero). The function $|F_s(t)|$ is smoothly diminishing in the interval $[0, \epsilon]$, and the point t_0 of the local maximum of the function $|F_s(t)|$ in the partly open interval $]0, \epsilon^*]$ belongs to the interval $[\epsilon, \epsilon^*]$. So, we can enlarge the interval $[\epsilon^*, 1]$, from inequality (2.8), up to $[\epsilon, 1]$ like in (2.4),

$$\max_{t\in[\epsilon,\epsilon^*]} |F_s(t)| = \max_{t\in[\epsilon^*,1]} |F_s(t)| = \max_{t\in[\epsilon,1]} |F_s(t)| \le \eta_2 = \eta_1 = \eta.$$

Property (2.5) is valid because the polynomial $G_{s-p}(t)$ is the least deviation from zero on the interval $[\epsilon^*, 1]$ with weight function $w_p(t)$. According to Chebyshev's theorem about approximations by a polynomial [19, 18], there exist $s - p + 1$ points $\{t_i\}_{i=1}^{s+1-p}$, $t_i \in [\epsilon^*, 1] \in [\epsilon, 1]$ in Chebyshev alternation (2.5), where $\eta_1 = \eta$,

$$F_s(t_1) = (-1)^p\eta_1, \ldots, F_s(t_i) = (-1)^{p+i-1}\eta_1.$$

We can now formulate the algorithm of construction of the "near" optimal polynomials.

ALGORITHM 2.1. Choose the desired value of damping: $\eta \le 1$ (for example, $0.96 \le \eta \le 0.985$). Take some initial approximations of the roots γ_i ($i = p+1, \ldots, s$) of the polynomial $G_{s-p}(t)$ and of values l and ϵ^*.

1. Find the parameters of the weight $w_p(t)$ from (2.11).

2. Find the new roots of the polynomial of least deviation from zero $G_{s-p}(t)$ with the weight function $w_p(t)$ in the interval $[\epsilon^*, 1]$ with the algorithm explained in Chapter 3.

3. Calculate the value of deviation from zero η_1 on the interval $[\epsilon^*, 1]$ and η_2-value of local maximum of the function $|F_s(t)|$ in the interval $]0, \epsilon^*]$, and recalculate the new values l and ϵ^* as

$$l_{new} = l + \zeta_1(\eta - \eta_1),$$
$$\epsilon^*_{new} = \epsilon^* + \zeta_2(\eta_1 - \eta_2),$$

where the values ζ_1 and ζ_2 are some iteration parameters. Then we accept the new values l_{new} and ϵ^*_{new} as new parameters l and ϵ^* and return to Step 1.

Table 2.1: The stability regions of the "near" optimal second order polynomials.

Degree s	Stability region l	Value $\beta_2 = l/s^2$	Degree s	Stability region l	Value $\beta_2 = l/s^2$
5	19.389406	0.7755762	7	38.988738	0.7956885
9	65.044521	0.8030187	17	234.00230	0.8096965
25	507.29811	0.8116769	45	1646.0316	0.8128551

Table 2.2: The stability regions of the "near" optimal third order polynomials.

Degree s	Stability region l	Value $\beta_3 = l/s^2$	Degree s	Stability region l	Value $\beta_3 = l/s^2$
3	2.500512	0.2778346	6	15.967696	0.4435471
9	38.317952	0.4730611	15	109.96357	0.4887269
21	217.95736	0.4942343	36	644.30201	0.4971466
48	1145.8047	0.497311	243	29376.454	0.4974928

Table 2.3: The stability regions of the "near" optimal fourth order polynomials.

Degree s	Stability region l	Value $\beta_4 = l/s^2$	Degree s	Stability region l	Value $\beta_4 = l/s^2$
6	9.9330219	0.2759172	8	19.852540	0.3101959
10	32.545736	0.3254573	14	66.644594	0.3400234
16	87.9657997	0.3436164	24	201.71096	0.3501926
40	565.19543	0.3532471	50	884.68508	0.353874

This iterative process leads us to the "near" optimal polynomials $F_s(t)$ which satisfy conditions (2.5) and (2.4).

The author has computed these polynomials for order 2, 3 and 4 up to degree 243. The results for degrees 9, which correspond to the polynomials drawn in Figure 1.1, are presented in Table 2.4. The other values can be obtained from the author. Tables 2.1, 2.2 and 2.3 contain the values of l for some selected values of s. It can be observed that they behave like $l \approx \beta_p s^2$ where $\beta_2 \approx 0.81$, $\beta_3 \approx 0.49$, $\beta_4 \approx 0.35$. This confirms the values found by Metzger [15], Lomax [12] and van der Houwen [4] (see also Verwer [21]). Compared to Euler's method whose stability interval has length 2 per step, our methods allow an acceleration of $v_p = \frac{\beta_p s^2}{2s} = \frac{\beta_p}{2} s$, which becomes enormously advantageous for large s.

3 Construction of the least deviating from zero polynomials with weight functions.

Let there be given a smooth function $q(x)$, positive in the interval $(-1, 1)$.

PROBLEM 3.1: *Construct the polynomial*

$$P_n(x) = \prod^{n} (x - \beta_i), \tag{3.1}$$

Table 2.4: The roots of the first, second, third and fourth order stability polynomials degree 9 for $\eta = 0.98$.

Order	Roots
1	0.0077207095, 0.067104428, 0.17870931, 0.32907416, 0.50006276, 0.67105137, 0.82141622, 0.93302111, 0.99240483
2	(2.0092404E-2,± 2.0619529E-2), 0.15436564, 0.31091584, 0.48696657, 0.66256497, 0.81684573, 0.93131413, 0.99221162
3	(2.3078425E-2,± 6.4071797E-2), 0.057070367, 0.26509004, 0.45644436, 0.64340227, 0.80668190, 0.92754760, 0.99178672
4	(9.6446753E-3,± 1.2892855E-1), (9.0545672E-2,± 4.8412173E-2), 0.39412417, 0.60723386, 0.78806692, 0.92074069, 0.99102274

of the least deviation from zero with the weight function $q(x)$ in the interval $[-1, 1]$.

This means that we have to find the roots (or coefficients) of a polynomial of degree n so that

$$\inf_{\beta_i} \max_{x\in[-1,1]} |P_n(x)q(x)| = \max_{x\in[-1,1]} |P_n(x)q(x)|. \tag{3.2}$$

Analytical solutions of this problem are complicated and we know only some special solutions; for example, if $q(x) = 1$ the solutions are the Chebyshev polynomials, [19]. Let $p(x) = q^2(x)$. Here we describe a numerical algorithm for problem 3.1. The algorithm is based on the following theorem which we state without proof:

THEOREM 3.2 (see [1, 10, 13, 17]) *If $0 < \lambda < p(x) < \Lambda$ and $|p(x+\delta) - p(x)||\ln\delta|^{1+\epsilon} < K$ (ϵ and K some constants) the polynomial $P_n(x)$ of least deviation from zero with weight function $\sqrt{p(x)}$ in the interval $[-1, 1]$ is expressed in asymptotical form as*

$$\begin{aligned} P_n(x) &= \sqrt{\frac{2}{\pi p(x)}}\cos(n\theta + \psi(\theta)) + \epsilon_n(x), \qquad \theta = \arccos(x) \\ \psi(\theta) &= \frac{1}{2\pi}\int_{-1}^{1} \frac{\ln p(z) - \ln p(x)}{z - x}\sqrt{\frac{1-x^2}{1-z^2}}\,dz. \end{aligned} \tag{3.3}$$

COROLLARY 3.3: *The roots β_k of the polynomial of Theorem 3.2 satisfy asymptotically*

$$n\theta_k + \psi(\theta_k) \approx \pi(k - 0,5), \quad \beta_k = \cos\theta_k, \tag{3.4}$$

$k = 1, 2, \ldots, n$, $n \to \infty$.

Szegö proved that $\epsilon_n(x) \to 0$ uniformly as $O((\ln n)^{-\epsilon})$ if $p(x)$ is Riemann integrable and is of the form (see [17])

$$p(x) = w(x)|x - x_1|^{2\tau_1} \cdots |x - x_l|^{2\tau_l},$$

where $\tau_i > 0$, $|x_i| \leq 1$, $0 < const_1 \leq w(x) \leq const_2$. If this is the case the $\epsilon_n(x) \to 0$ pointwise for $n \to \infty$.

Let us consider the special case of a polynomial weight function

$$w_p(x) = \prod_{i=1}^{p}(x - a_i).$$

In this case the function ψ is the sum of the functions ψ_i, which possess an elementary integral

$$\begin{aligned} \psi_i(\theta) &= \frac{1}{2\pi}\int_{-1}^{1} \frac{\ln(z-a_i)^2 - \ln(x-a_i)^2}{z-x}\sqrt{\frac{1-x^2}{1-z^2}}dz \\ &= 2\arctan\Big(\frac{\sin\theta}{\cos\theta - \rho_i}\Big), \quad i = 1,\ldots,p \end{aligned} \tag{3.5}$$

where $\rho_i = a_i + \sqrt{a_i^2 - 1}$. Formula (3.5) furnishes excellent initial approximations of the roots β_i of $P_n(x)$ with the weight function $w_p(x)$ as solutions of the equations

$$\begin{aligned} \theta_k &= \frac{\pi(k-0,5)}{n} - \frac{1}{n}\sum_{i=1}^{p}\psi_i(\theta_k), \quad k = 1,\ldots,n. \\ \beta_k &= \cos\theta_k \end{aligned} \tag{3.6}$$

One can solve these equations by some iteration method, for example, by

$$\begin{aligned} \theta_k^{j+1} &= \frac{\pi(k-0,5)}{n} - \frac{1}{n}\sum_{i=1}^{p}\psi_i(\theta_k^j), \quad k = 1,\ldots,n, \\ \beta_k^{j+1} &= \cos\theta_k^{j+1}, \end{aligned} \tag{3.7}$$

starting from

$$\theta_k^1 = \frac{\pi(k-0,5)}{n}, \quad k = 1,\ldots,n.$$

The rate of convergence of the algorithm speeds up with n. This property follows immediately from (3.7). Then program BERN improves iteratively these solutions which leads in a few steps to the solution of (3.1) (see Lebedev [10] for details).

We remark that program BERN can treat also more general weight functions such as the product of exponential functions with argument as Chebyshev series, spline interpolations and Lagrange interpolations. Moreover, the program can construct the polynomials of least deviation from smooth functions.

4 Construction of *p*th order explicit composition methods.

After having constructed pth order stability polynomials with maximal stability region in Chapter 2, we shall now construct pth order Runge–Kutta methods

for these stability polynomials. For this aim, we shall use the theory of composition methods.

Let us consider two consecutive steps by explicit methods **A** and **B**:

A:

$$\begin{aligned} \hat{Y}_i &= y_0 + h\sum_{j=1}^{s_1} \hat{a}_{ij} f(t_0 + \hat{c}_j h, \hat{Y}_j), \quad i = 1, \ldots, s_1, \\ \tilde{y}_0 &= y_0 + h\sum_{j=1}^{s_1} \hat{b}_j f(t_0 + \hat{c}_j h, \hat{Y}_j), \quad \hat{a}_{ij} = 0 \text{ if } i \le j, \end{aligned} \tag{4.1}$$

B:

$$\begin{aligned} \tilde{Y}_i &= \tilde{y}_0 + h\sum_{j=1}^{s_2} \tilde{a}_{ij} f(t_0 + \tilde{c}_j h, \tilde{Y}_j), \quad i = 1, \ldots, s_2, \\ y_1 &= \tilde{y}_0 + h\sum_{j=1}^{s_2} \tilde{b}_j f(t_0 + \tilde{c}_j h, \tilde{Y}_j), \quad \tilde{a}_{ij} = 0 \text{ if } i \le j. \end{aligned} \tag{4.2}$$

DEFINITION 4.1. We will call the method which is the result of one step of **A** and one step of **B** as the composition method **C** =**B**(**A**):

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(t_0 + c_j h, Y_j), \quad i = 1, \ldots, s, \\ y_1 &= y_0 + h\sum_{j=1}^{s} b_j f(t_0 + c_j h, Y_j), \quad a_{ij} = 0 \text{ if } i \le j, \end{aligned} \tag{4.3}$$

where $s = s_1 + s_2$.

The stability function of the composition method **C** is the product

$$R_s(z) = Q_{s_2}(z) P_{s_1}(z), \tag{4.4}$$

where $Q_{s_2}(z)$ and $P_{s_1}(z)$ are the stability functions of **B** and **A** respectively. For the method **C** with coefficients a_{ij}, b_j must hold the usual order conditions

$$\begin{aligned} p=1: \quad c(\tau) &:= \sum_{i}^{s} b_i = 1, \\ p=2: \quad c(t_{21}) &:= 2\sum_{i,j=1}^{s} b_i a_{ij} = 2\sum_{i=1}^{s} b_i c_i = 1, \\ p=3: \quad c(t_{32}) &:= 6\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} = 6\mathbf{b}^t A^2 \mathbf{e} = 1, \\ c(t_{31}) &:= 3\sum_{i,j,k=1}^{s} b_i a_{ij} a_{ik} = 3\sum_{i=1}^{s} b_i c_i^2 = 1, \\ p=4: \quad c(t_{44}) &:= 24\sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{jk} a_{kl} = 24\mathbf{b}^t A^3 \mathbf{e} = 1, \\ c(t_{43}) &:= 12\sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{jk} a_{jl} = 12\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = 1, \\ c(t_{42}) &:= 8\sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{ik} a_{jl} = 8\sum_{i,j=1}^{s} b_i c_i a_{ij} c_j = 1, \\ c(t_{41}) &:= 4\sum_{i,j,k,l=1}^{s} b_i a_{ij} a_{ik} a_{il} = 4\sum_{i=1}^{s} b_i c_i^3 = 1. \end{aligned} \tag{4.5}$$

For the solution of our problem, we take some pth order optimal polynomial $R_s(z)$ (from Chapter 2) and divide the set of its roots in two subsets. First we choose $s_1 = s - p$ roots for the stability polynomial $P_{s_1}(z)$ of method **A**, and the remaining $p = s_2$ roots for the polynomial $Q_{s_2}(z)$:

$$R_s(z) = Q_{s_2}(z)P_{s_1}(z) = 1 - z + \ldots + (-1)^p z^p/p! + \sum_{i=p+1}^{s} d_i z^i.$$

Any stability function of a method with $p \geq 2$ has at least two complex conjugate roots. We include all complex roots into the stability function of method **B**. One can express the order conditions of a composition of methods in terms of the parameters of the submethods **A** and **B** as follows (see Hairer, Nørsett, and Wanner [2, §II.11]):

$$\begin{aligned}
c(\tau) &= b(\tau) + a(\tau),\\
c(t_{21}) &= b(t_{21}) + 2b(\tau)a(\tau) + a(t_{21}),\\
c(t_{31}) &= b(t_{31}) + 3b(\tau)a^2(\tau) + 3b(t_{21})a(\tau) + a(t_{31}),\\
c(t_{32}) &= b(t_{32}) + 3b(\tau)a(t_{21}) + 3b(t_21)a(\tau) + a(t_{32}),\\
c(t_{41}) &= b(t_{41}) + 4b(\tau)a^3(\tau) + 6b(t_{21})a^2(\tau) + 4b(t_{31})a(\tau) + a(t_{41}),\\
c(t_{42}) &= b(t_{42}) + 4b(\tau)a(t_{21})a(\tau) + 6\big(\tfrac{1}{3}b(t_{21})a(t_{21}) + \tfrac{2}{3}b(t_{21})a^2(\tau)\big)\\
&\quad + 4\big(\tfrac{2}{3}b(t_{31})a(\tau) + \tfrac{1}{3}b(t_{32})a(\tau)\big) + a(t_{42}),\\
c(t_{43}) &= b(t_{43}) + 4b(\tau)a(t_{31}) + 6b(t_{21})a^2(\tau) + 4b(t_{32})a(\tau) + a(t_{43}),\\
c(t_{44}) &= b(t_{44}) + 4b(\tau)a(t_{32}) + 6b(t_{21})a(t_{21}) + 4b(t_{32})a(\tau) + a(t_{44}),
\end{aligned} \tag{4.6}$$

where $c(t)$ are defined in (4.5) and $a(t)$ respectively $b(t)$ are the analogous expressions for methods **A** and **B**. One can rewrite the order conditions (4.5) as follows

order	*conditions*
$p = 1$	$c(\tau) = 1,$
$p = 2$	$c(t_{21}) = 1,$
$p = 3$	$c(t_{31}) = 1, \quad c(t_{32}) = 1,$
$p = 4$	$c(t_{41}) = 1, \quad c(t_{42}) = 1, \quad c(t_{43}) = 1, \quad c(t_{44}) = 1.$

(4.7)

We construct the pth order composition method as follows. We first choose some $s - p$ stage method **A** (with stability polynomial $P_{s-p}(z)$ from (4.4)). For example, we can take method **A** as a sequence of $s-p$ steps of Lebedev's method (1.8) with the parameters (take even values $s - p$ here)

$$\begin{aligned}
\alpha_i &= \frac{1}{2}\left(\frac{1}{\tilde\gamma_{i+p}} + \frac{1}{\tilde\gamma_{s-i+1}}\right), \quad i = 1, \ldots, (s-p)/2,\\
\nu_i &= 1 - 4\frac{\tilde\gamma_{i+p}\tilde\gamma_{s-i+1}}{(\tilde\gamma_{i+p} + \tilde\gamma_{s-i+1})^2},
\end{aligned}$$

where $\tilde\gamma_i$ are the roots of the stability polynomial $P_{s-p}(z)$. In this case the table

of the method **A** is the following:

$$
\begin{array}{c|cccc}
0 & 0 & 0 & \cdots & 0 \\
\alpha_1 & \alpha_1 & 0 & \cdots & 0 \\
2\alpha_1 & (1+\nu_1)\alpha_1 & (1-\nu_1)\alpha_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
. & (1+\nu_1)\alpha_1 & \cdots & \alpha_{\frac{s-p}{2}} & 0 \\
\hline
\alpha & (1+\nu_1)\alpha_1 & \cdots & (1+\nu_{\frac{s-p}{2}})\alpha_{\frac{s-p}{2}} & (1-\nu_{\frac{s-p}{2}})\alpha_{\frac{s-p}{2}}
\end{array}
\tag{4.8}
$$

where

$$\alpha = 2 \sum_{i=1}^{(s-p)/2} \alpha_i = -P'_{s-p}(0).$$

Now we can simply calculate the expressions $a(\tau), a(t_{21}), \ldots, a(t_{44})$. The equations (4.6) and the order conditions (4.7) now allow to determine the expressions $b(\tau), b(t_{21}), \ldots, b(t_{44})$:

$$
\begin{aligned}
p=1: \quad b(\tau) &= 1 - a(\tau), \\
p=2: \quad b(t_{21}) &= 1 - 2b(\tau)a(\tau) - a(t_{21}), \\
p=3: \quad b(t_{31}) &= 1 - 3b(\tau)a^2(\tau) - 3b(t_{21})a(\tau) - a(t_{31}), \\
b(t_{32}) &= 1 - 3b(\tau)a(t_{21}) + 3b(t_21)a(\tau) + a(t_{32}), \\
p=4: \quad b(t_{41}) &= 1 - 4b(\tau)a^3(\tau) - 6b(t_{21})a^2(\tau) - 4b(t_{31})a(\tau) - a(t_{41}), \\
b(t_{42}) &= 1 - 4b(\tau)a(t_{21})a(\tau) - 6\big(\tfrac{1}{3}b(t_{21})a(t_{21}) + \tfrac{1}{3}2b(t_{21})a^2(\tau)\big) \\
&\quad -4\big(\tfrac{2}{3}b(t_{31})a(\tau) + \tfrac{1}{3}b(t_{32})a(\tau)\big) - a(t_{42}\big), \\
b(t_{43}) &= 1 - 4b(\tau)a(t_{31}) - 6b(t_{21})a^2(\tau) - 4b(t_{32})a(\tau) - a(t_{43}), \\
b(t_{44}) &= 1 - 4b(\tau)a(t_{32}) - 6b(t_{21})a(t_{21}) - 4b(t_{32})a(\tau) - a(t_{44}).
\end{aligned}
\tag{4.9}
$$

The values $b(\tau)$, $b(t_{21})$, $b(t_{32})$ and $b(t_{44})$ are the p coefficients of the stability polynomial $Q_p(z) = \frac{R_s(z)}{P_{s-p}(z)}$. Consequently, if we find the parameters of the method **B** from equations (4.9), method $\mathbf{C} = \mathbf{B}(\mathbf{A})$ will have the stability polynomial $R_s(z)$ with the optimised stability region and it will have order p. We have computed the parameters $\tilde{b}_i$ and $\tilde{a}_{i,j}$ from equations (4.9) for $p \leq 4$ and $s \leq 243$ (see the next section).

5 Embedded explicit composition method.

To write a reliable code one needs to write an error estimation procedure. For this, we constructed embedded formulas of order $p-1$ which allows a step size control procedure (see Hairer, Nørsett, and Wanner [2, §II.4]).

We have to find formulas for the coefficients $\tilde{a}_{ij}$, $\tilde{b}_j$ of a method **B** for which $\mathbf{C} = \mathbf{B}(\mathbf{A})$ will be of order p, together with coefficients $\hat{b}_i$ of an embedded method $\mathbf{B}'$ for which $\mathbf{C}' = \mathbf{B}'(\mathbf{A})$ is of order $p-1$.

THIRD ORDER METHOD:
The table of the method **B** is

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \tilde{c}_2 & \tilde{a}_{21} & 0 & 0 \\ \tilde{c}_3 & \tilde{a}_{31} & \tilde{a}_{32} & 0 \\ \hline & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 \\ & \bar{b}_1 & \bar{b}_2 & \bar{b}_3 \quad \bar{b}_4 \end{array}.$$

In this case equations (4.9) are

$$\begin{cases} \tilde{b}_1+\tilde{b}_2+\tilde{b}_3 = b(\tau) \\ \tilde{b}_2\tilde{c}_2+\tilde{b}_3\tilde{c}_3 = b(t_{21})/2 \\ \tilde{b}_2\tilde{c}_2^2+\tilde{b}_3\tilde{c}_3^2 = b(t_{31})/3 \\ \tilde{b}_3\tilde{a}_{32}\tilde{c}_2 = b(t_{32})/6 \end{cases} \quad \text{and} \quad \begin{cases} \bar{b}_1+\bar{b}_2+\bar{b}_3+\bar{b}_4 = b(\tau) \\ \bar{b}_2\tilde{c}_2+\bar{b}_3\tilde{c}_3+\bar{b}_4 * b(\tau) = b(t_{21})/2 \end{cases}.$$

The first system has four equations and six variables, the second one two equations and four variables. Consequently, one can add four equations to obtain some additional properties of the method, like internal stability, economic storage and so on.

We optimize the error control procedure by taking into account the formulas for the stability functions of the method $\mathbf{C} = \mathbf{B}(\mathbf{A})$, of the embedded method $\mathbf{C'} = \mathbf{B'}(\mathbf{A})$ and the difference $E(z) = R_s(z) - \bar{R}_{s+1}(z)$:

$$\begin{aligned} R_s(z) &= \left(1 + d_1 z + \sum_{k=2}^{p} d_k z^k/k!\right) P_{s-p}(z), \\ \bar{R}_{s+1}(z) &= \left(1 + \bar{d}_1 z + \sum_{k=2}^{p+1} \bar{d}_k z^k/k!\right) P_{s-p}(z), \\ E(z) &= \frac{z^p}{p!}\left((d_p - \bar{d}_p) - \frac{\bar{d}_{p+1} z}{p+1}\right) P_{s-p}(z), \end{aligned} \tag{5.1}$$

where $P_{s-p}(z)$ is the stability function of the method **A**, d_i and $\bar{d}_i$ ($d_i = \bar{d}_i, i = 1, \ldots, p-1$) are the coefficients of the stability function of methods **B** and **B'** respectively. The coefficients $\bar{d}_i$ are linear functions of the parameters $\bar{b}_j$, $j = 1, \ldots, p+1$. Consequently, some requirement for the stability function of the method **B'**: $\bar{R}_{p+1}(x) = y$ gives us linear equation for the coefficients $\bar{b}_j$.

For example, one can take additional equations as follows:

$$\begin{aligned} \tilde{a}_{31} &= \tilde{b}_1, \qquad \tilde{a}_{32} = \tilde{b}_2, \\ \bar{R}_{s+1}(x_1) &= 0, \quad \bar{R}_{s+1}(x_2) = 0, \end{aligned} \tag{5.2}$$

where x_1 is a real root of the stability function of method **B** and $x_2 = \beta_3(s)$ a bound for the stability region. This choice is taken in program DUMKA. Program DUMKA uses only three arrays of storage. The stability functions $R_s(z)$, $\bar{R}_{s+1}(z)$ and $E(z)$ are shown in figure 5.1 (left) for $s = 9$.

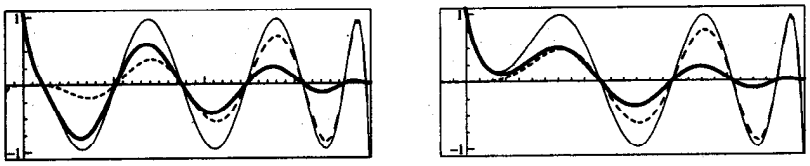

Figure 5.1: Stability polynomials of the method **C** (thin line), embedded method **C** (thick line) and its difference $E(z)$ (dashed line).

FOURTH ORDER METHOD:
The table of method **B** is

0	0	0	0	0	
$\tilde{c}_2$	$\tilde{a}_{21}$	0	0	0	
$\tilde{c}_3$	$\tilde{a}_{31}$	$\tilde{a}_{32}$	0	0	
$\tilde{c}_4$	$\tilde{a}_{41}$	$\tilde{a}_{42}$	$\tilde{a}_{43}$	0	
	$\tilde{b}_1$	$\tilde{b}_2$	$\tilde{b}_3$	$\tilde{b}_4$	
	$\bar{b}_1$	$\bar{b}_2$	$\bar{b}_3$	$\bar{b}_4$	$\bar{b}_5$

In this case equations (4.9) are

$$\begin{cases} \tilde{b}_1+\tilde{b}_2+\tilde{b}_3+\tilde{b}_4 = b(\tau) \\ \tilde{b}_2\tilde{c}_2+\tilde{b}_3\tilde{c}_3+\tilde{b}_4\tilde{c}_4 = b(t_{21})/2 \\ \tilde{b}_2\tilde{c}_2^2+\tilde{b}_3\tilde{c}_3^2+\tilde{b}_4\tilde{c}_4^2 = b(t_{31})/3 \\ \tilde{b}_2\tilde{c}_2^3+\tilde{b}_3\tilde{c}_3^3+\tilde{b}_4\tilde{c}_4^3 = b(t_{41})/4 \\ \tilde{b}_3\tilde{a}_{32}\tilde{c}_2+\tilde{b}_4(\tilde{a}_{42}\tilde{c}_2+\tilde{a}_{43}\tilde{c}_3) = b(t_{32})/6 \\ \tilde{b}_3\tilde{c}_3\tilde{a}_{32}\tilde{c}_2+\tilde{b}_4\tilde{c}_4(\tilde{a}_{42}\tilde{c}_2+\tilde{a}_{43}\tilde{c}_3) = b(t_{42})/8 \\ \tilde{b}_3\tilde{a}_{32}\tilde{c}_2^2+\tilde{b}_4(\tilde{a}_{42}\tilde{c}_2^2+\tilde{a}_{43}\tilde{c}_3^2) = b(t_{43})/12 \\ \tilde{b}_4\tilde{a}_{43}\tilde{a}_{42}\tilde{c}_2 = b(t_{44})/24 \end{cases}$$

and

$$\begin{cases} \bar{b}_1+\bar{b}_2+\bar{b}_3+\bar{b}_4+\bar{b}_5 = b(\tau) \\ \bar{b}_2\tilde{c}_2+\bar{b}_3\tilde{c}_3+\bar{b}_4\tilde{c}_4+\bar{b}_5 * b(\tau) = b(t_{21})/2 \\ \bar{b}_2\tilde{c}_2^2+\bar{b}_3\tilde{c}_3^2+\bar{b}_4\tilde{c}_4^2+\bar{b}_5 * b^2(\tau) = b(t_{31})/3 \\ \bar{b}_3\tilde{a}_{32}\tilde{c}_2+\bar{b}_4(\tilde{a}_{42}\tilde{c}_2+\tilde{a}_{43}\tilde{c}_3)+\bar{b}_5(\tilde{b}_2\tilde{c}_2+\tilde{b}_3\tilde{c}_3+\tilde{b}_4\tilde{c}_4) = b(t_{32})/6 \end{cases}$$

The first system has eight equations and ten variables, the second one has fou equations and five variables. We can choose two parameters in the first syster and one parameter in the second. To obtain the solution of the first system on can use the same ideas as in the case of the usual fourth order method with $b(\tau)$ $b(t_{ij}) = 1$.

The additional equations for program DUMKA4 are the following:

$$c_2 = b(\tau)/3, \quad c_3 = 2b(\tau)/3, \quad \bar{R}_{s+1}(x_1) = 0, \tag{5.3}$$

where $x_1 = \beta_4(s)$ is a bound for the stability region. The stability function $R_s(z)$, $\bar{R}_{s+1}(z)$ and $E(z)$ are shown in the Figure 5.1 (right) for $s = 9$.

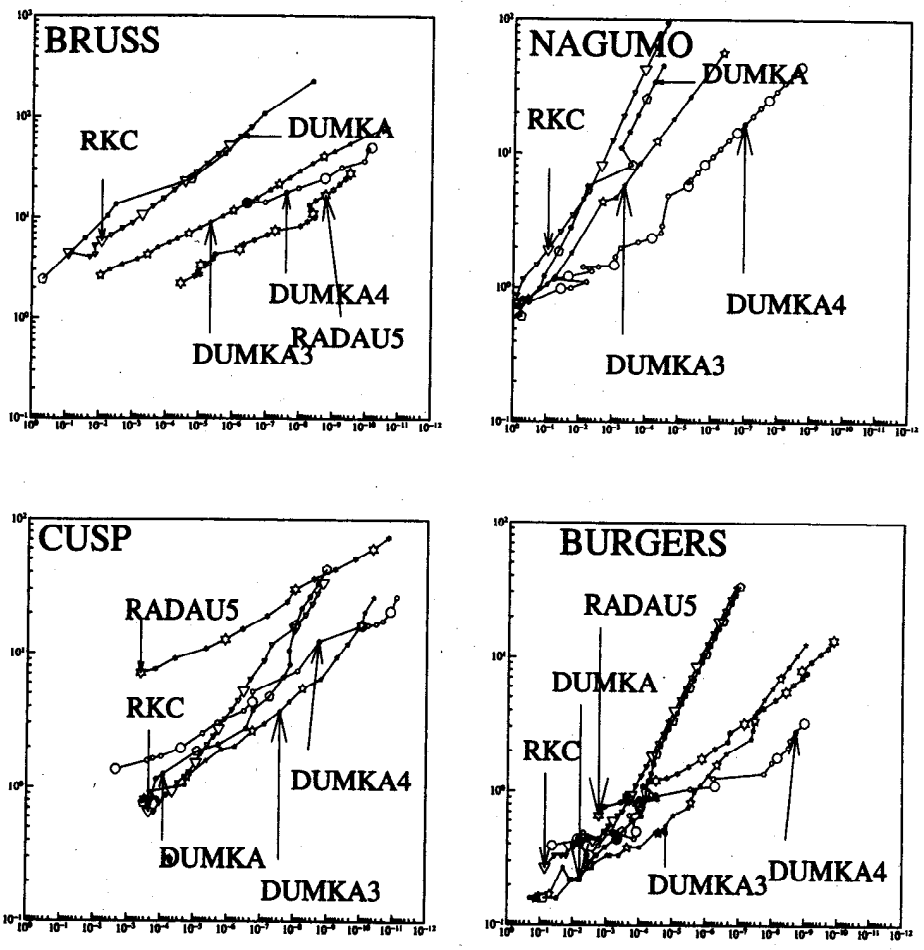

Figure 6.1: Actual obtained precision versus computing time of the tested codes.

Finally the code uses a standard step-size control procedure (see e.g. Hairer, Nørsett, and Wanner [2, §II. 4], or Hairer and Wanner [3, §IV.2]):

$$
\begin{aligned}
err_1 &= \|y_1 - \bar{y}_1\|, \\
h_{new} &= h \min(fac_{max}, \max(fac_{min}, fac(tol/err_1)^{1/p})) \quad \text{or} \\
h_{new} &= h \min(fac_{max}, \max(fac_{min}, fac(tol/err_1)^{\alpha/p}(err_0/tol)^{\beta/p})),
\end{aligned} \tag{5.4}
$$

where $\|\cdot\|$ is a some norm in R^n, h is the step-size of the previous step, fac_{min} and fac_{max} some factors, tol is the tolerance which is required for our calculations, err_0 is an error in the previous step, and α and β are parameters of the step-size control procedure.

6 Numerical results.

We have used the test problems Brusselator, Nagumo, Cusp and Burgers from the book of Hairer and Wanner [3, first edition; p.167]. We took four programs RKC (Sommeijer [6]), DUMKA [11] (first and second order explicit method) and DUMKA3[2] (third order method), DUMKA4 (fourth order method with the coefficients computed above) and solved these problems with varying values of *tol*. The results are represented in Figure 6.1 in logarithmic axes (in the abscissa the accuracy, in the ordinates the computing time in seconds).

The codes DUMKA3 and DUMKA4 are provided with variable step size control. They adjust the number of stages s automatically as to satisfy in each integration step the stability condition $h\sigma(f'(y)) \leq \beta_p(s)$ where $\sigma(f'(y))$ is the spectral radius of the Jacobian matrix. DUMKA3 has two realizations with 3 and 5 arrays of storage respectively and uses stability polynomials of degrees s up to $s_{max} = 243$ with stability region $\beta_3(s_{max}) \approx 29\,376$. The use of the programs DUMKA3 and DUMKA4 requires two SUBROUTINEs: a subroutine for calculation of the right hand of the equation and a program for an upper estimation of $\sigma(f'(y))$, which normally renders no problem.

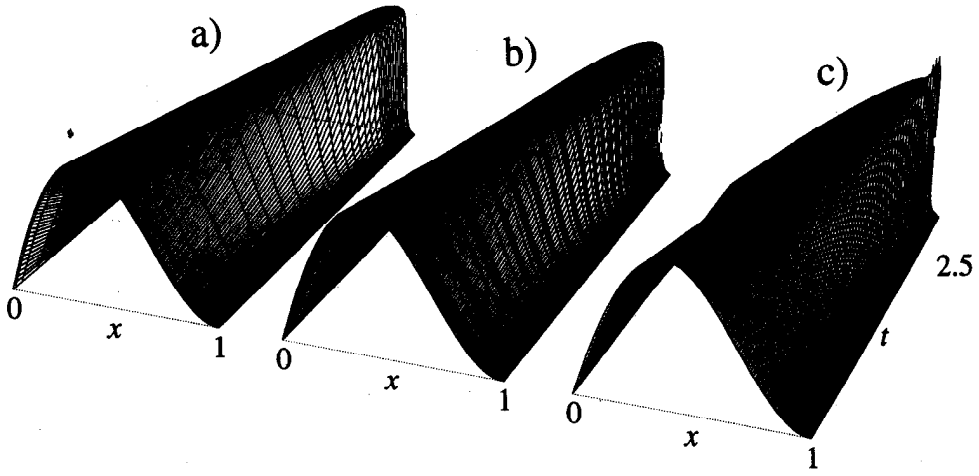

Figure 6.2: Solution of Burgers equation by DUMKA3 (step size and each function evaluation) and Euler method (with: a) every computed step, b) every function evaluation, c) every Euler step).

Figure 6.2 presents an illustration of the performance of the code DUMKA3 compared to the explicit Euler method. We choose the semi-discrete viscous

[2]Source codes for DUMKA3 and DUMKA4 and some examples can be obtained from

Burgers equation

$$\frac{du_k}{dt} = -\frac{u_{k+1}^2 - u_{k-1}^2}{4\Delta x} + \mu\frac{u_{k-1} - 2u_k + u_{k+1}}{\Delta x^2},$$

$$u(0) = 1.5x(1-x)^2, \quad \mu = 0.0005, \quad x \in [0,1], \quad t \in [0,2.5].$$

It can be observed how for $\Delta x = 1/150$ DUMKA3 adjusts the values of s from $s = 3$ in the starting step, to $s = 6$ and $s = 9$ in the smooth phase, and finally, after appearance of the shock, reduces s to 6 and 3 again. For smaller value of Δx (e.g. $\Delta x = 1/5000$) the largest value of s reaches 243.

In general, the mean time step (per one function evaluation) for parabolic equations in Euler's method is approximately $\bar{h}_{eul} = O(\Delta x^2/\mu)$, while for DUMKA3 we have $\bar{h}_{dum} = O(\Delta x/\sqrt{\mu})$.

Acknowledgements.

This article was written with the kind help of professors V. I. Lebedev, G. Wanner, and E. Hairer. Many thanks also to an anonymous referee for valuable suggestions.

REFERENCES

1. S. N. Bernstein, *Leçons sur les propriétés extrémales et la meilleure approximation des fonctions analytiques d'une variable réelle*, in Professées á la Sorbonne, Paris, 1926.
2. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, 2nd ed., Springer-Verlag, Berlin, 1993.
3. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.
4. P. J. van der Houwen and J. Kok, *Numerical solution of a maximal problem*, Report TW 124/71, Mathematical Center, Amsterdam, 1971.
5. P. J. van der Houwen, *Construction of Integration Formulas for Initial Value Problems*, North-Holland, Amsterdam, 1977.
6. P. J. van der Houwen and B. P. Sommeijer, *On the internal stability of explicit m-stage Runge–Kutta methods for large m-values*, Z. Angew. Math. Mech., 60 (1980), pp. 479–485.
7. V. I. Lebedev and A. A. Medovikov, *Methods of second order accuracy with variable time steps*, Izvestiya Vuzov, Matematika, N10, Russia, 1995.
8. V. I. Lebedev, *How to solve stiff systems of differential equations by explicit methods*, in Numerical Methods and Applications, CRC Press, Boca Raton, 1994, pp. 45–80.
9. V. I. Lebedev, *Zolotarev polynomials and extremum problem*, Russ. J. Numer. Anal. Math. Modelling. V. 9, N 3 (1994), pp. 191–314.
10. V. I. Lebedev, *A new method for determining the roots of polynomials of least deviation on a segment with weight and subject to additional conditions. Part I, Part II*, Russ. J. Numer. Anal. Math. Modelling. V. 8, N 3 (1993), pp. 195–222; V. 8, N 5 (1993), pp. 397–426.

11. V. I. Lebedev, *Explicit difference schemes with time-variable steps for the solution of stiff system of equations*, Preprint DNM AS USSR N177, 1987.
12. H. Lomax, *On the construction of highly stable, explicit numerical methods for integrating coupled ordinary differential equations with parasitic eigenvalues*, NASA Technical Note NASATND/4547, 1968.
13. A. A. Markov, *Lectures on the functions of least deviation*, 1892; in Selected Works, OGIZ, Moscow, 1948, pp. 244–291, (in Russian).
14. A. A. Medovikov, *Explicit and explicit-implicit methods of solution of unstationary problems*, PhD Thesis, INM, Moscow, 1992.
15. C. L. Metzger, *Méthodes de Runge–Kutta de rang supérieur à l'orde*, Thèse (troisième cycle), Université de Grenoble, France, 1967.
16. W. Riha, *Optimal stability polynomials*, Computing, 9 (1972), pp. 37–43.
17. G. Szegö, *Orthogonal Polynomials*, Amer. Math. Soc., New York, 1949.
18. P. L. Chebyshev, *On the problems about minimal values connected with appropriate representation of functions*, Mem. Acad. St. Petersburg (1859); Collected Works, Vol. 2, GITTL, Moscow–Leningrad, 1947 (in Russian); Oeuvres, Vol. I, p. 271.
19. P. L. Chebyshev, *On the functions slightly deviating from zero for certain values of the variables*, Prilojenie k XI tomy Zapisok Imperatorskoi Akademii nauk N3 (1881); *Collected Works*, Vol. 3, GITTL, Moscow–Leningrad, 1948, p.108; Oeuvres, Vol. II, p. 373 (in Russian).
20. J. G. Verwer, *A class of stabilized three-step Runge–Kutta methods for the numerical integration of parabolic equations*, J. Comput. Appl. Math., 3 (1977), pp. 155–166.
21. J. G. Verwer, *Explicit Runge–Kutta methods for parabolic partial differential equations*, 1996. To appear in the Runge–Kutta centennial special issue of Appl. Numer. Math.
22. E. I. Zolotarev, *On applying elliptic functions to the study of functions of least and greatest deviation from zero*, Zapiski Sankt-Peterburgskoi Akademii Nauk XXX, No 5 (1887), pp. 1–59 (in Russian).