stichting
mathematisch
centrum

MC

AFDELING TOEGEPASTE WISKUNDE TW

TW 124/71 APRIL

P.J. VAN DER HOUWEN

STABILIZED RUNGE-KUTTA METHODS WITH LIMITED STORAGE REQUIREMENTS

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

Contents

1. Introduction

In [1] and [2] explicit one-step methods were investigated for the numerical integration of initial value problems for linear equations of the type

$$\frac{d\tilde{U}}{dt} = D\tilde{U} + F, \tag{1.1}$$

where $\tilde{U}$ and F are (vector) functions of the variable t and D is a matrix with constant coefficients. These methods are based on repeated differentiation with respect to t of the right hand side of (1.1).

When initial value problems are to be solved for non-linear equations of the type

$$\frac{d\tilde{U}}{dt} = H(\tilde{U},t), \tag{1.2}$$

similar methods can be used, provided that the derivatives of $H(\tilde{U}(t),t)$ with respect to t can be expressed in the preceding ones. Locally, the stability conditions for equations of type (1.2) are the same as the conditions for (1.1).

When the function $H(\tilde{U},t)$ cannot be easily differentiated one may resort to Runge-Kutta type methods which are exclusively based on evaluations of the right hand side of (1.2). In this paper we shall derive n-point Runge-Kutta formulae which have an accuracy of order p, p = 1,2,3,4, and a stability polynomial of the type

$$P_n(z) = 1 + z + \frac{1}{2!} z^2 + \ldots + \frac{1}{p!} z^p + \beta_{p+1} z^{p+1} + \ldots + \beta_n z^n,$$

where the coefficients $\beta_{p+1}, \ldots, \beta_n$ may be chosed arbitrarily. This property enables us to apply the stability theory, developed in [1], [2], to Runge-Kutta schemes.

A second feature of the formulae given in this paper are the limited storage requirements of the computational scheme, which make them appropriate for integrating large systems of equations such as the systems arising from partial differential equations.

Finally, by choosing n sufficiently large, formulae are derived which approximate the first neglected terms in the Taylor expansion of the local analytical solution. These formulae can be used for estimating the local error of the method. Furthermore, when variable steps are to be used, one may base a step size strategy on monitoring this estimate for the local error. In connection with this it may be remarked, that in using standard Runge-Kutta formulae, only approximations of the last correction terms are available, which is a rather pessimistic starting point for step size prediction (cf [5]).

This paper is concluded with the explicit formulation of a four-point, first order exact and a five-point, second order exact Runge-Kutta formula which may be usefull in integrating respectively non linear parabolic and non linear hyperbolic differential equations.

2. Polynomial methods

Consider the non-linear initial value problem

$$\begin{cases} \dfrac{d\tilde{U}}{dt} = H(t,\tilde{U}),\ t \geq 0, \\ \tilde{U} = \tilde{U}_0,\ t = 0, \end{cases}$$

where $\tilde{U}_0$ is a given initial function. Suppose that H has continuous derivatives with respect to t and $\tilde{U}$ of up to order n. Then, in analogy to the scheme for linear problems given in [1], formula (2.8'), we may define the p-th order exact scheme

$$(2.2)\quad \begin{cases} u_0 = \tilde{U}_0, \\ u_{k+1} = u_k + \tau c_k^{(1)} + \frac{1}{2}\tau^2 c_k^{(2)} + \ldots + \frac{1}{p!}\tau^p c_k^{(p)} + \beta_{p+1}\tau^{p+1} c_k^{(p+1)} + \ldots \\ \qquad\qquad \ldots + \beta_n \tau^n c_k^{(n)},\ k = 0,1,2,\ldots, \\ c_k^{(j)} = \dfrac{d^{j-1}}{dt^{j-1}} H\Big|_{(t_k,u_k)}. \end{cases}$$

Here, u_k denotes the difference solution at $t = t_k = k\tau$.

Of course, schemes of this type only are of practical value when the derivatives of H can be obtained easily.

As in the linear case, the stability of (2.2) is investigated by considering the generating polynomial

$$(2.3)\qquad P_n(z) = 1 + z + \frac{1}{2}z^2 + \ldots + \frac{1}{p!}z^p + \beta_{p+1}z^{p+1} + \ldots + \beta_n z^n.$$

This is suggested by locally linearizing equation (2.1). In doing so we obtain

$$(2.4)\qquad \frac{d\tilde{U}}{dt} = D_k\tilde{U} + F_k(t),$$

where

$$F_k(t) \tilde{=} H(t_k,\tilde{U}(t_k)) + (t-t_k)\, H_t(t_k,\tilde{U}(t_k)) - D_k\tilde{U}(t_k)$$

and D_k is the matrix (d_{ij}) with

$$d_{ij} = \frac{\partial}{\partial \tilde{U}^{(j)}} H^{(i)}(t_k,\tilde{U}(t_k)).$$

In this expression $\tilde{U}^{(j)}$ and $H^{(i)}$ respectively denote the j-th and i-th component of the (vector)functions $\tilde{U}$ and H.
By applying the linear stability theory to equation (2.4) we arrive at polynomial (2.3) as the polynomial which governs the local stability. Although representation (2.4) is only approximate and, therefore, stability considerations based on (2.4) are not rigorous, the stability conditions obtained in this way are quite satisfactory in actual applications. For a discussion of the problem how to choose the polynomial $P_n(z)$ for a particular equation we refer to references [1] and [2].

It may be remarked that, centrary to the linear case, the stability condition associated to non-linear problems will depend on t_k since the eigenvalue spectrum of D_k may change with t_k.

3. Runge-Kutta type schemes

In many cases the function $H(t,\tilde{U})$ is a complicated one and derivatives of H are not easily obtained. To overcome this difficulty Runge [4] proposed a computational scheme which only requires the evaluation of the function H at, say n, points per timestep.

3.1. General structure of the scheme

The general n-point formula is of the form [4]

$$
(3.1)\quad \begin{cases}
u_0 = \tilde{U}_0, \\
u_{k+1} = u_k + \theta_0 r_k^{(0)} + \ldots + \theta_{n-1} r_k^{(n-1)}, \quad k = 0,1,2,\ldots, \\
r_k^{(0)} = \tau H(t_k, u_k), \\
r_k^{(1)} = \tau H(t_k+\mu_1\tau, u_k+\lambda_{10} r_k^{(0)}), \\
\qquad\qquad \ldots \\
r_k^{(j)} = \tau H(t_k+\mu_j\tau, u_k+\lambda_{j0} r_k^{(0)}+\ldots+\lambda_{jj-1} r_k^{(j-1)}), \\
\qquad\qquad \ldots \\
r_k^{(n-1)} = \tau H(t_k+\mu_{n-1}\tau, u_k+\lambda_{n-10} r_k^{(0)}+\ldots+\lambda_{n-1n-2} r_k^{(n-2)}).
\end{cases}
$$

This scheme can be characterized by the matrix

$$
(3.2)\quad R = \begin{bmatrix}
\mu_1 & \lambda_{10} & 0 & \ldots & 0 \\
\mu_2 & \lambda_{20} & \lambda_{21} & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mu_{n-1} & \lambda_{n-10} & \lambda_{n-11} & \ldots & \lambda_{n-1n-2} \\
\theta_0 & \theta_1 & \theta_2 & \ldots & \theta_{n-1}
\end{bmatrix}.
$$

In the following subsections we shall derive consistency and local stability conditions. These conditions will lead to relations between the entries of the matrix R. The final values of the parameters are determined by considerations of simplicity and storage requirements.

3.2. Consistency conditions

The vectors $r_k^{(j)}$, $j = 0,1,\ldots,n-1$, introduced in the preceding subsection are functions of the step τ. By expanding these vectors in a Taylor series with respect to τ (it is assumed that H has derivatives of sufficiently high order) we can set up the polynomial approximation in τ of u_{k+1}. By identifying the first p+1 terms of this polynomial with the Taylor series in τ of u_{k+1} we obtain the consistency conditions for p-th order accuracy. In carrying out this program one usually makes the assumptions (cf. [5])

$$\sum_{l=0}^{j-1} \lambda_{jl} = \mu_j, \quad j = 1,2,\ldots,n-1. \tag{3.3}$$

In this paper we always assume that these conditions are satisfied.

A straightforward calculation yields

$$\left\{\begin{aligned}
r_k^{(0)} &= \tau c_k^{(1)}, \\
r_k^{(1)} &= \tau c_k^{(1)} + \mu_1\tau^2 c_k^{(2)} + \tfrac{1}{2}\mu_1^2\tau^3(c_k^{(3)} - D_k c_k^{(2)}) + \ldots, \\
r_k^{(2)} &= \tau c_k^{(1)} + \mu_2\tau^2 c_k^{(2)} + \tfrac{1}{2}\mu_2^2\tau^3(c_k^{(3)} - D_k c_k^{(2)}) + \\
&\qquad + \mu_1\lambda_{21}\tau^3 D_k c_k^{(2)} + \ldots, \\
&\ldots \\
r_k^{(j)} &= \tau c_k^{(1)} + \mu_j\tau^2 c_k^{(2)} + \tfrac{1}{2}\mu_j^2\tau^3(c_k^{(3)} - D_k c_k^{(2)}) + \\
&\qquad + \sum_{l=1}^{j-1} \mu_l\lambda_{jl}\tau^3 D_k c_k^{(2)} + \ldots, \\
&\ldots .
\end{aligned}\right. \tag{3.4}$$

In these expressions the vectors $c_k^{(j)}$ are the same as in formula (2.2).

Substituting the Taylor series for $r_k^{(j)}$ into formula (3.1) we find an expression of the form

$$(3.5)\qquad u_{k+1} = u_k + \beta_1\tau c_k^{(1)} + \beta_2\tau^2 c_k^{(2)} + \beta_3\tau^3 c_k^{(30)} + \frac{1}{2}\beta_{31}\tau^3 c_k^{(31)} + \\ + \beta_4\tau^4 c_k^{(40)} + \frac{1}{2}\beta_{41}\tau^4 c_k^{(41)} + \beta_{42}\tau^4 c_k^{(42)} + \frac{1}{6}\beta_{43}\tau^4 c_k^{(43)} + \\ + O(\tau^5).$$

Here, the vectors $c_k^{(j1)}$ can be expressed in the partial derivatives of the function $H(t,\tilde{U})$ (see [5]). For instance

$$c_k^{(30)} = D_k c_k^{(2)},$$

$$c_k^{(31)} = c_k^{(3)} - D_k c_k^{(2)}.$$

The parameters β_{j1} are defined as

$$(3.6)\qquad \begin{cases} \beta_1 = \sum_{j=0}^{n-1} \theta_j, \\ \beta_2 = \sum_{j=1}^{n-1} \theta_j\mu_j, \\ \beta_3 = \sum_{j=2}^{n-1} \theta_j \sum_{l=1}^{j-1} \lambda_{jl}\mu_l, \quad \beta_{31} = \sum_{j=1}^{n-1} \theta_j\mu_j^2, \\ \beta_4 = \sum_{j=3}^{n-1} \theta_j \sum_{l=2}^{j-1} \lambda_{jl} \sum_{i=1}^{l-1} \lambda_{li}\mu_i, \quad \beta_{41} = \sum_{j=2}^{n-1} \theta_j \sum_{l=1}^{j-1} \lambda_{jl}\mu_l^2, \\ \qquad\qquad \beta_{42} = \sum_{j=2}^{n-1} \theta_j\mu_j \sum_{l=1}^{j-1} \lambda_{jl}\mu_l, \\ \qquad\qquad \beta_{43} = \sum_{j=1}^{n-1} \theta_j\mu_j^3, \\ \qquad . \ . \ . \ . \end{cases}$$

On the other hand, we have for the local analytical solution $\tilde{U}'$, i.e. the integral curve through the point (t_k,u_k), the Taylor expansion (cf. [5]):

$$(3.7)\qquad \tilde{U}'_{k+1} = u_k + \tau c_k^{(1)} + \frac{1}{2}\tau^2 c_k^{(2)} + \frac{1}{6}\tau^3 c_k^{(30)} + \frac{1}{6}\tau^3 c_k^{(31)} + \frac{1}{24}\tau^4 c_k^{(40)} + \frac{1}{24}\tau^4 c_k^{(41)} + \frac{1}{8}\tau^4 c_k^{(42)} + \frac{1}{24}\tau^4 c_k^{(43)} + O(\tau^5).$$

By identifying the coefficients of corresponding terms the consistency conditions can be derived. We find

Table 3.1. Consistency conditions for p = 1,2,3,4.

p	β_1	β_2	β_3	β_{31}	β_4	β_{41}	β_{42}	β_{43}
1	1							
2	1	1/2						
3	1	1/2	1/6	1/3				
4	1	1/2	1/6	1/3	1/24	1/12	1/8	1/4

3.3. Stability conditions

In order to investigate the local stability properties of Runge-Kutta type schemes we apply the scheme to the locally linear representation (2.4) of the differential equation (compare section 2). It is easily verified that scheme (3.1) reduces to

$$u_{k+1} = u_k + \theta_0 r_k^{(0)} + \ldots + \theta_{n-1} r_k^{(n-1)},$$

$$r_k^{(0)} = \tau[D_k u_k + F_k(t_k)],$$

$$r_k^{(1)} = \tau[D_k u_k + \lambda_{10} D_k r_k^{(0)} + F_k(t_k + \mu_1 \tau)],$$

$$\ldots,$$

or more compactly

$$(3.6)\qquad u_{k+1} = P_n(\tau D_k) u_k + \tau \bar{g}_k^{(n)},$$

where $P_n(z)$ is the polynomial

$$(3.7) \qquad P_n(z) = 1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_n z^n.$$

The coefficients β_j are expressed in the Runge-Kutta parameters by (3.5). The vector $\bar{g}_k^{(n)}$ is determined by the vectors $F_k(t_k+\mu_j t)$. We observe that $\bar{g}_k^{(n)}$ is not identical to the inhomogeneous term $g_k^{(n)}$ which is obtained when the polynomial method described in [1] is applied to equation (2.4). It can be proved that they differ by a term $O(\tau^p)$, p being the order of accuracy of the scheme (compare [1], section 3).

Having established the polynomial which governs the local stability of the scheme we can set up the stability conditions by applying the linear theory as presented in [1] and [2]. For future reference we give in tabel 3.2 some important examples of stability polynomials which in the linear case generate a first, second, third and again a second order exact scheme, respectively.

Table 3.2. Coefficients of some stability polynomials

n	β_1	β_2	β_3	β_4	β_5	stability condition
4	1	5/32	1/128	1/8192		δ real, $\tau \leq 32/\sigma(D_k)$
4	1	1/2	.078	.0036		δ real, $\tau \leq 12/\sigma(D_k)$
4	1	1/2	1/6	.0185		δ real, $\tau \leq 6/\sigma(D_k)$
5	1	1/2	3/16	1/32	1/128	δ imaginary, $\tau \leq 4/\sigma(D_k)$.

Comparing this table of β_j values with table 3.1 we see that the order of accuracy in the non-linear case is the same as in the linear case.

3.4. Conditions for saving storage

Finally, conditions are given to limit the storage needed to accomplish the calculations. This is important when dealing with sets of equations arising from partial differential equations. In such cases the number of equations equals the number of grid points used to discretize the space derivatives. This number may be very large (1000 or more).

Consider the following computation scheme for u_{k+1}:

$$
(3.8)\quad \begin{cases}
\bar{r}_k^{(0)} = \tau H(t_k,u_k), & u_{k+1}^{(0)} = u_k + \bar{\theta}_0\bar{r}_k^{(0)},\\
\bar{r}_k^{(1)} = \tau H(t_k+\mu_1\tau,u_{k+1}^{(0)}+\lambda_1\bar{r}_k^{(0)}) + \nu_1\bar{r}_k^{(0)}, & u_{k+1}^{(1)} = u_{k+1}^{(0)} + \bar{\theta}_1\bar{r}_k^{(1)},\\
\bar{r}_k^{(2)} = \tau H(t_k+\mu_2\tau,u_{k+1}^{(1)}+\lambda_2\bar{r}_k^{(1)}) + \nu_2\bar{r}_k^{(1)}, & u_{k+1}^{(2)} = u_{k+1}^{(1)} + \bar{\theta}_2\bar{r}_k^{(2)},\\
\qquad\qquad\cdots\cdots\cdots & \\
\bar{r}_k^{(n-1)} = \tau H(t_k+\mu_{n-1}\tau,u_{k+1}^{(n-2)}+\lambda_{n-1}\bar{r}_k^{(n-2)}) + \nu_{n-1}\bar{r}_k^{(n-2)}, & u_{k+1} =\\
& = u_{k+1}^{(n-2)} + \bar{\theta}_{n-1}\bar{r}_k^{(n-1)}.
\end{cases}
$$

Here, $\bar{\theta}_j$, λ_j, ν_j and μ_j are free parameters.
Obviously, this scheme requires storage of only three and in cases where the coupling of the differential equations is weak two arrays. Also, when $\lambda_j = 0$, only two arrays are necessary.

We shall try to write our Runge-Kutta formulae in this form. For instance, for n = 3 scheme (3.8) is equivalent with scheme (3.1) generated by the matrix

$$
R = \begin{bmatrix}
\mu_1 & \bar{\theta}_0+\lambda_1 & 0\\
\mu_2 & \bar{\theta}_0+\bar{\theta}_1\nu_1+\lambda_2\nu_1 & \bar{\theta}_1+\lambda_2\\
\bar{\theta}_0+\bar{\theta}_1\nu_1+\bar{\theta}_2\nu_1\nu_2 & \bar{\theta}_1+\bar{\theta}_2\nu_2 & \bar{\theta}_2
\end{bmatrix}.
$$

A straighforward calculation yields the following expressions for the parameters $\bar{\theta}_j$, λ_j and ν_j in terms of the Runge-Kutta parameters θ_j, λ_{jl}:

$$
(3.9)\quad \begin{cases}
\nu_1 = \dfrac{\lambda_{20}-\theta_0}{\lambda_{21}-\theta_1}\\
\bar{\theta}_0 = \theta_0 - \theta_1\nu_1,\ \bar{\theta}_1 = \theta_1 - \theta_2\nu_2,\ \bar{\theta}_2 = \theta_2,\\
\lambda_1 = \lambda_{10} - \bar{\theta}_0, \lambda_2 = \lambda_{21} - \bar{\theta}_1.
\end{cases}
$$

The parameter ν_2 may be arbitrarily chosen.
From (3.9) we conclude that a 3-point Runge-Kutta type scheme can always be written in from (3.8) provided that

$$\lambda_{21} \neq \theta_1. \tag{3.10}$$

An important special case of scheme (3.8) arises when

$$\nu_j = 0,\ j = 1, \ldots, n-1. \tag{3.11}$$

In this case the analysis of consistency and stability simplifies considerably. The corresponding matrix R reduces to

$$R = \begin{bmatrix}
\mu_1 & \lambda_1+\bar{\theta}_0 & 0 & 0 & \cdots & 0 \\
\mu_2 & \bar{\theta}_0 & \lambda_2+\bar{\theta}_1 & 0 & \cdots & 0 \\
\mu_3 & \bar{\theta}_0 & \bar{\theta}_1 & \lambda_3+\bar{\theta}_2 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\mu_{n-2} & \bar{\theta}_0 & \bar{\theta}_1 & \cdots & \bar{\theta}_{n-4}\lambda_{n-2}+\bar{\theta}_{n-3} & \\
\mu_{n-1} & \bar{\theta}_0 & \bar{\theta}_1 & \cdots & \bar{\theta}_{n-4} & \bar{\theta}_{n-3}\ \ \lambda_{n-1}+\bar{\theta}_{n-2} \\
\bar{\theta}_0 & \bar{\theta}_1 & \bar{\theta}_2 & \cdots & \bar{\theta}_{n-3} & \bar{\theta}_{n-2}\ \ \bar{\theta}_{n-1}
\end{bmatrix} \tag{3.12}$$

Evidently, we have

$$\begin{cases} \bar{\theta}_j = \theta_j,\ j = 0, 1, \ldots, n-1, \\ \lambda_j = \lambda_{jj-1} - \theta_{j-1},\ j = 1, 2, \ldots, n-1. \end{cases} \tag{3.11'}$$

Furthermore, the Runge-Kutta parameters λ_{jl} have to satisfy the constraints

(3.12) $\lambda_{j1} = \theta_1, \ 0 \leq 1 \leq j-2.$

In many cases we may further simplify the scheme by putting

(3.13) $\theta_j = 0, \ j = 1, 2, \ldots, n-2.$

4. Solution of the consistency and stability equations

In this section we try to solve the equations which arise when the parameters β_j and β_{jl}, defined in formula (3.5), are given fixed values, for instance the values listed in tables 3.1 and 3.2.

4.1. First order exact schemes

Let us start with the class of first order exact schemes and let us try to find a solution of the consistency and stability equations under the constraints (3.12) and (3.13). First we give the expression for the parameters β_j when these storage saving conditions are introduced into (3.5). We find

$$
(4.1)\quad \begin{cases}
\beta_1 = \theta_0 + \theta_{n-1}, \\
\beta_2 = \theta_{n-1}\,(\theta_0+\lambda_{n-1n-2}), \\
\beta_3 = \theta_{n-1}\lambda_{n-1n-2}\,(\theta_0+\lambda_{n-2n-3}), \\
\quad \cdots \\
\beta_j = \theta_{n-1} \prod_{l=n-j+2}^{n-1} \lambda_{ll-1}\,(\theta_0+\lambda_{n-j+1n-j}), \\
\quad \cdots \\
\beta_n = \theta_{n-1} \prod_{l=1}^{n-1} \lambda_{ll-1}.
\end{cases}
$$

These equations are easily solved for θ_0 and λ_{jj-1}:

$$
(4.2)\qquad
\begin{cases}
\theta_0 = \beta_1 - \theta_{n-1}, \\
\lambda_{n-1n-2} = \dfrac{\beta_2}{\theta_{n-1}} - \theta_0, \\
\lambda_{n-2n-3} = \dfrac{\beta_3}{\theta_{n-1}\lambda_{n-1n-2}} - \theta_0, \\
\quad \cdots \\
\lambda_{jj-1} = \dfrac{\beta_{n-j+1}}{\theta_{n-1}\prod_{j+1}^{n-1}\lambda_{ll-1}} - \theta_0, \\
\quad \cdots \\
\lambda_{10} = \dfrac{\beta_n}{\theta_{n-1}\prod_{2}^{n-1}\lambda_{ll-1}}.
\end{cases}
$$

Next we consider the conditions for a first order, stabilized scheme. From the preceding section it follows that these conditions are

$$
(4.3)\qquad
\begin{cases}
\mu_1 = \lambda_{10}, \\
\mu_j = \theta_0 + \lambda_{jj-1}, \quad j = 2, \ldots, n-1, \\
\beta_1 = 1, \\
\beta_j \text{ has a prescribed value}, \quad j = 2, \ldots, n.
\end{cases}
$$

Obviously, these conditions can be satisfied and the Runge-Kutta parameters μ_j, λ_{jj-1} and θ_0 are completely determined by (4.2) and (4.3), provided that

$$
(4.4)\qquad \theta_{n-1} \neq 0, \ \lambda_{jj-1} \neq 0, \ j = 2, \ldots, n-1.
$$

Note that θ_{n-1} may assume every non-zero value. We may take advantage

of this free parameter to simplify the difference scheme. For instance, let us take $\theta_{n-1} = \beta_1$. Then the Runge-Kutta parameters can be expressed in terms of the parameters β_j alone and we arrive at the generating matrix

$$
(4.5) \qquad R = \begin{bmatrix}
\frac{\beta_n}{\beta_{n-1}} & \frac{\beta_n}{\beta_{n-1}} & 0 & \cdots & & 0 \\
\frac{\beta_{n-1}}{\beta_{n-2}} & 0 & \frac{\beta_{n-1}}{\beta_{n-2}} & \cdots & & 0 \\
\vdots & \vdots & \ddots & & & \vdots \\
\frac{\beta_3}{\beta_2} & 0 & \cdots & 0 & \frac{\beta_3}{\beta_2} & 0 \\
\frac{\beta_2}{\beta_1} & 0 & \cdots & 0 & 0 & \frac{\beta_2}{\beta_1} \\
0 & 0 & \cdots & 0 & 0 & \beta_1
\end{bmatrix}
$$

4.2. Second order exact schemes

When, in addition to (4.3), we impose on the difference scheme the condition (see table 3.1)

$$
(4.6) \qquad \beta_2 = \frac{1}{2} ,
$$

we obtain the class of second order, stabilized schemes. The considerations of the preceding subsection also apply to the second order case since we only require that the prescribed value of β_2 in (4.3) is just $\frac{1}{2}$. In particular, the generating matrix (4.5) can be used with $\beta_1 = 1$ $\beta_2 = \frac{1}{2}$.

4.3. Third order exact schemes

For third order accuracy we have to require (see table 3.1)

$$(4.7)\qquad \begin{cases} \beta_3 = \frac{1}{6}, \\ \beta_{31} = \frac{1}{3}. \end{cases}$$

We have from (3.5) and (4.3) that

$$(4.8)\qquad \beta_{31} = \theta_{n-1}\,(\theta_0+\lambda_{n-1n-2})^2.$$

Substituting the expression for θ_0 and λ_{n-1n-2} as given in (4.2) we find

$$(4.9)\qquad \theta_{n-1} = \frac{3}{4}.$$

Thus, the relations (4.2), (4.3) and (4.9) with $\beta_2 = \frac{1}{2}$ and $\beta_3 = \frac{1}{6}$ completely determine the parameters of the class of third order, stabilized Runge-Kutta type schemes.

4.4. Fourth order exact schemes

Since we have to satisfy three additional conditions in the case of fourth order accuracy (cf. table 2.1) we temporaly drop the storage saving conditions. Consider the standard fourth order Runge-Kutta formula which is characterized by the matrix (cf. [4])

$$(4.10)\qquad R = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 1 & 0 & 0 & 1 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{bmatrix}.$$

This formula suggests to consider n-point formulae characterized by matrices of the type

$$
(4.11)\qquad R = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & \cdots & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & \cdots & 0 \\
\mu_3 & \lambda_{30} & \lambda_{31} & \lambda_{32} & 0 & 0 & \cdots & 0 \\
\mu_4 & \lambda_{40} & \lambda_{41} & \lambda_{42} & \lambda_{43} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & & \vdots \\
\mu_{n-2} & \lambda_{n-20} & \lambda_{n-21} & \lambda_{n-22} & \cdots & & \lambda_{n-2n-3} & 0 \\
1 & 0 & 0 & 0 & \cdots & & 0 & 1 \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots & & 0 & \frac{1}{6}
\end{bmatrix}.
$$

A simple calculation leads to the following expression for the coefficients β_j and β_{j1} as defined by formula (3.5):

$$
(4.12)\qquad \begin{cases}
\beta_1 = 1, \\
\beta_2 = \frac{1}{2}, \\
\beta_3 = \frac{1}{12} + \frac{1}{6}\,\mu_{n-2}, \\
\beta_{31} = \frac{1}{3}, \\
\beta_4 = \frac{1}{6}\sum_{j=1}^{n-3} \lambda_{n-2j}\mu_j, \\
\beta_{42} = \frac{1}{24} + \frac{1}{6}\,\mu_{n-2}, \\
\beta_{41} = \frac{1}{24} + \frac{1}{6}\,\mu_{n-2}^2, \\
\beta_{43} = \frac{1}{4}.
\end{cases}
$$

The consistency conditions for fourth order accuracy are given by (3.3) and table 3.1. These conditions lead to the equations

$$\mu_j = \sum_{l=0}^{j-1} \lambda_{jl}, \quad j = 1, 2, \ldots, n-1,$$

$$\frac{1}{12} + \frac{1}{6}\mu_{n-2} = \frac{1}{6},$$

$$\frac{1}{6}\sum_{j=1}^{n-3} \lambda_{n-2j}\mu_j = \frac{1}{24},$$

$$\frac{1}{24} + \frac{1}{6}\mu_{n-2} = \frac{1}{8},$$

$$\frac{1}{24} + \frac{1}{6}\mu_{n-2}^2 = \frac{1}{12}.$$

From this it easily follows that the parameters λ_{jl} have to satisfy the relations

$$(4.13)\quad \begin{cases} \sum_{j=0}^{n-3} \lambda_{n-2j} = \frac{1}{2}, \\ \sum_{j=1}^{n-3} \lambda_{n-2j} \sum_{l=0}^{j-1} \lambda_{jl} = \frac{1}{4}. \end{cases}$$

To these equations we have to add the stability conditions, i.e.

(4.14) β_j has a prescribed value for $j = 5, 6, \ldots, n$.

First, we solve (4.13) and (4.14) for $n = 5$ and $n = 6$. Then, when we have obtained enough information about the structure of the equations to be solved, a general solution will be given.

Five-point formula

For $n = 5$ we find

$$(4.15)\quad \beta_5 = \frac{1}{24}\lambda_{32}.$$

Giving β_5 the value prescribed by the stability polynomial to be used, we obtain from (4.13) and (4.15) the equations

$$\begin{cases} \lambda_{30} + \lambda_{31} + \lambda_{32} = \frac{1}{2}, \\ (\lambda_{31}+\lambda_{32}) = \frac{1}{2}, \\ \lambda_{32} = 24\beta_5. \end{cases}$$

These equations are solved by

$$(4.16) \qquad \lambda_{30} = 0,\ \lambda_{31} = \frac{1}{2} - 24\beta_5,\ \lambda_{32} = 24\beta_5,$$

which leads to the generating matrix

$$(4.17) \qquad R = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} - 24\beta_5 & 24\beta_5 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{6} \end{bmatrix}.$$

Six-point formula

For n = 6 we have

$$(4.18) \begin{cases} \beta_5 = \frac{1}{24}\,[\lambda_{42}+2\lambda_{43}(\lambda_{31}+\lambda_{32})], \\ \beta_6 = \frac{1}{24}\ \lambda_{43}\lambda_{32}. \end{cases}$$

Consistency and stability conditions together lead to the following set of equations for the parameters λ_{j1}:

$$(4.19)\quad \begin{cases} \lambda_{40} + \lambda_{41} + \lambda_{42} + \lambda_{43} = \frac{1}{2}, \\ \lambda_{41} + \lambda_{42} + 2\lambda_{43}\,(\lambda_{30}+\lambda_{31}+\lambda_{32}) = \frac{1}{2}, \\ \lambda_{42} + 2\lambda_{43}\,(\lambda_{31}+\lambda_{32}) = 24\beta_5, \\ \lambda_{43}\lambda_{32} = 24\beta_6. \end{cases}$$

Putting

$$(4.20)\quad \lambda_{30} = \lambda_{40} = \lambda_{42} = 0$$

we arrive at the solution

$$(4.21)\quad \begin{cases} \lambda_{31} = \frac{1}{2} - \frac{\beta_6}{\beta_5}, \\ \lambda_{32} = \frac{\beta_6}{\beta_5}, \\ \lambda_{41} = \frac{1}{2} - 24\beta_5, \\ \lambda_{43} = 24\beta_5. \end{cases}$$

The generating matrix is given by

$$(4.22)\quad R = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} - \frac{\beta_6}{\beta_5} & \frac{\beta_6}{\beta_5} & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} - 24\beta_5 & 0 & 24\beta_5 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{6} \end{bmatrix}.$$

The structure of the generating matrices given by (4.17) and (4.22) suggests to try matrices of the type

$$
(4.23)\qquad R = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 & & & & \cdots & & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 & & & & \cdots & & 0 \\
\mu_3 & 0 & \lambda_{31} & \lambda_{32} & 0 & & & \cdots & & 0 \\
\mu_4 & 0 & \lambda_{41} & 0 & \lambda_{43} & 0 & & \cdots & & 0 \\
\mu_5 & 0 & 0 & 0 & 0 & \lambda_{54} & 0 & \cdots & & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & \vdots \\
\mu_{n-2} & 0 & 0 & 0 & \cdots & & & 0 & \lambda_{n-2n-3} & 0 \\
1 & 0 & 0 & 0 & \cdots & & & & 0 & 1 \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{3} & 0 & \cdots & & & & 0 & \frac{1}{6}
\end{bmatrix}.
$$

For this type of matrices we have the following expressions for the coefficients β_j:

$$
(4.24)\qquad \begin{cases}
\beta_n = \frac{1}{24} \prod_{j=3}^{n-2} \lambda_{jj-1}, & n \geq 7, \\
\beta_{n-1} = \frac{1}{12} \prod_{j=4}^{n-2} \lambda_{jj-1}\,(\lambda_{31}+\lambda_{32}), & n \geq 7, \\
\beta_{n-2} = \frac{1}{6} \prod_{j=5}^{n-2} \lambda_{jj-1}\,(\frac{1}{2}\lambda_{41}+\lambda_{43}(\lambda_{31}+\lambda_{32})), & n \geq 7, \\
\beta_{n-3} = \frac{1}{6} \prod_{j=5}^{n-2} \lambda_{jj-1}\,(\lambda_{41}+\lambda_{43}), & n \geq 8, \\
\beta_j = \frac{1}{6} \prod_{l=n-j+1}^{n-2} \lambda_{ll-1}, & j = 5, 6, \ldots, n-4,\ n \geq 9.
\end{cases}
$$

The consistency conditions (4.13) reduce to

$$(4.25)\quad \begin{cases} n = 7 \begin{cases} \lambda_{54} = \frac{1}{2} \\ \lambda_{54}\,(\lambda_{41}+\lambda_{43}) = \frac{1}{4} \end{cases}, \\ n \geq 8 \begin{cases} \lambda_{n-2n-3} = \frac{1}{2} \\ \lambda_{n-2n-3}\lambda_{n-3n-4} = \frac{1}{4} \end{cases}. \end{cases}$$

Giving the parameters β_j in (4.24) the values of the corresponding coefficients of the stability polynomial $P_n(z)$ the equations (4.24) and (4.25) become the relations which determine a stabilized, fourth order exact n-point Runge-Kutta scheme. In solving these equations we distinguish the cases $n = 7$ and $n \geq 8$.

Seven-point formula

A straightforward calculation yields

$$(4.26)\quad \begin{cases} \lambda_{31} = 48\,\dfrac{\beta_6-2\beta_7}{1-48(\beta_5-2\beta_6)}, \\ \lambda_{32} = 96\,\dfrac{\beta_7}{1-48(\beta_5-2\beta_6)}, \\ \lambda_{41} = 24\,(\beta_5-2\beta_6), \\ \lambda_{43} = \frac{1}{2}\,(1-48(\beta_5-2\beta_6)), \\ \lambda_{54} = \frac{1}{2}. \end{cases}$$

The parameters μ_j follow from (3.3).

n-point formula (n>8)

First, we define the number

$$(4.27) \qquad L_n = \prod_{j=5}^{n-2} \lambda_{jj-1}.$$

From (4.24) and (4.25) it is easily derived that

$$(4.27') \qquad L_n = \begin{cases} \frac{1}{4}, & n = 8 \\ 6\beta_{n-4}, & n \geq 9 \end{cases}.$$

Next we solve λ_{31}, λ_{32}, λ_{41} and λ_{43} from the first four equations of (4.2) to obtain

$$(4.28) \begin{cases} \lambda_{31} = 2 \dfrac{\beta_{n-1}-2\beta_n}{\beta_{n-3}-2\beta_{n-2}+4\beta_{n-1}}, \\ \lambda_{32} = 4 \dfrac{\beta_n}{\beta_{n-3}-2\beta_{n-2}+4\beta_{n-1}}, \\ \lambda_{41} = 12 \dfrac{\beta_{n-2}-2\beta_{n-1}}{L_n}, \\ \lambda_{43} = 6 \dfrac{\beta_{n-3}-2\beta_{n-2}+4\beta_{n-1}}{L_n}. \end{cases}$$

Furthermore, we have from (4.25)

$$(4.29) \begin{cases} \lambda_{n-2n-3} = \frac{1}{2}, \\ \lambda_{n-3n-4} = \frac{1}{2}. \end{cases}$$

When n = 8 formulae (4.28), (4.29) determine the generating matrix completely. When $n \geq 9$ we deduce from the last equation of (4.24) and (4.29)

$$(4.30)\quad \begin{cases} \lambda_{n-4n-5} = 24\beta_5, \ n \geq 9, \\ \lambda_{jj-1} = \dfrac{\beta_{j-1}}{\beta_j}, \ j = 5, 6, \ldots, n-5, \ n \geq 10. \end{cases}$$

5. An estimate for the local error

In reference [1] it was shown that in the linear case the discretization error ε_k, i.e. the difference between the analytical and difference solution satisfies difference scheme

$$\varepsilon_{k+1} = P_n(\tau D)\varepsilon_k + \rho_k, \tag{5.1}$$

where $P_n(z)$ is the generating polynomial of the scheme and ρ_k is the local error defined by

$$\rho_k = \tilde{U}_{k+1} - u'_{k+1}, \tag{5.2}$$

u'_{k+1} being the numerical result when the scheme is applied at the point $(t_k, \tilde{U}_k)$.

In the non-linear case we have by first linearizing the differential equation a similar relation for ε_k which, however, only holds approximately. Let $\rho_n(z)$ be the stability polynomial (3.7) then we have

$$\varepsilon_{k+1} \tilde{=} P_n(\tau D_k)\varepsilon_k + \rho_k. \tag{5.1'}$$

This approximation is better as τ and ε_k are smaller.

In order to control the total or global discretization error ε_k one should require that $||\rho_k||$ is less than same quantity η_k. In actual computation, however, we cannot compute ρ_k since $\tilde{U}_{k+1}$ is not known. Therefore, one controls the difference

$$\rho'_k = \tilde{U}'_{k+1} - u_{k+1}, \tag{5.3}$$

where $\tilde{U}'$ is the local analytical solution (cf. the discussion given in [2], section 2.2).

We now discuss the estimation of ρ'_k in terms of the vectors $r_k^{(j)}$, $j = 0, 1, \ldots, n-1$. From (3.5) and (3.7) we have

$$(5.3')\quad \rho_k' = (1-\beta_1)\tau c_k^{(1)} + (\tfrac{1}{2}-\beta_2)\tau^2 c_k^{(2)} + (\tfrac{1}{6}-\beta_3)\tau^3 c_k^{(30)} + \tfrac{1}{2}(\tfrac{1}{3}-\beta_{31})\tau^3 c_k^{(31)} +$$
$$+ (\tfrac{1}{24}-\beta_4)\tau^4 c_k^{(40)} + \tfrac{1}{2}(\tfrac{1}{12}-\beta_{41})\tau^4 c_k^{(41)} + (\tfrac{1}{8}-\beta_{42})\tau^4 c_k^{(42)} +$$
$$+ \tfrac{1}{6}(\tfrac{1}{4}-\beta_{43})\tau^4 c_k^{(43)} + O(\tau^5).$$

Furthermore, we consider a linear combination e_k of the first n' vectors $r_k^{(j)}$, i.e.

$$(5.4)\qquad e_k = \theta_0' r_k^{(0)} + \ldots + \theta_{n'-1}' r_k^{(n'-1)}.$$

By substituting Taylor expansions of the vectors $r_k^{(j)}$ into (5.4) we obtain (compare section 3.2)

$$(5.4')\quad e_k = \beta_1'\tau c_k^{(1)} + \beta_2'\tau^2 c_k^{(2)} + \beta_3'\tau^3 c_k^{(30)} + \tfrac{1}{2}\beta_{31}'\tau^3 c_k^{(31)} +$$
$$+ \beta_4'\tau^4 c_k^{(40)} + \tfrac{1}{2}\beta_{41}'\tau^4 c_k^{(41)} + \beta_{42}'\tau^4 c_k^{(42)} + \tfrac{1}{6}\beta_{43}'\tau^4 c_k^{(43)} + O(\tau^5),$$

where the coefficients β_j' and β_{j1}' are defined by formulae (3.6) when θ_j and n are replaced by θ_j' and n_1', respectively.
Comparison of (5.3') and (5.4') suggests to choose the parameters θ_j', j = 0, 1, ..., n' in such a way that the first terms in these series coincide. In table 5.1 several situations are listed.

Table 5.1. Conditions for approximation of ρ_k'

ρ_k'	β_1'	β_2'	β_3'	β_{31}'	β_4'	β_{41}'	β'_{42}	β_{43}'
$e_k+O(\tau^2)$	$1-\beta_1$							
$e_k+O(\tau^3)$	$1-\beta_1$	$\frac{1}{2}-\beta_2$						
$e_k+O(\tau^4)$	$1-\beta_1$	$\frac{1}{2}-\beta_2$	$\frac{1}{6}-\beta_3$	$\frac{1}{3}-\beta_{31}$				
$e_k+O(\tau^5)$	$1-\beta_1$	$\frac{1}{2}-\beta_2$	$\frac{1}{6}-\beta_3$	$\frac{1}{3}-\beta_{31}$	$\frac{1}{24}-\beta_4$	$\frac{1}{12}-\beta_{41}$	$\frac{1}{8}-\beta_{42}$	$\frac{1}{4}-\beta_{43}$

Suppose we have a Runge-Kutta formula which is second order exact, i.e. $\rho_k' = O(\tau^3)$. Then we wish a third order exact approximation of ρ_k', for instance the approximation $e_k + O(\tau^4)$. From table 5.1 it follows that four conditions have to be satisfied. Hence, at least four parameters θ_j' are necessary, i.e. $n' \geq 4$. We shall discuss the cases listed in table 5.1 in greater detail.

$\underline{\rho_k' = e_k + O(\tau^2)}$

Clearly, the conditions are satisfied by

$$(5.5) \qquad n' = 1,\ \theta_0' = 1 - \beta_1.$$

$\underline{\rho_k' = e_k + O(\tau^3)}$

From table 5.1 we have

$$\theta_0' + \theta_1' + \ldots + \theta_{n'-1}' = 1 - \beta_1,$$

$$\theta_1'\mu_1 + \ldots + \theta_{n'-1}'\mu_{n'-1} = \frac{1}{2} - \beta_2.$$

These equations are solved by

$$(5.6) \qquad n' = 2,\ \theta_0' = 1 - \beta_1 - \frac{\frac{1}{2} - \beta_2}{\lambda_{10}},\ \theta_1' = \frac{\frac{1}{2} - \beta_2}{\lambda_{10}}.$$

$\underline{\rho_k' = e_k + O(\tau^4)}$

We have four conditions to satisfy. Let us try $n' = 4$, i.e.

$$\theta_0' + \ldots + \theta_3' = 1 - \beta_1,$$

$$\theta_1'\mu_1 + \ldots + \theta_3'\mu_3 = \frac{1}{2} - \beta_2,$$

$$\theta_1'\mu_1^2 + \ldots + \theta_3'\mu_3^2 = \frac{1}{3} - \beta_{31},$$

$$\theta_2'\lambda_{21}\mu_1 + \theta_3'(\lambda_{31}\mu_1 + \lambda_{32}\mu_2) = \frac{1}{6} - \beta_3.$$

A straightforward calculation yields:

$$(5.7)\quad \begin{cases} \theta_3' = \dfrac{\mu_2(\mu_1-\mu_2)(\frac{1}{6}-\beta_3) - \mu_1\lambda_{21}[\mu_1(\frac{1}{2}-\beta_2) - (\frac{1}{3}-\beta_{31})]}{\mu_2(\mu_1-\mu_2)(\lambda_{31}\mu_1+\lambda_{32}\mu_2) - \mu_1\mu_3\lambda_{21}(\mu_1-\mu_3)}, \\ \theta_2' = \dfrac{\frac{1}{6} - \beta_3 - (\mu_1\lambda_{31}+\mu_2\lambda_{32})\theta_3'}{\mu_1\lambda_{21}}, \\ \theta_1' = \dfrac{\frac{1}{2} - \beta_2 - \mu_2\theta_2' - \mu_3\theta_3'}{\mu_1}, \\ \theta_0' = 1 - \beta_1 - (\theta_1'+\theta_2'+\theta_3'). \end{cases}$$

These expressions give the weights θ_j', provided that

$$(5.7')\qquad \mu_1 \neq 0,\ \lambda_{21} \neq 0,\ \mu_2(\mu_1-\mu_2)(\lambda_{31}\mu_1+\lambda_{32}\mu_2) \neq \mu_1\mu_3\lambda_{21}(\mu_1-\mu_3).$$

In this way we can find still higher order approximations of ρ_k'. A fourth order exact representation requires the solution of 8 linear equations (see table 5.1), a fifth order exact representation 17 equations, and so on. In an actual application it is most convenient to solve these equations numerically for the particular values of β_j used.

6. Numerical stability

As observed in [1], section 2.3, where linear differential equations were discussed, we have besides the error ρ_k a numerical error ρ_k^* which is due to round-off errors in each step of the integration. In order to control this error we require that the scheme is numerically stable, that is round-off errors shall not accumulate during the performance of one step. This feature is important when large values of n are used.

Consider a Runge-Kutta formula which can be written in form (3.8). For stability considerations we linearize the function $H(t,\tilde{U})$ to obtain formulae of the type

$$(6.1)\qquad \begin{pmatrix} \bar{r}_k^{(j+1)} \\ \bar{u}_k^{(j+1)} \end{pmatrix} = \begin{pmatrix} \lambda_{j+1}\tau D_k + \nu_{j+1} & \tau D_k \\ \bar{\theta}_{j+1}(\lambda_{j+1}\tau D_k + \nu_{j+1}) & 1 + \bar{\theta}_{j+1}\tau D_k \end{pmatrix} \begin{pmatrix} \bar{r}_k^{(j)} \\ \bar{u}_k^{(j)} \end{pmatrix},$$

where D_k is the Jacobian matrix of the set of differential equations.

For numerical stability we require that the eigenvalues $\bar{\delta}$ of the matrix in (6.1) are on or within the unit circle. Let δ be an eigenvalue of D_k then we have two eigenvalues $\bar{\delta}$ given by

$$(6.2)\qquad \bar{\delta}^2 - (1+\nu_{j+1}+(\lambda_{j+1}+\bar{\theta}_{j+1})\tau\delta)\bar{\delta} + (\nu_{j+1}+\lambda_{j+2}\tau\delta) = 0.$$

In this way all eigenvalues $\bar{\delta}$ van be calculated and conditions can be derived to force $\bar{\delta}$ within the unit circle.

7. Applications

We now are in a position to give explicitly difference schemes with prescribed accuracy and stability properties. The examples presented below are chosen from references [1] and [2].
We shall derive the generating matrix R, the stability conditions and the formula for the local error ρ_k'.

7.1. Equations with negative eigenvalues

In cases where the Jacobian D_k has negative eigenvalues of which the spectral radius $\sigma(D_k)$ is large, it is recommended to identify the stability polynomial $P_n(z)$ with the shifted Chebyshev polynomial $T_n(1+n^{-2}z)$ (compare [1], section 4.1). Here, we apply the theory of the preceding sections to the case $n = 4$.

The stability polynomial is given by

$$(7.1) \qquad T_4(1+\frac{z}{16}) = 1 + z + \frac{5}{32} z^2 + \frac{1}{128} z^3 + \frac{1}{8192} z^4.$$

This polynomial is compatible with the class of four-point Runge-Kutta schemes of first order and, therefore, a generating matrix of type (4.5) may be used. Substitution of $\beta_1 = 1$, $\beta_2 = 5/32$, $\beta_3 = 1/128$ and $\beta_4 = 1/8192$ leads to the matrix.

$$(7.2) \qquad R = \begin{bmatrix} \frac{1}{64} & \frac{1}{64} & 0 & 0 \\ \frac{1}{20} & 0 & \frac{1}{20} & 0 \\ \frac{5}{32} & 0 & 0 & \frac{5}{32} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The stability condition becomes (see [1], p. 20)

$$(7.3) \qquad \tau \leq \frac{32}{\sigma(D_k)}.$$

Since the scheme is first order exact a representation of ρ_k' is necessary which is at least second order exact. From formula (5.6) it follows that

$$\rho_k' \tilde{=} -22r_k^{(0)} + 22r_k^{(1)} + O(\tau^3). \tag{7.4}$$

A third order exact representation can be obtained by applying formula (5.7). We find

$$\rho_k' \tilde{=} \frac{1}{921}\left[355046r_k^{(0)} - 516918r_k^{(1)} + 159050r_k^{(2)} + 2822r_k^{(3)}\right] + O(\tau^4). \tag{7.5}$$

Finally, we consider the numerical stability of the difference scheme. The generating matrix satisfies the storage conditions (3.11) - (3.13), or more precisely

$$\nu_1 = \nu_2 = \nu_3 = 0,\ \bar{\theta}_0 = \bar{\theta}_1 = \bar{\theta}_2 = 0,\ \bar{\theta}_3 = 1,$$

$$\lambda_1 = \frac{1}{64},\ \lambda_2 = \frac{1}{20},\ \lambda_3 = \frac{5}{32}.$$

Substituting these values into (6.2) yields

$$\bar{\delta}^2 - (1+\lambda_{j+1}\tau\delta)\bar{\delta} + \lambda_{j+1}\tau\delta = 0,\ j = 0,\ 1,$$

$$\delta^{-2} - (1+\frac{37}{32}\tau\delta)^- + \frac{5}{32}\tau\delta = 0,\ j = 2.$$

The condition $|\bar{\delta}| \leq 1$ for $j = 0, 1, 2$ leads, respectively, to the inequalities:

$$\tau \leq \frac{64}{\sigma(D_k)},\ \tau \leq \frac{20}{\sigma(D_k)},\ \tau \leq \frac{32}{21\sigma(D_k)}. \tag{7.6}$$

Thus, when calculations are made with $\tau = 32/\sigma(D_k)$ as allowed by (7.3), we have an amplification of round-off errors in the calculation of $r_k^{(2)}$ and $r_k^{(3)}$. For this relatively low value of n this is not dangerous in an actual computation since only a few instable evaluations are made in succession. When a great number instable evaluations are made in succes-

sion the results will be seriously influenced by round-off errors (compare a similar situation in the theory of iterative processes for symmetric matrix equations [3]).

7.2. Equations with imaginary eigenvalues

In solving symmetric hyperbolic differential equations we are faced with systems of ordinary differential equations which have Jacobian matrices with purely imaginary eigenvalues. When a Runge-Kutta type difference scheme is used to solve this system of equations it is convenient to identify the stability polynomial $P_n(z)$ with the class of polynomials given in [1], section 5.1. We shall analyse the case n = 5. The stability polynomial is then given by (compare also table 3.2)

$$P_n(z) = 1 + z + \frac{1}{2}z^2 + \frac{3}{16}z^3 + \frac{1}{32}z^4 + \frac{1}{128}z^5. \tag{7.7}$$

This polynomial is compatible with a second order exact scheme, so that a generating matrix of type (4.5) with $\beta_1 = 1$, $\beta_2 = 1/2$ may be used, i.e.

$$R = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ \frac{1}{6} & 0 & \frac{1}{6} & 0 & 0 \\ \frac{3}{8} & 0 & 0 & \frac{3}{8} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \tag{7.8}$$

The stability condition becomes

$$\tau \leq \frac{4}{\sigma(D_k)} . \tag{7.9}$$

For stepsize control we need a third order exact representation of ρ_k'. Solving equations (5.7) we arrive at the error formula

$$(7.10)\qquad \rho_k' \simeq \frac{1}{78}\left[-739r_k^{(0)} + 966r_k^{(1)} - 603r_k^{(2)} + 376r_k^{(3)}\right] + O(\tau^4).$$

The numerical stability of this scheme is governed by the eigenvalue equations

$$\bar{\delta}^2 - (1+\lambda_j\delta)\bar{\delta} + \lambda_j\tau\delta = 0,\ \lambda_j = \frac{1}{4}, \frac{1}{6}, \frac{3}{8}$$

and

$$\bar{\delta}^2 - (1+\frac{3}{2}\delta)\bar{\delta} + \frac{1}{2}\tau\delta = 0.$$

The first equation leads to the sufficient conditions

$$(7.11)\qquad \tau \leq \frac{4}{\sigma(D_k)}\ ,\ \tau \leq \frac{6}{\sigma(D_k)}\ ,\ \tau \leq \frac{8}{3\sigma(D_k)}\ ,$$

the second equation to the necessary condition ($|\bar{\delta}_1\bar{\delta}_2| \leq 1$)

$$(7.11')\qquad \tau \leq \frac{2}{\sigma(D_k)}\ .$$

This means that, integrating with the maximal step $\tau = 4/\sigma(D_k)$ as prescribed by (7.9), numerical errors introduced in the last two function evaluations of a Runge-Kutta step are not damped out. Thus three stable evaluations are followed by two instable evaluations. In practice, however, this will not be dangerous, because every complete step is a stable one by virtue of condition (7.9).

References

[1] Houwen, P.J. van der, One-step methods for linear initial value problems I. Polynomial methods, TW report 119/70, Mathematisch Centrum,Amsterdam (1970).

[2] Houwen, P.J. van der, One-step methods for linear initial value problems II. Applications to stiff equations, TW report 122/70, Mathematisch Centrum, Amsterdam (1970).

[3] Houwen, P.J. van der, On the acceleration of Richardson's method II. Numerical aspects, TW report 107, Mathematisch Centrum, Amsterdam (1967).

[4] Runge, C., Uber die numerische Auflösung von Differentialgleichungen, Math. Ann., 46, p. 167-178 (1895).

[5] Zonneveld, J.A., Automatic numerical integration, MC tract 8, Mathematisch Centrum, Amsterdam (1964).