stichting
mathematisch
centrum

MC

AFDELING NUMERIEKE WISKUNDE NW 125/82 MAART
(DEPARTMENT OF NUMERICAL MATHEMATICS)

J.G. VERWER, S. SCHOLZ, J.G. BLOM & M. LOUTER-NOOL

A CLASS OF RUNGE-KUTTA-ROSENBROCK METHODS FOR SOLVING STIFF DIFFERENTIAL EQUATIONS

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, 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.).

1980 Mathematical subject classification: 65L05

ACM Computer review categories: 5.17

A class of Runge-Kutta-Rosenbrock methods for solving stiff differential equations [*)]

by

J.G. Verwer, S. Scholz[**)], J.G. Blom & M. Louter-Nool

ABSTRACT

A class of Runge-Kutta-Rosenbrock methods is discussed for the numerical solution of the initial value problem for stiff systems of ordinary differential equations. We pay special attention to the use of time-lagged Jacobian matrices. The aim is to obtain an appreciable reduction in the number of Jacobian matrix evaluations and related matrix factorizations. When dealing with large systems the costs of these computations normally form a large proportion of the total integration costs of the Rosenbrock method.

KEY WORDS & PHRASES: *Numerical analysis, Stiff systems of ordinary differential equations, Runge-Kutta-Rosenbrock methods*

[*)] This report will be submitted for publication elsewhere.

[**)] Technische Universität, Sektion Mathematik, WB Num. Math., Mommsenstr. 13, 8027 Dresden DDR

1. INTRODUCTION

This paper deals with the numerical solution of the initial value problem for stiff systems of ordinary differential equations (ODEs)

$$(1.1) \qquad \dot{y} = f(y),\ x \geq x_0,\ y(x_0) = y_0,$$

by means of a Runge-Kutta-Rosenbrock (RKR) method [13]. The numerical solutions we consider are defined by the s-stage RKR formula

$$(1.2a) \qquad \begin{aligned} E_{n+\eta}k_i &= f(y_n + h\sum_{j=1}^{i-1} a_{ij}k_j) + \sum_{j=1}^{i-1} c_{ij}k_j, \qquad i = 1,\dots,s, \\ y_{n+1} &= y_n + h\sum_{i=1}^{s} w_i k_i, \qquad n \geq 0, \end{aligned}$$

where the matrix $E_{n+\eta}$ is given by

$$(1.2b) \qquad E_{n+\eta} = I - \gamma h f_y(y_{n+\eta}).$$

Here a_{ij}, c_{ij}, w_i, γ are real parameters, whereas η is assumed to be a non-positive integer. By setting $\eta = 0$, we obtain a scheme which has been thoroughly studied in several papers, e.g. [7,9,11,18]. Many of the results presented in these papers carry over to schemes where η may also be negative. We are interested in such schemes for the following reason. If $\eta = 0$, method (1.2) requires an f_y-evaluation at every integration step. Because this evaluation (normally) implies an LU-decomposition, the costs involved may form a considerable proportion of the total costs of one step, especially when dealing with large systems. Now suppose that η will be chosen negative and in such a way that for some number of steps $y_n \rightarrow y_{n+1}$ the sum $n + \eta$ is constant. During these steps the matrix $E_{n+\eta}$ is then also constant provided the stepsize $h = x_{n+1}-x_n$ does not change. In this way we thus can escape the necessity of performing an f_y-evaluation and LU-decomposition at every step. It should be demanded, however, that the order of consistency of the schemes is independent of η (cf. [15,17], see also [16]). This demand highly complicates the treatment of the order conditions.

In short, the contents of the paper are as follows. Section 2 deals with some preliminaries on the definition of the right-hand side function f and the increment vectors k_i. The order conditions are discussed in section 3, while section 4 contains some known results on absolute stability. Absolute stability is associated to linear problems with constant coefficients for which the choice of η plays no role of course. In section 5 we propose an extrapolation scheme (we call it a modified Richardson extrapolation) which exploits the possibility of using a time-lagged Jacobian matrix without a decrease of the order of consistency. An example of such an extrapolation scheme, which is A-stable and of order 4, is given in section 6. In section 7 this scheme is provided with stepsize and local error control, in the usual way, and then compared with the codes GEAR (see [5]) and ROW4A. ROW4A is an A-stable RKR code developed by KAPS & RENTROP [8] and modified by GOTTWALD & WANNER [5]. ROW4A is based on a fairly accurate scheme of order 4 and especially equipped for problems showing sudden changes from a smooth to a non-smooth behaviour (see [5]). The final section, section 8, is devoted to some conclusions.

2. SOME PRELIMINARY REMARKS

It may be clarifying to remark that method (1.2) can also be applied to non-autonomous ODE systems. Consider the linearly implicit system

$$(2.1)\qquad M\dot{y} = f(x,y), \quad x \geq x_0, \quad y(x_0) = y_0,$$

where M is a constant matrix. For this problem (1.2) can be reformulated to (see also [9,14])

$$(2.2)\qquad \begin{aligned} E_{n+\eta} &= M - \gamma h f_y(x_{n+\eta}, y_{n+\eta}) \\ E_{n+\eta} k_i &= f(x_n + A_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j) + B_i h f_x(x_{n+\eta}, y_{n+\eta}) + \\ &\qquad M \sum_{j=1}^{i-1} c_{ij} k_j, \qquad i = 1,\ldots,s, \\ y_{n+1} &= y_n + h \sum_{i=1}^{s} w_i k_i, \end{aligned}$$

where

$$B_i = \gamma + \sum_{j=1}^{i-1} c_{ij}B_j, \qquad A_i = \sum_{j=1}^{i-1} a_{ij}B_j/\gamma.$$

Note that the partial derivative vector f_x now enters into the computation. The matrix M has been added just for the sake of completeness. Normally, M = I. In the following we confine ourselves to this case. Examples of stiff problems where M ≠ I arise, e.g., in mechanics and in the numerical solution of time-dependent partial differential equations (finite element space discretization).

It is also of interest to observe that within the class of RKR methods using the present E-matrix, formulation (1.2) is very general. Consider the increment vector definition

$$E_{n+\eta}g_i = h\sum_{j=1}^{i} \beta_{ij}f(y_n + \sum_{m=1}^{j-1} \alpha_{jm}g_m) + h\sum_{j=1}^{i-1} (\gamma_{ij}f_y(y_{n+\eta}) + \xi_{ij}I)g_j$$

for an increment vector g_i. When compared with the corresponding formula in (1.2a), we added all earlier computed f-values from the current step plus the products $f_y(y_{n+\eta})g_j$. It is not difficult to show however, by simple transformations, that all RKR formulas based on g_i can be written in the more simple form (1.2).

The following formulation has been used in [7,9,11,18]:

$$\begin{aligned} & u_i = y_n + \sum_{j=1}^{i-1} \alpha_{ij}g_j, \\ (2.3) \qquad & E_{n+\eta}g_i = hf(u_i) + hf_y(y_{n+\eta})\sum_{j=1}^{i-1} \gamma_{ij}g_j, \qquad i = 1,\ldots,s, \\ & y_{n+1} = y_n + \sum_{i=1}^{s} \mu_i g_i. \end{aligned}$$

All theoretical results for $\eta = 0$ in these papers have actually been derived for scheme (2.3). We therefore adopt this formulation in our following sections. Actual integrations will always be carried out with scheme (1.2), however. This scheme avoids the matrix vector multiplications occurring in (2.3). The relations between the coefficients of (2.3) and (1.2) are given by (cf. [9])

$$(2.4)\qquad \begin{aligned} c_{ij} &= \sum_{k=1}^{i-1} \gamma_{ik}(\delta_{kj}-c_{kj})/\gamma, \quad a_{ij} = \sum_{k=1}^{i-1} \alpha_{ik}(\delta_{kj}-c_{kj}), \\ w_j &= \sum_{k=1}^{s} \mu_k(\delta_{kj}-c_{kj}), \ \delta_{kj} \text{ the Kronecker symbol.} \end{aligned}$$

In scheme (2.3) the exact Jacobian matrix is used. With respect to the order of consistency p it is allowed to replace $f_y(y_{n+\eta})$ by a difference approximation $\Delta f_y(y_{n+\eta})$, provided, elementwise,

$$(2.5)\qquad \Delta f_y(y_{n+\eta}) = f_y(y_{n+\eta}) + O(h^{p-1}).$$

3. THE ORDER CONDITIONS

In this section we consider scheme (2.3) written in the form

$$(3.1.a)\qquad E_{n+\eta}g_i = f(y_n+h\sum_{j=1}^{i-1} \alpha_{ij}g_j) + hf_y(y_{n+\eta})\sum_{j=1}^{i-1}\gamma_{ij}g_j, \quad i = 1,\dots,s,$$

$$(3.1.b)\qquad y_{n+1} = y_n + h\sum_{i=1}^{s} \mu_i g_i.$$

The conditions for the coefficients of a one step method of order p for the solution of (1.1) can be obtained by equating the coefficients of the expansion of the approximate solution y_{n+1} to the solution y(x) that satisfies the demand $y(x_n) = y_n$:

$$(3.2)\qquad y(x_n+h) = y_n + \sum_{r=1}^{\infty} \frac{h^r}{r!} \sum_{\mathrm{ord}F=r} \kappa(F)[F]_n, \quad \text{where } [F]_n \text{ denotes } F(y_n).$$

The coefficients $\kappa(F)$ of the elementary differential F are recursively defined as (see BUTCHER [3]):

(3.3)

if order F is 1: $\kappa(F) = 1$, and

if order F is r and $F = \{F_1^{\nu_1}\dots F_\sigma^{\nu_\sigma}\}$: $\kappa(F) = (r-1)!\displaystyle\prod_{i=1}^{\sigma}\frac{1}{\nu_i!}\left(\frac{\kappa(F_i)}{r_i!}\right)^{\nu_i}$

where r_i is the order of F_i.

The technique to obtain an analogous expansion of y_{n+1} from (3.1), i.e.

(3.4) $$y_{n+1} = y_n + \sum_{r=1}^{\infty} \frac{h^r}{r!} \sum_{\mathrm{ord}F=r} \phi(F)[F]_n ,$$

is in essence the same as in the case $\eta = 0$ (see [7,9,11,18]). To illustrate the changes caused by the use of a time lagged Jacobian matrix we give a lemma and a theorem which lead to the desired expansion (3.4). Details of the derivation can be found in the institute report [1].

If h is small enough, g_i can be expanded in a power series

(3.5) $$g_i = \sum_{r=0}^{\infty} \frac{h^r}{r!} K_{i,r}.$$

If we replace g_i and g_j in equation (3.1.a) by the corresponding power series (3.5) and expand all terms of the equation we get a recurrence relation for $K_{i,r}$ for $r > 0$. By means of this recurrence relation we can prove by mathematical induction on r that $K_{i,r-1}$ can be expressed as a linear combination of the elementary differentials of order r. This result is stated in the following lemma:

<u>LEMMA 3.1</u>. *Let* $f = f(y)$ *be analytic and let* $y_{n+\eta}$ *have an expansion of the form*

$$y_{n+\eta} = y_n + \sum_{r=1}^{\infty} \frac{h^r}{r!} z_r, \qquad \textit{where } z_r = \eta^r [D^{r-1} f]_n ,$$

and denote $\gamma_{ii} = \gamma$. *Then* $K_{i,r-1}$ *can be written as*

(3.6) $$K_{i,r-1} = \sum_{\mathrm{ord}F=r} \psi_i(F)[F]_n ,$$

where $\psi_i(F)$ *is a polynomial in* α_{ij}, γ_{ij} *and* η *that satisfies the recurrence relation*

(3.7)

if order F is 1: $\psi_i(f) = 1$,

if order F is r and $F = \{F_1^{\nu_1} \ldots F_\sigma^{\nu_\sigma}\}$:

$$\psi_i(F) = (r-1)! \prod_{m=1}^{\sigma} \frac{1}{\nu_m!} \frac{1}{r_m!} \nu_m \{ \prod_{m=1}^{\sigma} (\sum_{j=1}^{i-1} \alpha_{ij} r_m \psi_j(F_m))^{\nu_m} +$$

$$\prod_{m=1}^{\sigma} (\eta^{r_m} \kappa(F_m))^{\nu_m} \sum_{q=1}^{\sigma} \frac{\nu_q r_q}{\eta^{r_q} \kappa(F_q)} \sum_{j=1}^{i} \gamma_{ij} \psi_j(F_q) \},$$

where r_i *is the order of* F_i. □

This leads to

THEOREM 3.2. *If the conditions of lemma 3.1 are fulfilled, the next statement holds: to each elementary differential* F *corresponds an elementary weight* $\phi = \phi(F)$ *that is defined by*

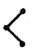
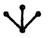
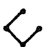
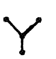
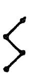
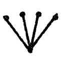
$$(3.8) \qquad \phi(F) = \sum_{i=1}^{s} \mu_i \, r \, \psi_i(F),$$

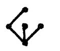
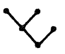
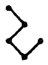
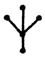
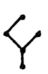
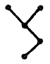
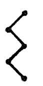
where r *is the order of* F, $\psi_i(F)$ *is given by* (3.7) *and* $\kappa(F)$ *by* (3.3). □

The order conditions $\phi(F) = \kappa(F)$ have been derived from the recurrence relations using a formula manipulation program (see [1]). The conditions up to order 5 have been listed in Table 1 in the same style as in KAPS & WANNER [9], Table 2. In Table 1 the following abbreviations have been used:

$$\beta_{ij} = \alpha_{ij} + \gamma_{ij}, \ \alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \ \beta_i = \sum_{j=1}^{i-1} \beta_{ij}, \ \alpha_{ij} = \gamma_{ij} = 0 \text{ for } i \le j.$$

TABLE 1. Order conditions for p ≤ 5.

1		$\Sigma\mu_i = p_1, \ p_1 = 1$
2		$\Sigma\mu_i\beta_i = p_2, \ p_2 = \frac{1}{2} - \gamma$
3		$\Sigma\mu_i\alpha_i^2 + \eta(1-2\Sigma\mu_i\alpha_i) = p_3, \ p_3 = \frac{1}{3}$
4		$\Sigma\mu_i\beta_{ij}\beta_j = p_4, \ p_4 = \frac{1}{6} - \gamma + \gamma^2$
5		$\Sigma\mu_i\alpha_i^3 + \eta^2(\frac{3}{2} - 3\Sigma\mu_i\alpha_i) = p_5, \ p_5 = \frac{1}{4}$
6		$\Sigma\mu_i\alpha_i\alpha_{ij}\beta_j + \eta(\frac{1}{6} + \gamma(\Sigma\mu_i\alpha_i - 1) - \Sigma\mu_i\alpha_{ij}\beta_j) +$ $\eta^2(\frac{1}{4} - \frac{1}{2}\Sigma\mu_i\alpha_i) = p_6, \ p_6 = \frac{1}{8} - \frac{\gamma}{3}$
7		$\Sigma\mu_i\beta_{ij}\alpha_j^2 + \eta(\frac{1}{3} - \gamma - 2\Sigma\mu_i\beta_{ij}\alpha_j) = p_7, \ p_7 = \frac{1}{12} - \frac{\gamma}{3}$
8		$\Sigma\mu_i\beta_{ij}\beta_{jk}\beta_k = p_8, \ p_8 = \frac{1}{24} - \frac{\gamma}{2} + \frac{3\gamma^2}{2} - \gamma^3$
9		$\Sigma\mu_i\alpha_i^4 + \eta^3(2 - 4\Sigma\mu_i\alpha_i) = p_9, \ p_9 = \frac{1}{5}$

10		$\Sigma\mu_i\alpha_i^2\alpha_{ij}\beta_j + \eta^2(\frac{1}{6}-\frac{3}{2}\gamma + 2\gamma\Sigma\mu_i\alpha_i - \Sigma\mu_i\alpha_{ij}\beta_j) +$ $\eta^3(\frac{1}{2}-\Sigma\mu_i\alpha_i) = p_{10},\ p_{10} = \frac{1}{10}-\frac{\gamma}{4}$
11		$\Sigma\mu_i\alpha_{ij}\beta_j\alpha_{ik}\beta_k + \eta(-\frac{\gamma}{3}+\gamma^2 + 2\gamma\Sigma\mu_i\alpha_{ij}\beta_j) +$ $\eta^2(\frac{1}{6}-\frac{1}{2}\gamma - \Sigma\mu_i\alpha_{ij}\beta_j) = p_{11},\ p_{11} = \frac{1}{20}-\frac{\gamma}{4}+\frac{\gamma^2}{3}$
12		$\Sigma\mu_i\alpha_i\alpha_{ij}\alpha_j^2 + \eta(\frac{1}{3}-2\Sigma\mu_i\alpha_i\alpha_{ij}\alpha_j - \Sigma\mu_i\alpha_{ij}\alpha_j^2) +$ $\eta^2(-\frac{1}{3}+2\Sigma\mu_i\alpha_{ij}\alpha_j) + \eta^3(-\frac{1}{3}+\frac{2}{3}\Sigma\mu_i\alpha_i) = p_{12},\ p_{12} = \frac{1}{15}$
13		$\Sigma\mu_i\alpha_i\alpha_{ij}\beta_{jk}\beta_k + \eta(\frac{1}{24}-\frac{1}{3}\gamma + \gamma^2 - \gamma^2\Sigma\mu_i\alpha_i - \Sigma\mu_i\alpha_{ij}\beta_{jk}\beta_k) +$ $\eta^2(-\frac{\gamma}{2}+\gamma\Sigma\mu_i\alpha_i) + \eta^3(\frac{1}{12}-\frac{1}{6}\Sigma\mu_i\alpha_i) = p_{13},\ p_{13} = \frac{1}{30}-\frac{\gamma}{4}+\frac{\gamma^2}{3}$
14		$\Sigma\mu_i\beta_{ij}\alpha_j^3 + \eta^2(\frac{1}{2}-\frac{3\gamma}{2}-3\Sigma\mu_i\beta_{ij}\alpha_j) = p_{14},\ p_{14} = \frac{1}{20}-\frac{\gamma}{4}$
15		$\Sigma\mu_i\beta_{ij}\alpha_j\alpha_{jk}\beta_k + \eta(\frac{1}{24}-\frac{1}{2}\gamma + \gamma^2 + \gamma\Sigma\mu_i\beta_{ij}\alpha_j - \Sigma\mu_i\beta_{ij}\alpha_{jk}\beta_k) +$ $\eta^2(\frac{1}{12}-\frac{1}{4}\gamma - \frac{1}{2}\Sigma\mu_i\beta_{ij}\alpha_j) = p_{15},\ p_{15} = \frac{1}{40}-\frac{5\gamma}{24}+\frac{\gamma^2}{3}$
16		$\Sigma\mu_i\beta_{ij}\beta_{jk}\alpha_k^2 + \eta(\frac{1}{12}-\frac{2\gamma}{3}+\gamma^2 - 2\Sigma\mu_i\beta_{ij}\beta_{jk}\alpha_k) = p_{16},\ p_{16} = \frac{1}{60}-\frac{\gamma}{6}+\frac{\gamma^2}{3}$
17		$\Sigma\mu_i\beta_{ij}\beta_{jk}\beta_{k\ell}\beta_\ell = p_{17},\ p_{17} = \frac{1}{120} -\frac{\gamma}{6}+\gamma^2 - 2\gamma^3 + \gamma^4$

4. THE ABSOLUTE STABILITY

When applied to the test equation for absolute stability, i.e.

$$y' = \lambda y, \qquad \lambda \in \mathbb{C},$$

scheme (2.3) yields

$$y_{n+1} = R(z)y_n,\ z = h\lambda,$$

where

$$(4.1) \qquad R(z) = 1 + \sum_{\ell=1}^{s} (\frac{z}{1-\gamma z})^{\ell}\ \vec{\mu}^T B^{\ell-1}\vec{e}.$$

B is the lower triangular matrix $(\alpha_{ij}+\gamma_{ij})$, $\vec{\mu}^T = [\mu_1,\ldots,\mu_s]$ and $\vec{e}^T = [1,\ldots,1]$. The rational function R is the well-known stability function which determines the region of absolute stability and which, for $z \to 0$, approximates the exponential. Let q denote the order of R, i.e. q is the largest integer such that $R(z) = e^z + O(z^{q+1})$, $z \to 0$. If it is required that $q \geq s$, R is known [2,10,11], viz.

$$(4.2) \qquad R(z) = \left(\sum_{j=0}^{s} z^j \sum_{i=0}^{j} \binom{s}{i}\frac{(-\gamma)^i}{(j-i)!}\right)/(1-\gamma z)^s.$$

Hence, if $q \geq s$, the absolute stability is completely determined by the parameter γ. In what follows we therefore shall use the notation $R(\gamma,z)$ for (4.2). The requirement of L-stability, i.e. A-stability and $R(\gamma,\infty) = 0$, fixes γ. BURRAGE [2] has given ranges for γ for which $R(\gamma,z)$ is A-stable. For $s = 4$, for example, this range is approximately given by [0.39434, 1.28057].

5. A MODIFIED RICHARDSON EXTRAPOLATION

During the investigation several approaches have been tried in order to find the best possible use of integrating with a constant E-matrix and an order p which is independent of η. As a rule, the extrapolation process described in this section turned out to be the most fruitful one. In this process the parameter η alternates between 0 and -1.

Again we consider formulation (2.3). It is convenient to associate with (2.3) the map

$$(5.1) \qquad v \to S[\gamma,\eta;h,v],$$

v being the vector to which the RKR scheme is applied ($v = y_n$ in (2.3)). Now let δ be a positive real and let v_n be an approximation to the exact solution $y = y(x)$ of (1.1) at the point $x = x_n$. For some given h, we then compute v_{n+2}, an approximation at $x = x_{n+2} = x_n + (1+\delta)h$, by the "modified Richardson extrapolation"

$$(5.2a) \qquad v_{n+1} = S[\gamma,0;h,v_n]$$

(5.2b) $$v_{n+2}^{(1)} = S[\gamma\delta^{-1},-1;\delta h,v_{n+1}],$$

(5.2c) $$v_{n+2}^{(2)} = S[\gamma(1+\delta)^{-1},0;(1+\delta)h,v_n],$$

(5.2d) $$v_{n+2} = v_{n+2}^{(1)} + \alpha(v_{n+2}^{(1)} - v_{n+2}^{(2)}).$$

The above formulation may require some additional explanation. Suppose we have constructed a p-th order RKR scheme S in such a way that γ and $\eta \in \{0,-1\}$ are still free. The computations (5.2a) - (5.2c) are then all of the same order p and, as can be readily seen, use only one E-matrix which is given by $I - \gamma hf_y(v_n)$. Note that the original Richardson process for our RKR scheme requires the treatment of three different E-matrices (two f_y-evaluations and three LU-decompositions). The parameter δ has been introduced in order to have the possibility to perform the second step $v_{n+1} \rightarrow v_{n+2}^{(1)}$ with a somewhat smaller stepsize than the first step $v_n \rightarrow v_{n+1}$. This is desirable due to the fact that the error constants of method (2.3) will grow as η will decrease. This can be deduced from the expressions for the order conditions.

Because we actually apply three different integration schemes in (5.2), the parameter α occurring in the extrapolation formula (5.2d) cannot be used to enlarge the order of v_{n+2} to p+1. In the next section we will use α as well as γ and δ, for defining an A-stable, 4-th order extrapolation scheme $v_n \rightarrow v_{n+2}$ with reasonably small principal error constants. The underlying RKR scheme S is especially adapted to (5.2) in the sense that the "double step" $v_n \rightarrow v_{n+2}^{(2)}$ is relatively cheap. Note that the difference $v_{n+2}^{(1)} - v_{n+2}^{(2)}$, which is of order p+1 in h, can be used for stepsize control.

Before proceeding with the next section we still give the stability function of the extrapolation scheme $v_n \rightarrow v_{n+2}$, as well as the expression for its error constants in terms of the error constants of the RKR scheme S.

Suppose that the scheme S is such that its stability function R is given by $R(\gamma,z)$ defined by (4.2). The stability function of the complete scheme (5.2), say $\tilde{R}$, is then given by

(5.3) $$\tilde{R}(z) = (1+\alpha)R(\gamma,z)R(\gamma\delta^{-1},\delta z) - \alpha R(\gamma(1+\delta)^{-1},(1+\delta)z).$$

Hence the absolute stability region of the complete scheme (5.2) is determined by the parameters γ, α and δ.

Suppose that the RKR scheme S referred to in (5.1) is of order p and that p does not depend on γ and η. For a sufficiently smooth solution $y = y(x)$ of the ODE system (1.1) we then can write

$$(5.4) \qquad y(x_n+h) - S[\gamma,\eta;h,y(x_n)] = h^{p+1}\sum_i C_i(\gamma,\eta)F_i(y(x_n)) + O(h^{p+2}),$$

where F_i is used as a simple notation for the i-th elementary differential of order p. Hence $C_i = C_i(\gamma,\eta)$ represents the i-th principal local error constant.

When associating the map

$$(5.5) \qquad v \to \tilde{S}[\alpha,\gamma,\delta;h,v]$$

with the extrapolation scheme (5.2) we can write, after some easy calculations,

$$(5.6) \qquad y(x_n+(1+\delta)h) - \tilde{S}[\alpha,\gamma,\delta;h,y(x_n)] = h^{p+1}\sum_i \tilde{C}_i(\alpha,\gamma,\delta)F_i(y(x_n)) + O(h^{p+2}),$$

where

$$(5.7) \qquad \tilde{C}_i(\alpha,\gamma,\delta) = (1+\alpha)[C_i(\gamma,0) + \delta^5 C_i(\gamma\delta^{-1},-1)] - \alpha(1+\delta)^5 C_i(\gamma(1+\delta)^{-1},0).$$

Observe that we expand in h while (5.2) steps from x_n to $x_n + (1+\delta)h$.

6. A 4-TH ORDER, A-STABLE EXTRAPOLATION SCHEME

The parameter values for this scheme are contained in Table 2. We omit the derivation of these parameters and confine ourselves to the following remarks:

(i) We have tabulated the parameters of the corresponding RKR formula (1.2). This formula should be implemented on the computer. Readers who are interested in the belonging parameter values of formula (2.3) should use relations (2.4).

(ii) The number of stages s of the RKR formula is equal to 4, but only two f-evaluations are required. The formula can be implemented in such a way that the third computation (5.2c) can be performed at the cost of one f(y)-evaluation and two forward-backward substitutions. This means that the step $v_n \to v_{n+2}$ costs one f_y-evaluation, one LU-decomposition, five f-evaluations and ten FB-substitutions. By way of comparison, the costs of two steps with the 4-th order RKR code ROW4A amount to two f_y-evaluations, two LU-decompositions, six f-evaluations and eight FB-substitutions.

(iii) The stability function $\tilde{R}(z)$ is A-stable. We verified this numerically. Further, $\tilde{R}(-\infty) \simeq 0.4$. Hence stiff solution components are sufficiently damped by our extrapolation scheme.

(iv) The parameter $\delta = 0.6$. We need a value reasonably smaller than one in order to cope with the error constants which we meet for $\eta = -1$. Some of these turn out to be significantly larger (a factor 10) than the corresponding error constants for $\eta = 0$. All nine error constants $C_i(\alpha,\gamma,\delta)$, see (5.7), have been listed in Table 3.

TABLE 2. Integration parameters of three different 4-stage RKR formulas (1.2) of order p = 4. For $\alpha = 0.1$, $\gamma = 0.4$ and $\delta = 0.6$ these three formulas can be combined to a 4-th order, A-stable extrapolation method (5.2). The parameters have been rounded to 11 decimal digits.

	(5.2a)	(5.2b)	(5.2c)
a_{21}	0	0	0
a_{31}	0.84375	1.35666117081	0
a_{32}	-0.046875	-0.33289385680	0
a_{41}	0.84375	1.35666117081	0
a_{42}	-0.046875	-0.33289385680	0.375
a_{43}	0	0	0
c_{21}	1	1	1
c_{31}	0	0	0
c_{32}	-1.125	-0.19780410790	1
c_{41}	0.92045454545	-0.03182829164	1.125
c_{42}	-0.92045454545	0.03182829164	-0.5625

c_{43}	0.81818181818	-0.16090814282	-0.5625
w_1	-0.45370370370	3.34089914352	-0.37037037037
w_2	1.27777777778	-1.89325651260	0.22222222222
w_3	1.08641975309	-1.26969525484	0.44444444444
w_4	-0.27160493827	2.36792462950	0.59259259259

TABLE 3. Absolute values of all nine error constants $\tilde{C}_i(\alpha,\gamma,\delta)$ of the 4-th order extrapolation method given in Table 2. The integer numbers correspond with the numbers of the order conditions from Table 1.

9	10	11	12	13	14	15	16	17
0.0002	0.0013	0.0007	0.0330	0.0006	0.0009	0.0029	0.0099	0.0073

7. NUMERICAL EXAMPLES

We have implemented the 4-th order extrapolation scheme from Table 2 in a research code, called RKRMC, and have compared this code with the aforementioned RKR code ROW4A. The results of the comparison have been collected in this section. For the sake of completeness we also have included results of the backward differentiation code GEAR [6]. Backward differentiation codes belong to the most popular ones in the field of stiff equations.

Our stepsize and local error control is to a great extent the same as in ROW4A. For example, we have adopted the idea of back-stepping. Our criterion for a back-step is different, however. Details can be found in the flow chart given in Figure 4. Please recall that the back-step strategy involves that the integration is always continued till two steps beyond x_{end} has been successful [5]. Facmax, facmin and fac represent the usual threshold parameters, while EST_{n+2} and h_{new} are defined by

$$EST_{n+2} = \alpha \max_i |v^{(1)}_{n+2,i} - v^{(2)}_{n+2,i}| / \max(1, |v_{n,i}|, |v_{n+2,i}|),$$

$$h_{new} = h * \min(facmax, \max(facmin, fac(TOL/EST_{n+2})^{1/5})).$$

Here, v_i denotes the i-th component of the vector variable v.

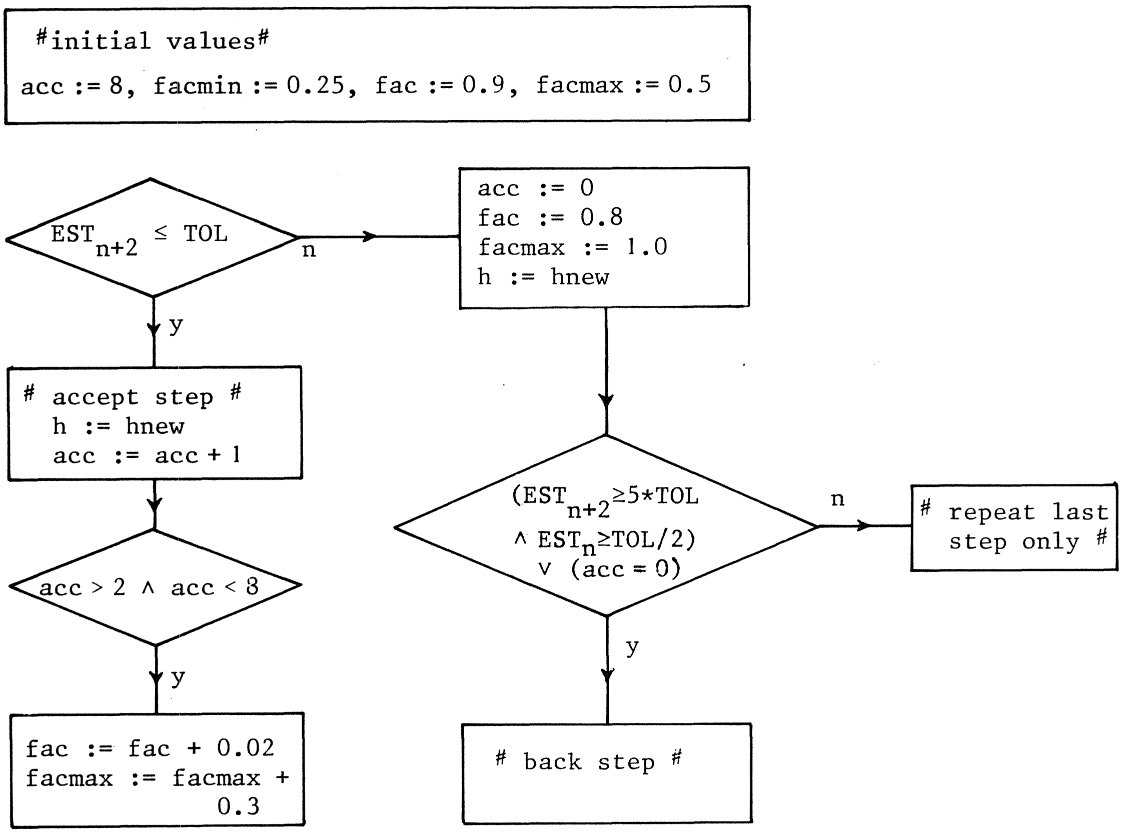


Fig. 4. Flow-chart describing the control strategy.

The experiments were carried out in a similar way as in the paper of GOTTWALD & WANNER [5]. We have compared the three codes for the 25 stiff test problems from ENRIGHT, HULL & LINDBERG [4] plus the 5 additional problems $F_1,\ldots,F_5$ suggested by Gottwald and Wanner. There are harder due to sudden changes from a smooth to a non-smooth behaviour. In order to detect these sudden changes in time, the idea of back-stepping has been suggested.

In all integrations of $F_1,\ldots,F_5$ the initial step size h was equal to 10^{-3}. For the other problems we always used the values given in [4]. Further, in all experiments the analytic partial derivative matrix f_y has been used. Concerning the use of GEAR, two further remarks must be made. We called GEAR with its method parameter INDEX = 2, i.e. the end point is always hit exactly and no output interpolation is made. Further, we changed the weights YMAX(I) which are used in the mixed relative-absolute local error test. In our calls, YMAX(I) = max(1,|current Y(I)-value|). After this change the overall picture of GEAR's performance slightly improved.

For three TOL values, namely 10^{-2}, 10^{-3} and 10^{-4}, we have plotted (see Fig. 5,6,7):

a) The maximal global error of the computed solution over the whole integration interval and all components in the mixed relative-absolute sense. To this purpose we called ROW4A over each current step interval with TOL = 10^{-8} and used the corresponding computed solution as an exact reference solution.

b) The number of f-evaluations.

c) The number of FB-substitutions. Note that for ROW4A and RKRMC this number is larger than the number of f-evaluations.

d) The number of LU-decompositions. For all three codes this number is equal to the number of f_y-evaluations, provided f has a non-constant Jacobian.

To a large extent, the numbers of f-evaluations, FB-substitutions, LU-decompositions and f_y-evaluations determine the amount of computing work involved.

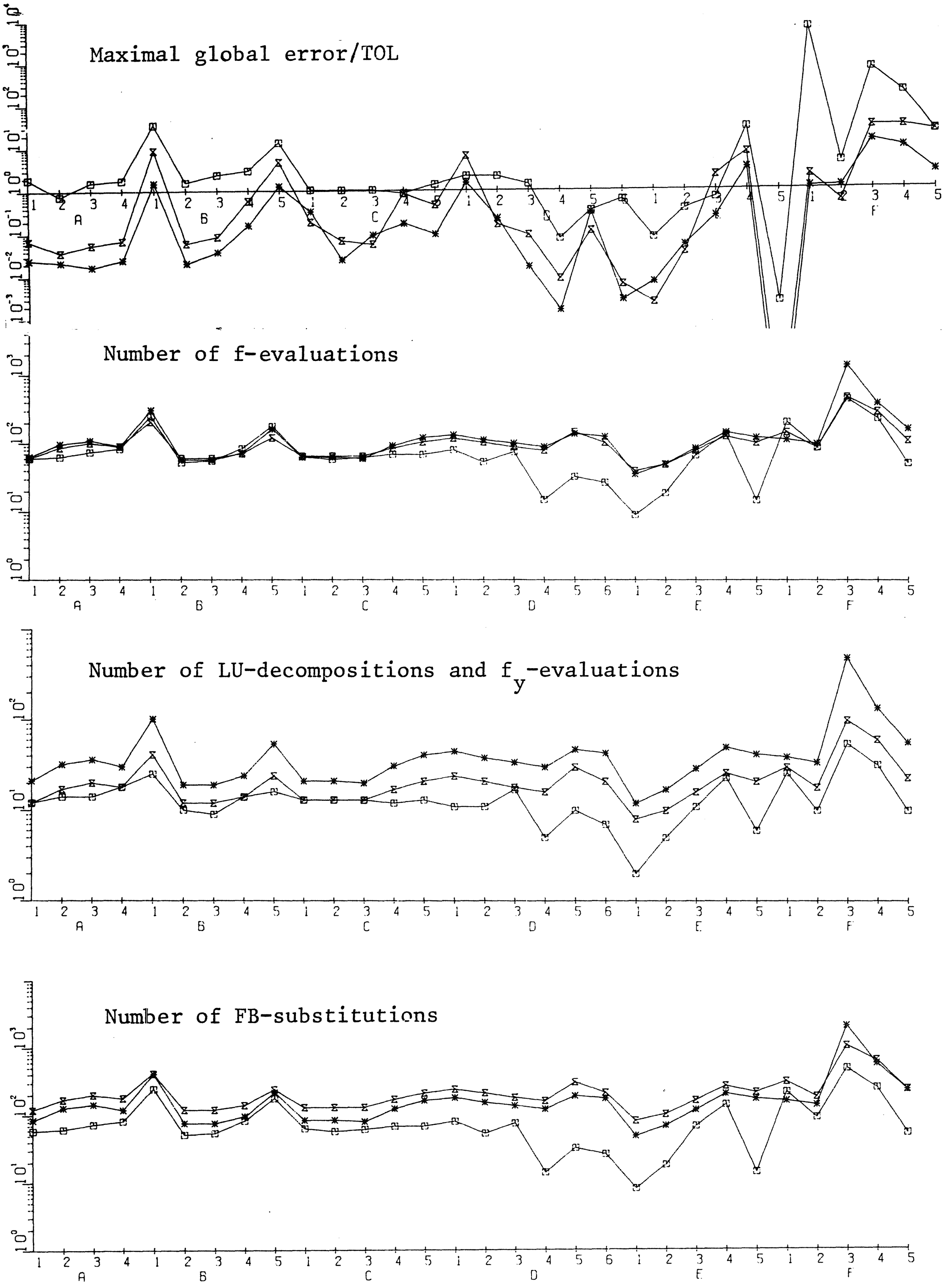


Fig. 5. Results for TOL = 10^{-2}; □ GEAR * ROW4A ⧗ RKRMC

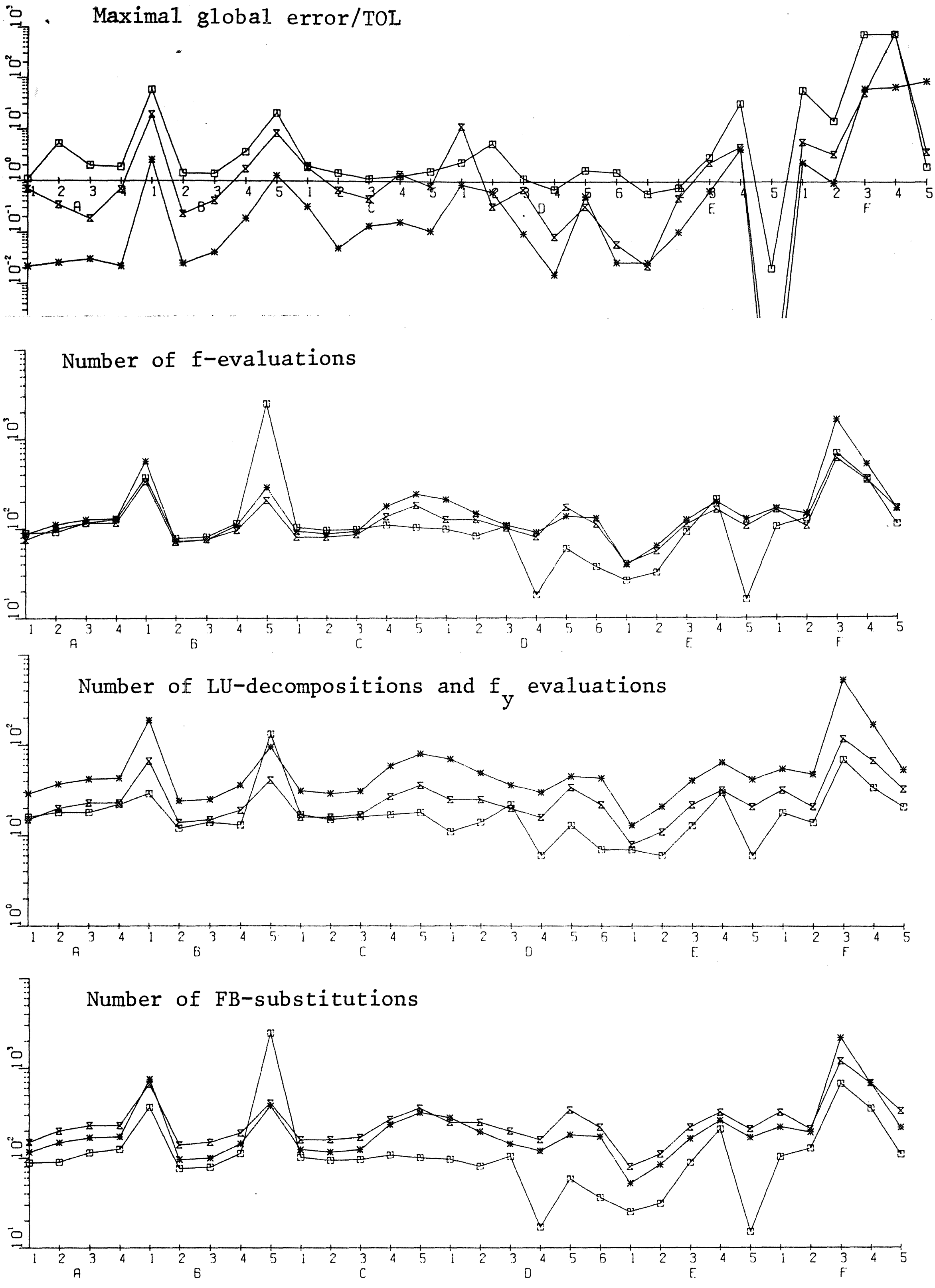


Fig. 6. Results for TOL = 10^{-3}; □ GEAR, *ROW4A ⧖ RKRMC

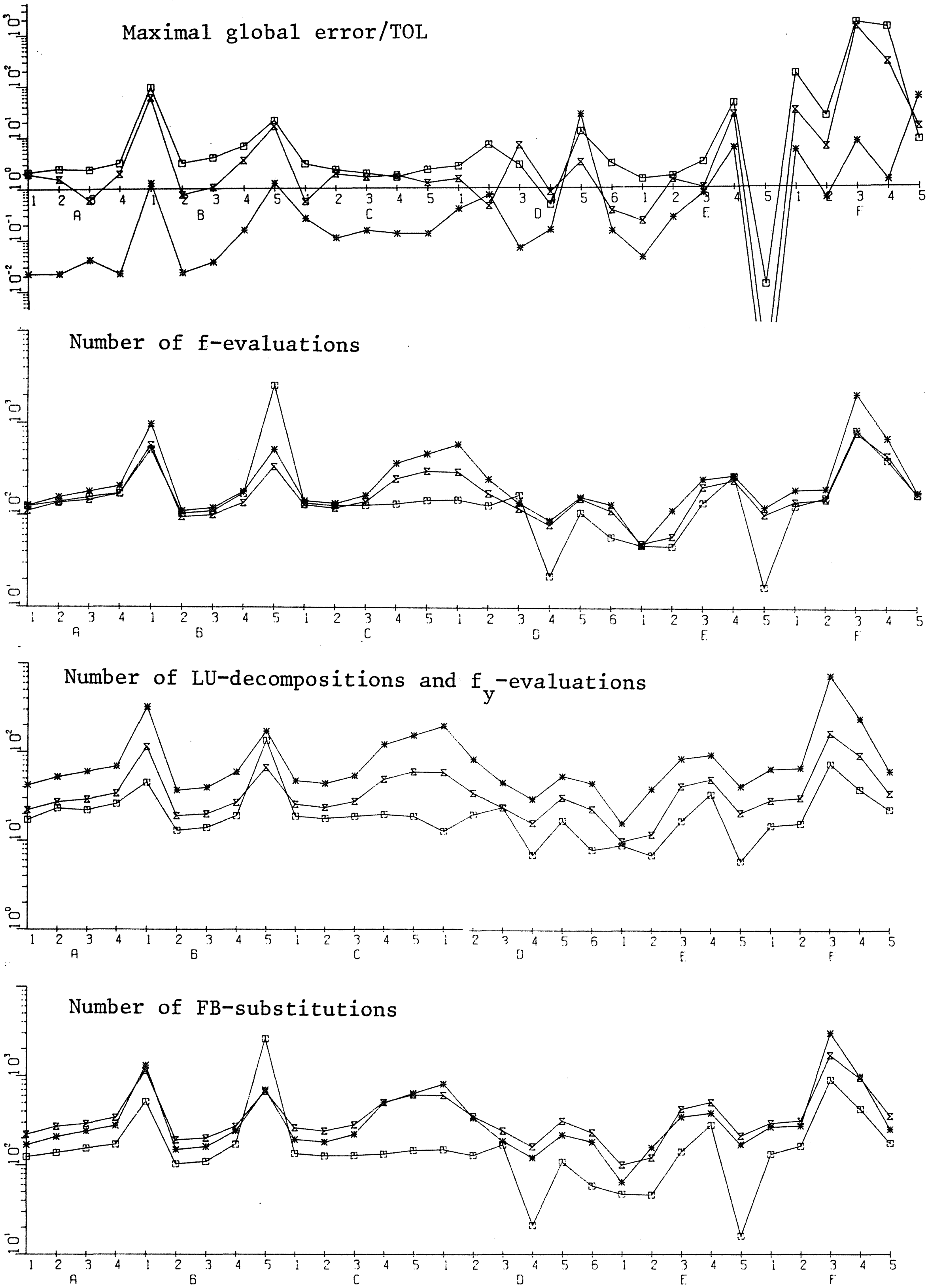


Fig. 7. Results for TOL = 10^{-4}; □ GEAR * ROW4A ⧖ RKRMC

8. CONCLUSIONS

The comparison of RKRMC with ROW4A shows that the use of the extrapolation scheme normally shall lead to an appreciable reduction in the required number of LU-decompositions and f_y-evaluations. On the present test set RKRMC computes approximately half of the number of LU-decompositions and f_y-evaluations used by ROW4A, while the number of f-evaluations plus FB-substitutions is almost equal. With respect to accuracy behaviour however, RKRMC normally falls behind ROW4A. For TOL = 10^{-2} the maximal global errors of RKRMC and ROW4A are more or less comparable, but for TOL = 10^{-3}, 10^{-4} ROW4A is, in most cases, clearly the more accurate code. It is difficult to say whether the use of a time-lagged Jacobian matrix should be blamed for this. The choice of the underlying RKR scheme also plays a role of course (compare the results for the linear problems from classes A and B).

As already observed by GOTTWALD & WANNER [5], with respect to accuracy behaviour GEAR is considerably less reliable than ROW4A. On the other hand, in almost all cases GEAR also requires considerably less computational work than ROW4A. In both respects, RKRMC lies between GEAR and ROW4A.

REFERENCES

[1] BLOM, J.G., *Order conditions for a class of Runge-Kutta-Rosenbrock methods*. Report NN25/82, Mathematical Centre, Amsterdam, 1982.

[2] BURRAGE, K., *A special family of Runge-Kutta methods for solving stiff differential equations*. BIT 18, 22-41 (1978).

[3] BUTCHER, J.C., *Coefficients for the study of Runge-Kutta integration processes*. J. Austral. Math. Soc. 3, 185-201 (1963).

[4] ENRIGHT, W., HULL, T.E., LINDBERG, B., *Comparing numerical methods for stiff systems of ODEs*. BIT 15, 10-48 (1975).

[5] GOTTWALD, B.A., G. WANNER, *A reliable Rosenbrock integrator for stiff differential equations*. Computing 27, 355-360 (1981).

[6] HINDMARSH, A.C., *GEAR*, *Ordinary differential equation system solver*, Lawrence Livermore Laboratory Report UCID-30001, Rev. 3, 1974.

[7] KAPS, P., *Modifizierte Rosenbrockmethoden der Ordnung 4,5 und 6 zur numerischen Integration steifer Differentialgleichungen.* Dissertation Universität Innsbruck (1977).

[8] KAPS, P., RENTROP, P., *Generalized Runge-Kutta methods of order 4 with stepsize control for stiff ordinary differential equations.* Numer. Math. 33, 55-68 (1979).

[9] KAPS, P., WANNER, G., *A study of Rosenbrock-type methods of high order.* Numer. Math. 38, 279-298 (1981).

[10] NORSETT, S.P., WANNER, G., *The real-pole sandwich for rational approximations and oscillation equations.* BIT 19, 79-94 (1979).

[11] NORSETT, S.P., WOLFBRANDT, A., *Attainable order of rational approximations to the exponential function with only real poles.* BIT 17, 200-208 (1977).

[12] NORSETT, S.P., WOLFBRANDT, A., *Order conditions for Rosenbrock type methods.* Numer. Math. 32, 1-15 (1979).

[13] ROSENBROCK, H.H., *Some general implicit processes for the numerical solution of differential equations.* Computer J. 18, 329-330 (1963).

[14] SHAMPINE, L.F., *Implementation of Rosenbrock methods.* Report SAND80 - 2367J, Sandia National Laboratories, Albuquerque, New Mexico (1980).

[15] SCHOLZ, S., *Modifizierte Rosenbrock-Verfahren mit genäherter Jacobi-Matrix,* Report Sektion Mathematik, Technische Universität Dresden (1979).

[16] STEIHAUG, T., WOLFBRANDT, A., *An attempt to avoid exact Jacobian and nonlinear equations in the numerical solution of stiff differential equations.* Math. Comp. 33, 521-534 (1979).

[17] VERWER, J.G., SCHOLZ, S., *Rosenbrock methods and time-lagged Jacobian matrices.* Beiträge zur Numerischen Mathematik 11, 1982.

[18] WOLFBRANDT, A., *A study of Rosenbrock processes with respect to order conditions and stiff stability.* Thesis Chalmers Univ. of Technology, Göteborg (1977).