

**stichting
mathematisch
centrum**



AFDELING NUMERIEKE WISKUNDE
(DEPARTMENT OF NUMERICAL MATHEMATICS)

NW 91/80

SEPTEMBER

B.P. SOMMEIJER & J.G. VERWER

A PERFORMANCE EVALUATION OF A CLASS OF
RUNGE-KUTTA-CHEBYSHEV METHODS FOR SOLVING
SEMI-DISCRETE PARABOLIC DIFFERENTIAL EQUATIONS

kruislaan 413 1098 SJ amsterdam

STICHTING WISKUNDE EN WISKUNDE
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 Mathematics Subject Classification: 65L05, 65M20

ACM Computing Review-Categories: 5.17

A performance evaluation of a class of Runge-Kutta-Chebyshev methods
for solving semi-discrete parabolic differential equations

by

B.P. Sommeijer & J.G. Verwer

ABSTRACT

The subject of the paper is the numerical integration of semi-discrete parabolic partial differential equations by means of Runge-Kutta-Chebyshev integration methods. These are explicit methods and can thus be easily applied to multi-space dimensional problems. On the other hand, the methods are conditionally stable. The real stability boundaries are proportional to m^2 , m being the number of function evaluations per time step. Thanks to a property called internal stability, it is possible to integrate with an arbitrarily large value of m . Herewith the disadvantage of conditional stability can be significantly reduced. The main purpose of the paper is a numerical performance evaluation of a set of 6 closely related methods.

KEY WORDS & PHRASES: *Numerical analysis, Parabolic equations, Method of lines, Stabilized explicit methods*

1. INTRODUCTION

The subject of this paper is the time-integration of parabolic initial-boundary value problems by means of Runge-Kutta-Chebyshev integration methods [2,7]. For that purpose we assume that the initial-boundary value problem has already been converted into an initial value problem for a system of ordinary differential equations by means of space-discretization. We write this system in the explicit, autonomous form

$$(1.1) \quad \frac{dy}{dt} = f(y).$$

For our discussion it is not necessary to define a particular class of parabolic problems or to specify the space-discretization technique. Our only restriction is that the eigenvalues of the Jacobian matrix

$$(1.2) \quad J(y) = \partial f(y) / \partial y$$

are situated in a long narrow strip along the negative axis of the complex plane. A lot of semi-discrete parabolic equations satisfy this restriction [5]. Further it is always assumed that the usually nonlinear vector function f is sufficiently often differentiable. The autonomous form is only for notational convenience.

Runge-Kutta-Chebyshev (RKC) methods may be characterized as follows. The methods are explicit and, consequently, can be easily applied to large problem classes. Indeed, their advantage, when compared with implicit or partly implicit methods (see e.g. [5]), is that they do not require the solution of large and complicated systems of nonlinear algebraic or transcendental equations (more dimensional problems). On the other hand, the methods are conditionally stable. To be more precise, the real absolute stability boundary, say β , of an RKC-method, is always of the form

$$(1.3) \quad \beta = c(m)m^2,$$

where c is a nearly constant function of m , m being the number of $f(y)$ -evaluations per time step. The essential property of an RKC-method is now

that, without the danger of instabilities within one time step [2,7], this number can be made arbitrarily large. Because $\beta \sim m^2$ as $m \rightarrow \infty$, it thus is possible to reduce the disadvantage of conditional stability significantly. The internal stability, i.e. stability per time step, is obtained by applying stable Chebyshev recursions. This explains the name of the method.

This paper reports a numerical performance evaluation of 6 related methods: (i) The first order 1-step method developed in [2] and a modification of their second order method (ii) A 2-step method of order 1 and a 2-step method of order 2 (iii) The 3-step methods of order 1 and 2 given in [7]. The modification of the second order 1-step method serves to make the scheme more accurate for non-linear problems. The 2-step methods may be considered as being obtained by slightly modifying the 3-step methods.

Our aim of the numerical comparison is to select one (or possibly more) RKC-formula which at a later time should form the basis for an automatic explicit solver for a wide class of semi-discrete parabolic equations. Such a program has already been published in [8]. Its underlying integration formula however is not internally stable and can therefore be used only for relatively small values of m .

2. RUNGE-KUTTA-CHEBYSHEV METHODS

In this section we briefly describe the methods we tested. Details on the construction and analysis of the methods are, as far as possible, omitted. The interested reader is referred to [2,7]. A common property of the methods is that their absolute stability regions (see e.g. [1,4]) all contain a long and narrow strip along the negative axis of the complex plane. Such a type of absolute stability region is attractive when dealing with semi-discrete parabolic equations. The length of the strip, i.e. the real absolute stability boundary β , depends on the number of stages m within the Runge-Kutta method. In all methods m is greater than or equal to 2. Below we tabulated the approximate expressions for β .

order	1-step	2-step	3-step
1	$1.94m^2$	$3.58m^2$	$5.17m^2$
2	$0.65(m^2-1)$	$1.19m^2$	$2.32m^2$

Because of the fact that the 2-step methods were derived from the 3-step ones it is convenient to begin with the latter methods. In all formulas y_n denotes the numerical approximation at time $t = t_n$ and $\tau = t_{n+1} - t_n$ denotes the stepsize.

2.1. The 3-step methods

The 3-step methods we tested, both of order 1 and order 2, are defined by the scheme [7]

$$\begin{aligned}
 y_{n+1}^{(0)} &= \mu_0 y_n + (1-\mu_0) y_{n-1}, \\
 y_{n+1}^{(1)} &= \mu_1 y_n + (1-\mu_1) y_{n-1} + \tau(\tilde{\gamma}_1 f(y_n) + \tilde{\delta}_1 f(y_{n-1})), \\
 (2.1) \quad y_{n+1}^{(j)} &= \mu_j y_{n+1}^{(j-1)} + (1-\mu_j) y_{n+1}^{(j-2)} + \tau \tilde{\mu}_j f(y_{n+1}^{(j-1)}), \quad j = 2(1)m, \\
 y_{n+1} &= \alpha(\alpha_0 y_{n+1}^{(m)} + \alpha_1 y_n + \alpha_2 y_{n-1}) + (1-\alpha) y_{n-2}, \quad m \geq 2.
 \end{aligned}$$

The internal stability property mentioned before is a property of the recurrence equation defining $y_{n+1}^{(j)}$, $j \geq 2$. The parameters μ_j and $\tilde{\mu}_j$ are chosen in such a way that for linear homogeneous problems $y' = Jy$ a stable 3-term Chebyshev recursion appears. The integration parameters are given by

$$\begin{aligned}
 (2.2) \quad \mu_0 &= \mu_1 = (a+b(1-p_0))/2a, \quad \tilde{\gamma}_1 = w_1 \mu_0 / w_0, \quad \tilde{\delta}_1 = w_1 (1-\mu_0) / w_0, \\
 \mu_j &= 2w_0 T_{j-1}(w_0) / T_j(w_0), \quad \tilde{\mu}_j = 2w_1 T_{j-1}(w_0) / T_j(w_0), \quad j = 2(1)m, \\
 \alpha &= 2/(2-p_0), \quad \alpha_0 = 2a, \quad \alpha_1 = (1-b)(1-p_0) - a, \quad \alpha_2 = 1 - 2a - \alpha_1,
 \end{aligned}$$

where

$$(2.3) \quad w_0 = 1 + 0.05/m^2, \quad w_1 = (\frac{1}{2} - \frac{1}{4}p_0) T_m(w_0) / a T'_m(w_0).$$

Here $T_j(x) = \cos[j \arccos x]$. The parameters a , b and p_0 determine the order of consistency of the scheme. For the first order formulas we have

$$(2.4) \quad a = 0.975, \quad b = 0.2, \quad p_0 = 124/229.$$

Order 2 is obtained if a , b and p_0 satisfy

$$(2.5) \quad a = 0.81, \quad b = 0.6,$$

$$\left(\frac{b}{a} + \frac{\xi}{4}\right)p_0^2 - \left(\frac{3b}{a} + \xi\right)p_0 + \xi + \frac{2b}{a} - 4 = 0,$$

where $\xi = T_m(w_0)T_m''(w_0)/aT_m'^2(w_0)$. For large values of m the parameter $p_0 \approx -0.66$.

2.2. The 2-step methods

By putting

$$(2.6) \quad \alpha = 1$$

equation (2.1) represents a 2-step method. Though this 2-step method was not discussed in [7], we decided to insert it for the sake of completeness. The derivation of integration parameters leading to absolute stability regions of the desired shape can be performed in the same way as in the 3-step case. The integration parameters are again defined by (2.2) and (2.3), where $p_0 = 0$ and the expression for w_0 is replaced by $w_0 = 1 + \varepsilon/m^2$. For the first order formulas we have

$$(2.7) \quad a = 0.925, \quad b = 0.45, \quad \varepsilon = 0.05.$$

For the second order formulas a , b and ε are given by (see (2.5))

$$(2.8) \quad a = \frac{1}{2}b + \frac{T_m(w_0)T_m''(w_0)}{4T_m'^2(w_0)}, \quad b = 0.50, \quad \varepsilon = 0.1.$$

Without proof we state that this choice leads to the stability boundaries given above.

2.3. The 1-step methods

The 1-step methods read [2,6]

$$\begin{aligned}
(2.9) \quad & y_{n+1}^{(0)} = y_n, \\
& y_{n+1}^{(1)} = y_n + \tilde{\mu}_1 \tau f(y_n), \\
& y_{n+1}^{(j)} = \mu_j y_{n+1}^{(j-1)} + \nu_j y_{n+1}^{(j-2)} + (1-\mu_j-\nu_j)y_n + \tilde{\mu}_j \tau f(y_{n+1}^{(j-1)}) + \\
& \quad + \tilde{\gamma}_j \tau f(y_n), \quad j = 2, \dots, m, \\
& y_{n+1}^{(m)} = y_{n+1}.
\end{aligned}$$

In comparison with the formulas described in [2], we left the first order scheme unchanged, whereas we slightly modified the second order one with the aim to make it more accurate for non-linear problems. More precisely, we now use internal stability polynomials of the form

$$(2.10) \quad R_j(z) = a_j + b_j T_j(w_0 + w_1 z), \quad j = 0, \dots, m,$$

where the parameters a_j and b_j are used to make $R_j(z)$ ($j \geq 2$) a second order consistent polynomial. In [2] they are only of first order for $j = 1, \dots, m-1$. The second order requirement for the intermediate polynomials leads to

$$\begin{aligned}
(2.11) \quad & b_j = \frac{T_j''(w_0)}{[T_j'(w_0)]^2}, \quad j = 2, \dots, m, \\
& a_j = 1 - b_j T_j(w_0), \quad j = 0, \dots, m.
\end{aligned}$$

Note that b_0 and b_1 are still free. We chose them both equal to b_2 . Both following from (2.10) and by applying (2.9) to $y' = Jy$, two inhomogeneous, 3-term recurrence relations for the polynomials $R_j(z)$ are obtained. Identification of these relations determines the integration parameters.

The parameters for the first order scheme are given by

$$\begin{aligned}
(2.12) \quad & \tilde{\mu}_1 = \frac{w_1}{w_0}, \\
& \mu_j = 2w_0 \frac{T_{j-1}(w_0)}{T_j(w_0)}, \quad \nu_j = -\frac{T_{j-2}(w_0)}{T_j(w_0)},
\end{aligned}$$

$$\tilde{\mu}_j = 2w_1 \frac{T_{j-1}(w_0)}{T_j(w_0)}, \quad \tilde{\gamma}_j = 0, \quad j = 2, \dots, m,$$

where

$$w_0 = 1 + \frac{1}{20m^2}, \quad w_1 = \frac{T_m(w_0)}{T'_m(w_0)}.$$

For the second order scheme we now obtain

$$(2.13) \quad \begin{aligned} \tilde{\mu}_1 &= b_1 w_1, \\ \mu_j &= 2w_0 \frac{b_j}{b_{j-1}}, & v_j &= -\frac{b_j}{b_{j-2}}, \\ \tilde{\mu}_j &= 2w_1 \frac{b_j}{b_{j-1}}, & \tilde{\gamma}_j &= -a_{j-1} \tilde{\mu}_j, \quad j = 2, \dots, m, \end{aligned}$$

where

$$w_0 = 1 + \frac{2}{13m^2}, \quad w_1 = \frac{T'_m(w_0)}{T''_m(w_0)}.$$

A numerical comparison of the second order scheme (2.9), (2.13) and the second order scheme from [2], indicates to the conclusion that in general our modification leads to a gain in accuracy. Observe that for linear problems the schemes are identical.

3. NUMERICAL EVALUATION

As mentioned before, the main purpose of this evaluation is to select one method to base an automatic, explicit solver on. In order to get insight into the behaviour of the various methods we applied them to a set of five test equations. Some of these equations served as a test example before [3,5,6,7].

We consider (scalar) equations in two space dimensions of which the exact solutions are known. The time-dependency of these solutions is of different type. The equations include difficulties like: strong non-linearities (to test the stability behaviour), an oscillating solution and a mixed derivative term.

3.1. The test equations

All equations belong to the general class

$$(3.1) \quad u_t = F(t, u, u_{x_1}, u_{x_2}, u_{x_1 x_1}, u_{x_1 x_2}, u_{x_2 x_2})$$

defined on $\{(t, x_1, x_2) \mid 0 \leq t \leq 1, (x_1, x_2) \in \Omega\}$, where Ω is given by either

$$\Omega_1 = \{(x_1, x_2) \mid (0 \leq x_1 \leq 1, 0 \leq x_2 \leq \frac{3}{7}) \cup (0 \leq x_1 \leq \frac{4}{7}, \frac{3}{7} < x_2 \leq 1)\}$$

or

$$\Omega_2 = \{(x_1, x_2) \mid 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}.$$

The initial conditions and the boundary conditions, which we always assume to be of Dirichlet type, are obtained from the exact solutions. The space discretization of all equations is performed using standard symmetrical differences on a uniform grid. For Ω_1 we used a grid size $h = 1/21$ and for Ω_2 a grid size $h = 1/20$, resulting in 292 and 361 internal grid points, respectively.

We now summarize the parabolic equations together with their exact solution and the domain they are defined on:

$$\text{I [6]} \quad u_t = \Delta u - e^{-t}(x_1^2 + x_2^2 + 4), \quad \Omega = \Omega_2,$$

$$u(t, x_1, x_2) = 1 + e^{-t}(x_1^2 + x_2^2).$$

$$\text{II [6]} \quad u_t = \frac{x_1 + x_2}{2(1+t)} \Delta(u^3) + \pi(x_1 + x_2) \cos(2\pi t) - \frac{3(x_1 + x_2)^2}{4(1+t)} \sin^3(2\pi t),$$

$$u(t, x_1, x_2) = \frac{1}{2} \sin(2\pi t) (x_1 + x_2). \quad \Omega = \Omega_2$$

$$\text{III [5,7]} \quad u_t = \Delta(u^5), \quad \Omega = \Omega_2$$

$$u(t, x_1, x_2) = [0.8(2t + x_1 + x_2)]^{\frac{1}{4}}.$$

$$\text{IV [3]} \quad u_t = [(1+u)/(1+x_1 x_2 (x_1 + x_2) e^{-t})]^{10} *$$

$$* \left[\left(\frac{1}{2} x_1^2 + x_2^2\right) u_{x_1 x_1} - (x_1^2 + x_2^2) u_{x_1 x_2} + \left(x_1^2 + \frac{1}{2} x_2^2\right) u_{x_2 x_2} \right], \quad \Omega = \Omega_1$$

$$u(t, x_1, x_2) = x_1 x_2 (x_1 + x_2) e^{-t}.$$

$$\begin{aligned}
 \text{v [3]} \quad u_t &= \sqrt{u} \cdot (\Delta u - 2u) + (\frac{1}{2}u - u_{x_1 x_2}) / (1+t), \quad \Omega = \Omega_1 \\
 u(t, x_1, x_2) &= e^{-(x_1 + x_2) / \sqrt{1+t}}.
 \end{aligned}$$

3.2. Strategy and results

The testing strategy is as simple as possible: all equations are integrated by the various methods using a sequence of constant step sizes τ , hence no estimation of the local truncation error is performed. An estimate of the spectral radius σ is delivered beforehand. Except for problem III, this estimate is constant. The number of $f(y)$ -evaluations, m , is minimized with respect to absolute stability requirements, i.e. $\tau\sigma \leq cm^2$ (see (1.3)). For problem III, where we made a t -dependent estimate for σ , m is minimized at each step. The estimates of the spectral radii are specified at the tables of results. Besides, for each problem an accuracy/computational effort-plot has been made. The accuracy is measured by sd , defined as the minimum of $-^{10}\log$ (absolute error at $t=1$) over all grid points. The computational effort is measured by $\int ev$, being the total number of $f(y)$ -evaluations. The starting values for the k -step methods, which start integrating at $t = (k-1)\tau$, are given by the exact solution. For $k \geq 2$ the number of $\int ev$ has been increased with the number of $f(y)$ -evaluations by which the multistep formula otherwise would have been benefitted. Finally we mention that the values of h and τ are chosen in such a way that for the testset under consideration the time-integration error dominates the space-discretization error.

The results of the experiments are presented in tables 3.1 - 3.5 and the corresponding figures. In the figures the dotted lines refer to first order results, while the continued lines refer to results of order two. The integers in the figures refer to k .

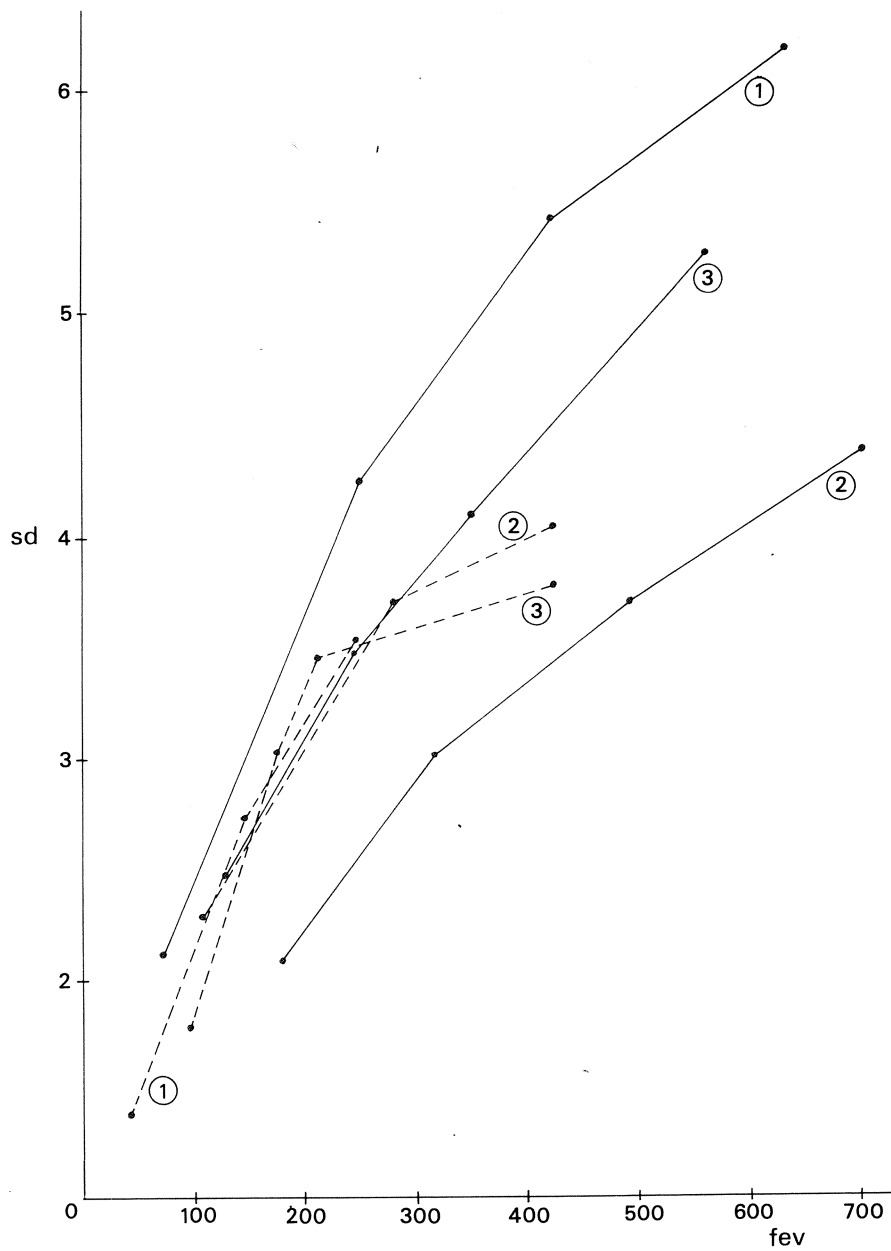


Table 3.1
Example I; $\sigma = 3200$

first order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	1.39	41				
1/12	2.74	144	2.28	108	1.79	96
1/35	3.52	245	3.11	210	3.02	175
1/70			3.73	280	3.47	210
1/140			4.05	420	3.78	420

second order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	2.12	71				
1/12	4.27	252	2.10	180	2.42	132
1/35	5.44	420	3.02	315	3.45	245
1/70	6.21	630	3.72	490	4.11	350
1/140			4.41	700	5.28	560

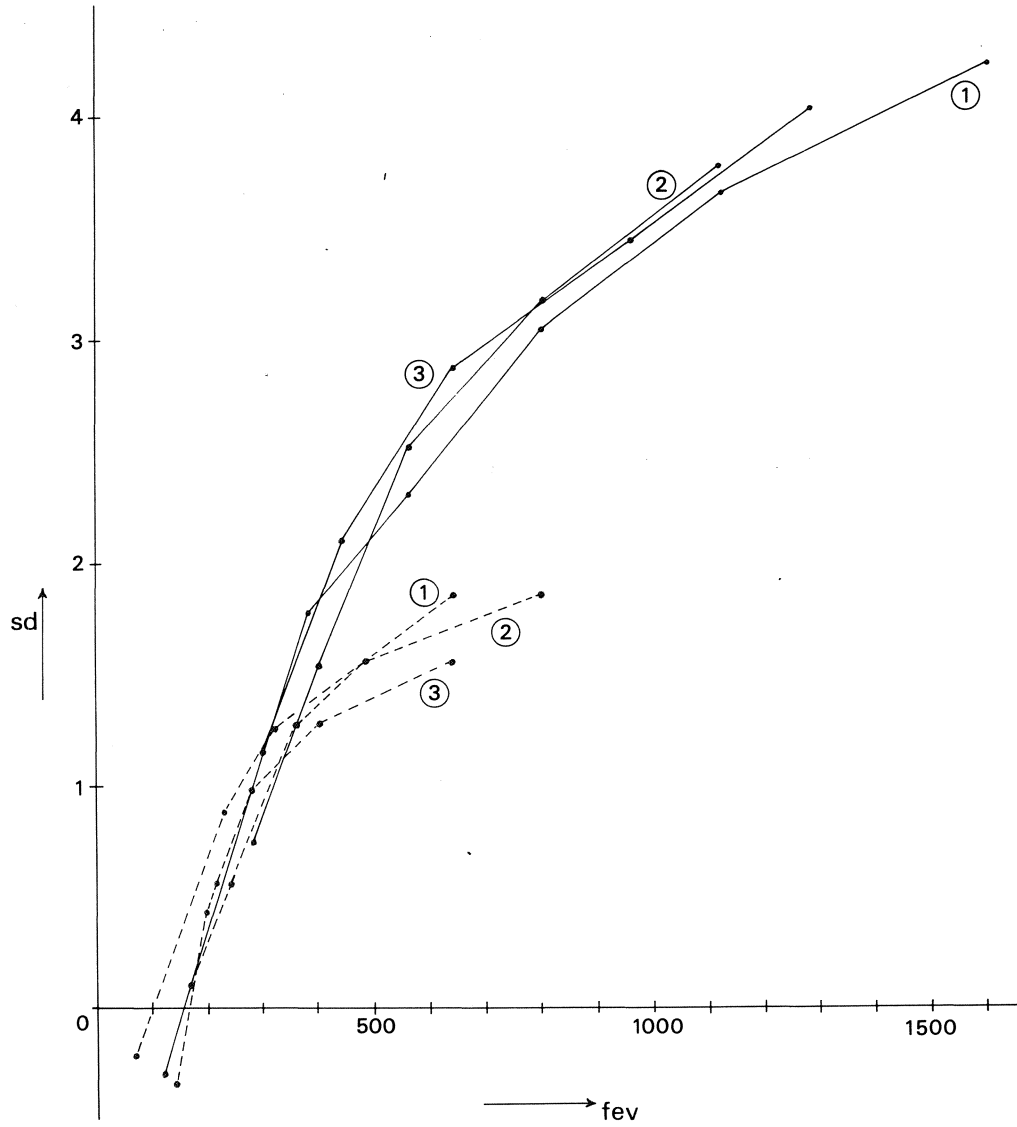


Table 3.2
Example II; $\sigma = 9600$

first order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	-.23	71				
1/10	.87	230	.10	170	-.35	140
1/20	1.25	320	.56	240	.44	200
1/40	1.56	480	1.26	360	.97	280
1/80	1.86	640	1.56	480	1.28	400
1/160			1.86	800	1.58	640

second order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	-.30	122				
1/10	1.76	380	.74	280	.54	210
1/20	2.31	560	1.53	400	1.13	300
1/40	3.06	800	2.51	560	2.11	440
1/80	3.67	1120	3.19	800	2.86	640
1/160	4.26	1600	3.79	1120	3.45	960
1/320					4.05	1280

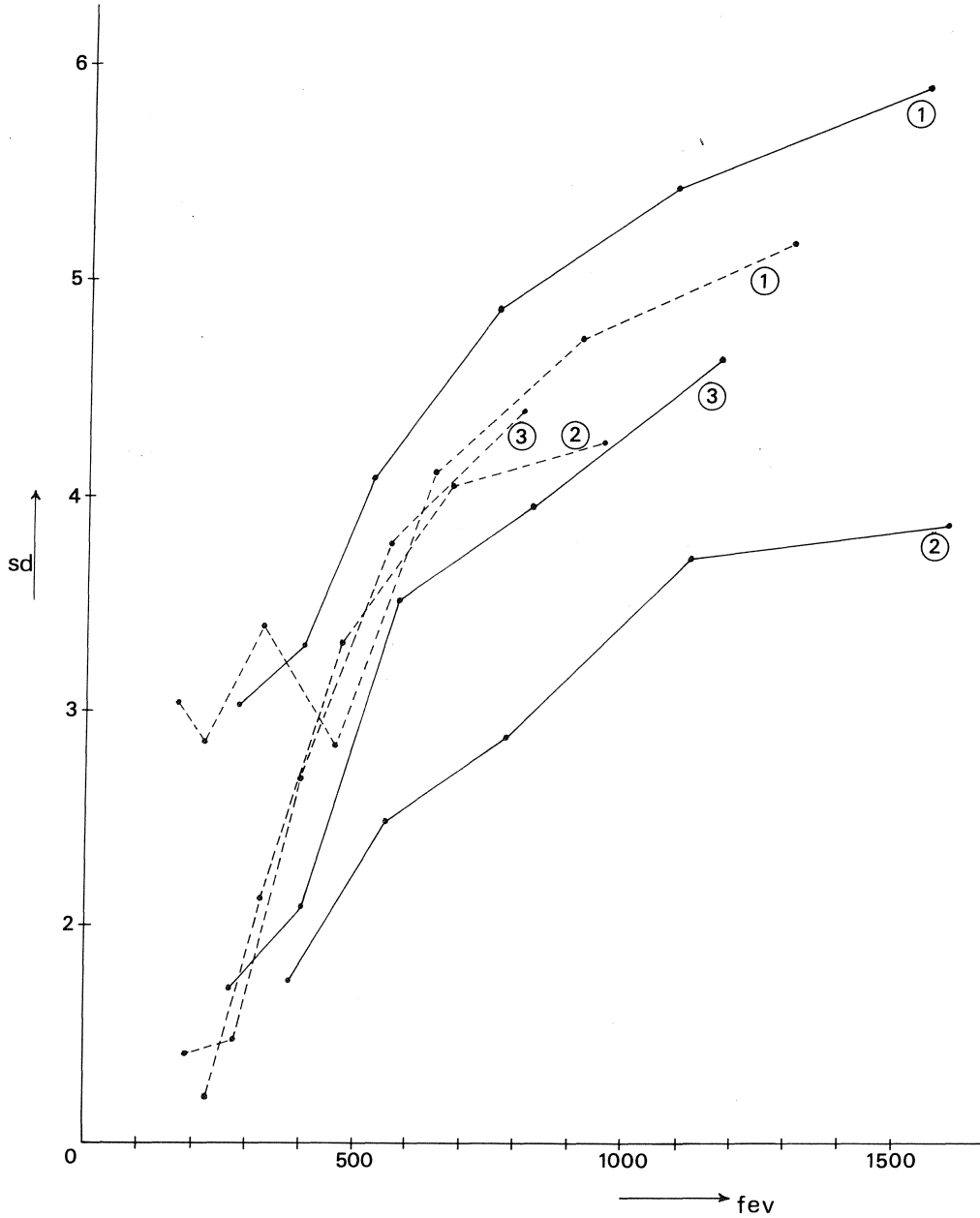


Table 3.3
Example III; $\sigma = 25600(1+t)$

first order

$\frac{k}{\tau}$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	3.03	163				
1/2	2.85	215				
1/5	3.40	325	1.22	224	1.40	188
1/10	2.85	455	2.13	324	1.48	273
1/20	4.13	639	3.33	467	2.72	389
1/40	4.76	910	4.07	669	3.78	561
1/80	5.22	1294	4.27	960	4.41	805

second order

$\frac{k}{\tau}$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	3.02	280				
1/2	3.32	400				
1/5	4.10	526	1.76	377	1.72	277
1/10	4.89	756	2.50	547	2.11	399
1/20	5.46	1083	2.90	780	3.52	573
1/40	5.95	1545	3.75	1114	3.98	819
1/80	7.05	2200	3.91	1592	4.66	1174

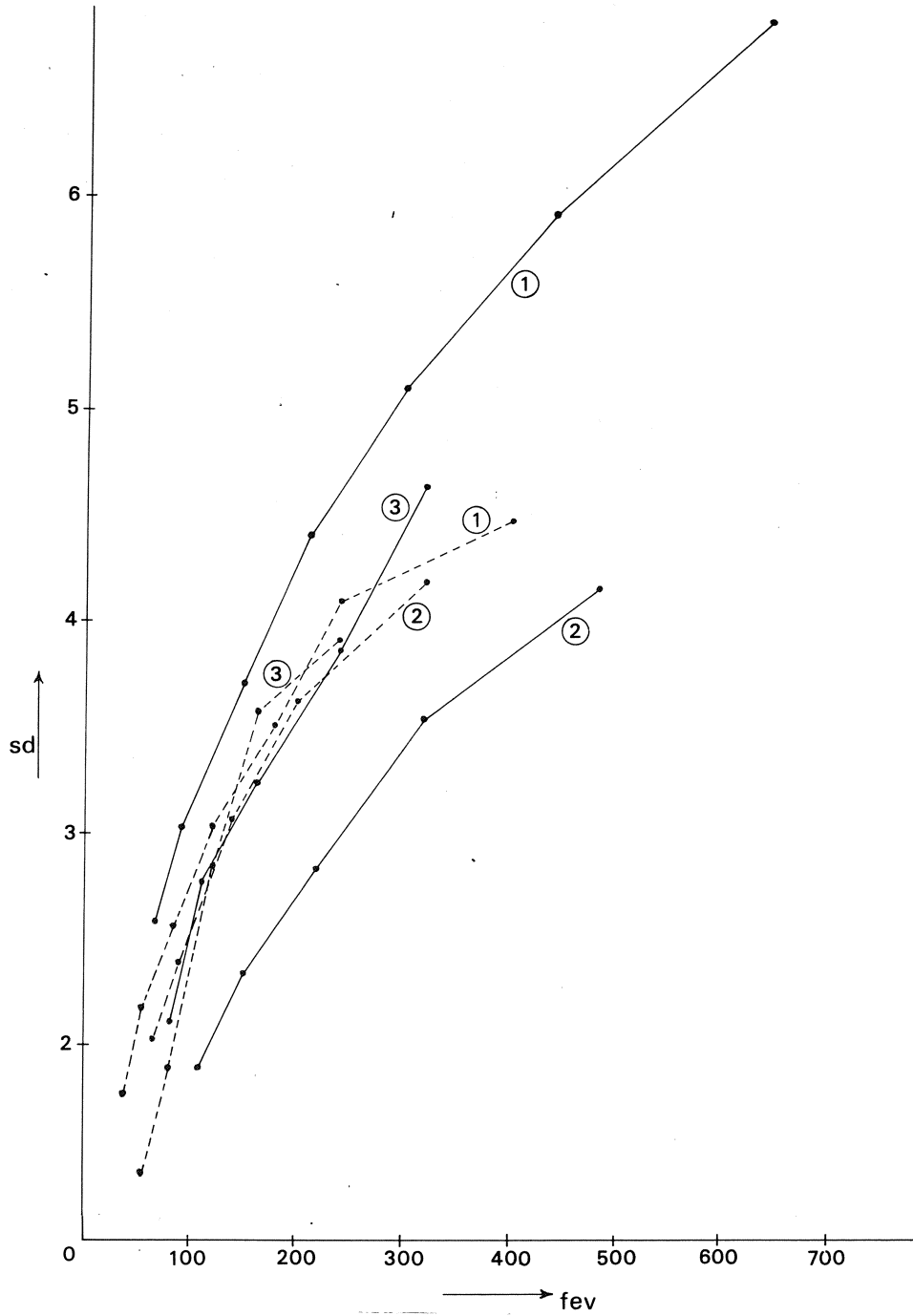


Table 3.4

Example IV; $\sigma = 2740$

first order

second order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	1.76	38				
1/2	2.17	54				
1/5	2.57	85	2.03	65	1.40	55
1/10	3.05	120	2.40	90	1.90	80
1/20	3.51	180	3.08	140	2.84	120
1/40	4.11	240	3.67	200	3.62	160
1/80	4.50	400	4.20	320	3.93	240

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	2.58	65				
1/2	3.03	92				
1/5	3.72	150	1.89	110	2.11	80
1/10	4.42	210	2.35	150	2.77	110
1/20	5.13	300	2.84	220	3.25	160
1/40	5.96	440	3.54	320	3.87	240
1/80	6.86	640	4.17	480	4.65	320

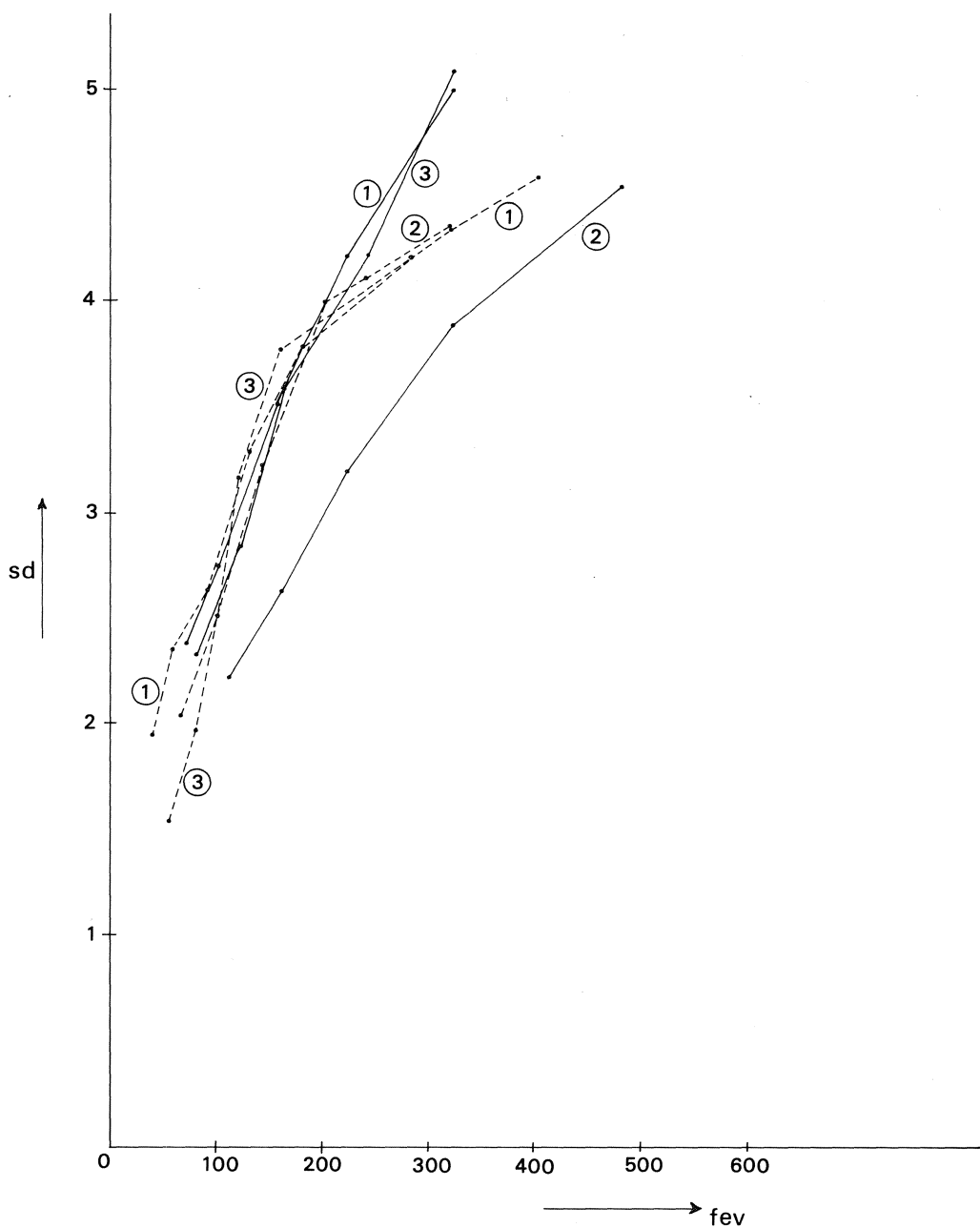


Table 3.5
Example V; $\sigma = 3000$

first order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	1.94	40				
1/2	2.35	56				
1/5	2.63	90	2.03	65	1.53	55
1/10	3.29	130	2.51	100	1.96	80
1/20	3.77	180	3.23	140	3.15	120
1/40	4.21	280	3.97	200	3.76	160
1/80	4.59	400	4.33	320	4.08	240

second order

$\tau \backslash k$	1		2		3	
	sd	fev	sd	fev	sd	fev
1	2.38	68				
1/2	2.73	98				
1/5	3.50	155	2.22	110	2.31	80
1/10	4.19	220	2.63	160	2.84	120
1/20	5.00	320	3.20	220	3.58	160
1/40			3.88	320	4.22	240
1/80			4.54	480	5.08	320

4. CONCLUDING REMARKS

This paper reports on a numerical performance evaluation of 6 closely related explicit Runge-Kutta methods developed for the integration of semi-discrete parabolic initial boundary value problems. We have attempted to show the efficiency (accuracy versus computational effort) of the various methods in relation to each other. Our conclusion can be summarized in 4 points.

- (i) With respect to stability all methods show a reliable behaviour. For example, without difficulties the rather non-linear problem III can be integrated with large step sizes and, consequently, with rather high values of m .
- (ii) There is not much difference between the first order results of the 3 classes of methods. On the other hand, the second order results significantly differ. Here the 1-step method stands out to be superior, whereas the 2-step method falls behind.
- (iii) In most cases the second order 2-step method is not even competitive with the first order schemes. Apart from the second order 2-step method, for low accuracies all methods show a more or less equal efficiency. However, as to be expected, for the more higher accuracies the second order schemes are preferable, in particular the 1-step scheme.
- (iv) Despite the fact that the absolute stability boundaries of the 3-step schemes are approximately 3 times larger than the corresponding 1-step boundaries, the 3-step schemes do not show a larger efficiency. On the contrary, in most cases the efficiency of the 1-step schemes is larger. We thus recommendate the second order 1-step method, also regarding that this method is easier to implement and requires less storage.

Note added in proof. For interesting problem classes the efficiency of the RKC-formulas may be significantly enlarged by linearizing (see [6,7]) the expressions $f(y_{n+1}^{(j)})$ to

$$(*) \quad f(y_n) + J(y_n)(y_{n+1}^{(j)} - y_n).$$

Thus $m-1$ evaluations of the non-linear function $f(y)$ are replaced by $m-1$ matrix-vector operations and 1 Jacobian matrix evaluation. Shortly before printing of this report the authors made a small comparison between the linearized 1-step and 3-step formulas of order 2. From this comparison we learned that conclusion (iv) given above is not valid for the linearized formulas. On the contrary, after linearization the accuracy of the 1-step formula strongly decreased, whereas the 3-step formula maintained its accuracy.

It thus seems that when using the linearization (*), the 3-step schemes turn out to be superior. This difference in performance should be subject of further investigation.

ACKNOWLEDGEMENT. *The authors gratefully acknowledge P.J. van der Houwen for his stimulating remarks and continuing interest during the course of the investigations.*

REFERENCES

- [1] HOUWEN, P.J. VAN DER, *Construction of integration formulas for initial value problems*, North-Holland Publ. Comp., Amsterdam, 1977.
- [2] HOUWEN, P.J. VAN DER & B.P. SOMMEIJER, *On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values*, Report NW 72/79, Mathematisch Centrum, Amsterdam (to appear in ZAMM), 1979.
- [3] HOUWEN, P.J. VAN DER, SOMMEIJER, B.P. & J.G. VERWER, *Comparing time integrators for parabolic equations in two space dimensions with a mixed derivative*, J. of Comp. and Applied Math. 5, no. 2, 1979.
- [4] LAMBERT, J.D., *Computational methods in ordinary differential equations*, John Wiley & Sons, London, 1973.
- [5] RICHTMYER, R.D. & K.W. MORTON, *Difference methods for initial value problems*, Interscience, New York, 1967.

- [6] SOMMEIJER, B.P. & P.J. VAN DER HOUWEN, *On the economization of stabilized Runge-Kutta methods with applications to parabolic initial value problems*, Report NW 74/79, Mathematisch Centrum, Amsterdam (to appear in ZAMM), 1979.
- [7] VERWER, J.G., *On a class of explicit three-step Runge-Kutta methods with extended real stability intervals*, Report NW 77/79, Mathematisch Centrum, Amsterdam (prepublication) 1979.
- [8] VERWER, J.G., *An implementation of a class of stabilized explicit methods for the time integration of parabolic equations*, ACM Transactions on Mathematical Software, June 1980.