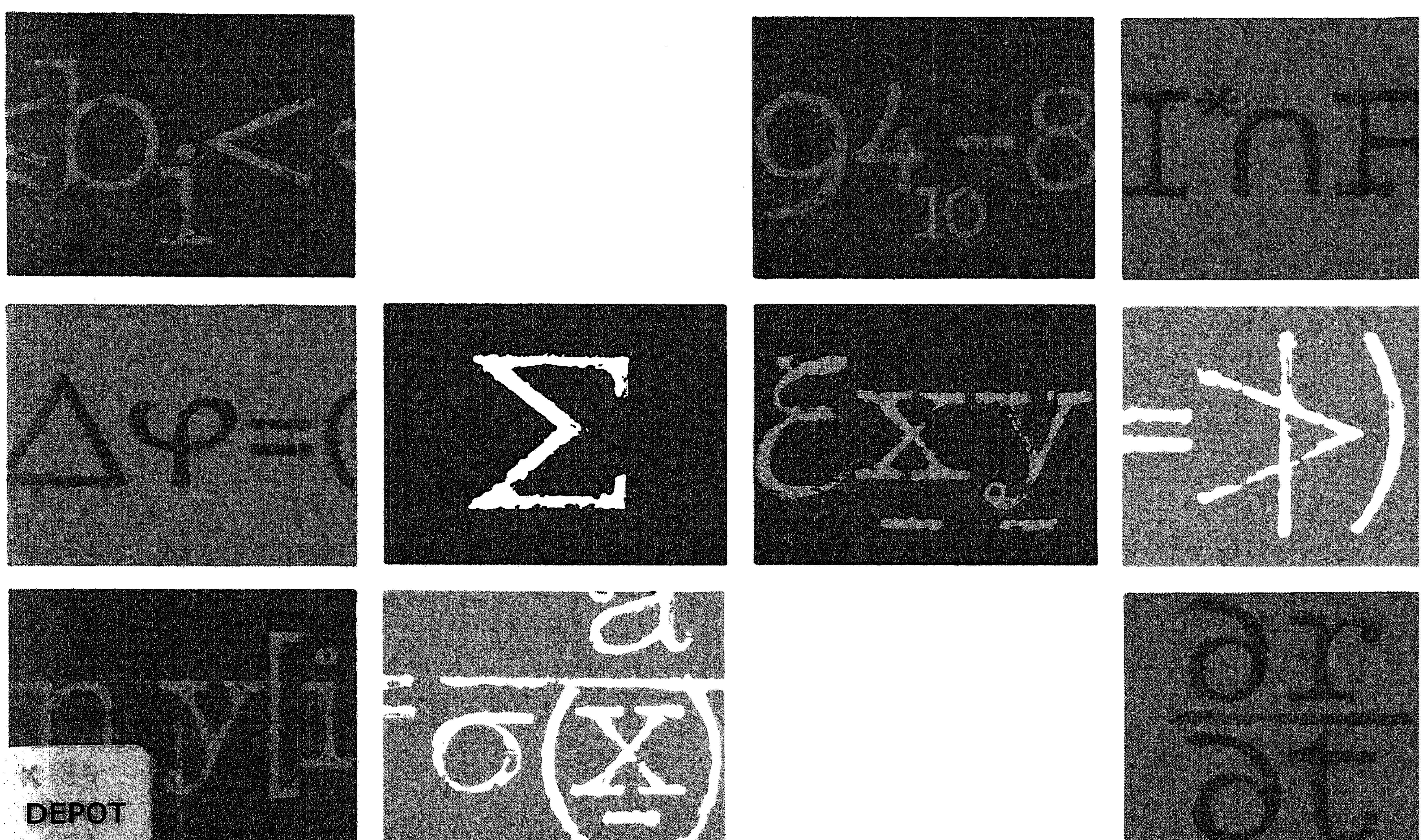
AUTOMATIC NUMERICAL INTEGRATION

J. A. ZONNEVELD



DEPOT
08850

MATHEMATICAL CENTRE TRACTS 8

Zonneveld, Tract 8, 2nd ed.

References

1. Hildebrand, F.B., Introduction to numerical analysis, McGraw-Hill, New York 1956.
2. Hartree, D.R., Numerical analysis, Oxford Univ. Press, Oxford 1952.
3. Nordsieck, A., On numerical integration of ordinary differential equations, MTAC 16(1962) 22-49.
4. Fröberg, E., On the solution of ordinary differential equations with digital computing machines, Fysiogr.Sällsk.Lund.Förhdl. 20 nr 11(1950).
5. Naur, P., Revised report on the algorithmic language ALGOL 60, Num. Math. 4(1963) 420-453, Comm. ACM 6(1963) 1-17, Regnecentralen, Copenhagen 1962.
6. Runge, C., Ueber die numerische Auflösung von Differentialgleichungen, Math. Ann. 46(1895) 166-178.
7. Heun, K., Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen, Zeitschr. Math. u. Phys. 45(1900) 23-38.
8. Kutta, W., Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, Zeitschr. Math. u. Phys. 46(1901) 435-453.
9. Nyström, E.J., Ueber die numerische Integration von Differentialgleichungen, Act. Soc. Sci. Fennicac, Tom L, no 13(1925).
10. Gill, S., A process for the step-by-step integration of differential equations in an automatic digital computing machine, Proc. Camb. Phil. Soc. 47(1950) 96-108.
11. Huťa, A., Une amélioration de la méthode Runge-Kutta-Nyström pour la résolution numérique des équations differentielles du premier ordre, Act.Fac.Rer.Nat.Univ. Comenianae Math. 1(1956) 201-224.
12. Huťa, A., Contribution à la formule de sixième ordre dans la méthode de Runge-Kutta-Nyström, Act.Fac.Rer.Nat.Univ. Comenianae Math. 2(1957) 21-23.
13. Zurmühl, R., Zur numerischen Integration gewöhnlicher Differentialgleichungen zweiter und höherer Ordnung, ZAMM 20(1940) 104-116.
14. Zurmühl, R., Runge-Kutta Verfahren zur numerischen Integration van Differentialgleichungen n-ter Ordnung, ZAMM 28(1948) 173-182.

15. Lemaître, G., Remarques sur certaines méthodes d'intégration des systèmes d'équations differentielles, Colloque sur l'analyse numérique, Gauthier-Villars, Paris 1961.

16. Minorsky, N., Nonlinear oscillations, van Nostrand, New York 1962.

17. Astronomical Papers, Vol XII, Coordinates of the five outer planets, United States Government Printing Office, Washington 1951.

18. Koch, H., Ueber die praktische Anwendung der Runge-Kuttaschen Methode zur numerischen Integration von Differentialgleichungen, Diss. Göttingen 1909.

19. Rutishauser, H., Ueber die Instabilität von Methoden zur Integration gewöhnlicher Differentialgleichungen, ZAMP 3(1942) 65-74.

20. Certaine, J., The solution of ordinary differential equations with large time constants, Ralston und Wilf: Mathematical methods for digital Computers, Wiley, New York 1960, p. 128-132.

21. Bard, A., F. Ceschino, J. Kuntzmann, P.J. Laurent, Formules de base de la méthode de Runge-Kutta, Chiffres 4(1961) 31-36.

22. Ceschino, F., Evaluation de l'erreur par pas dans les problèmes différentiels, Chiffres 5(1962) 223-229.

23. Scraton, R.E., The numerical solution of second-order differential equations not containing the first derivative explicitly, Comp.J. 6(1963) 368-370.

24. Warten, R.M., Automatic step-size control for Runge-Kutta integration, IBM J. Res. 7(1963) 340-341.

MATHEMATICAL CENTRE TRACTS

8

AUTOMATIC NUMERICAL INTEGRATION

BY

J. A. ZONNEVELD

2nd Edition

RA

MATHEMATISCH CENTRUM AMSTERDAM

1970

First edition in 1964

Contents

CHAPTER 0

Introduction

In this paper we derive and test some formulas to solve numerically sets of ordinary differential equations of first and second order with a one-point boundary condition. Several methods are available for the numerical solution of ordinary differential equations. We mention finite-difference methods, eg. Adams method (see Hildebrand [1]), Taylor series method (see Hartree [2]) and its variants like the Nordsieck method [3], and Runge-Kutta methods. All methods use a finite step in the independent variable. Change of this step is difficult with finite-difference methods, but easy with the other ones. The Taylor series method breaks down as soon as the functions occurring in the differential equations are complicated and cannot easily be differentiated a number of times. In the Nordsieck and Runge-Kutta methods the computation of the sum of a series, agreeing with the Taylor expansion of the solution of the differential equation in a given number of terms, is done numerically, thereby avoiding the differentiation of the functions occurring in the differential equation.

In the sequel we shall construct a set of Runge-Kutta formulas that are suitable for integration with varying step-size. The use of a varying step-size is important in reducing the amount of computational work to be done; this work is proportional to the number of steps performed. Depending on the character of the differential equation, for a given accuracy, the acceptable step-size may vary considerably over the interval of integration. With regard to a criterion for the accuracy we make the following remarks. Using finite-difference methods one takes into account as many differences as have an influence on the results to the accuracy specified.
Using the Taylor series method, one can follow at least two strategies: use with a given steplength as many terms of the (convergent)

series to get a result to the accuracy specified; or (this is in most cases the effective way) keep the number of terms fixed and compute the step length according to the accuracy prescribed. Using a Runge-Kutta formula neither of the two strategies is applicable. The use of a varying number of terms is impractical and adjustment of the step-size according to the last term taken into account is impossible as the Runge-Kutta formula gives the sum of the series instead of separate terms. In actual computational processes this difficulty is overcome in the following way.

Suppose we are given the equation

$$y'=f(x,y), \quad y(X)=Y$$

and we want to find $y(X+h)$. We use a Runge-Kutta formula that gives a computed value $\overset{*}{y}(X+h)$ such that the Taylor expansions of $y(X+h)$ and $\overset{*}{y}(X+h)$ agree in n terms. We denote the value obtained by doing two steps of length h by $\overset{*}{y}_h(X+2h)$ and the value obtained by doing one step of length 2h by $\overset{*}{y}_{2h}(X+2h)$. Now it follows that

$$y(X+2h)-\overset{*}{y}_h(X+2h) = 2h^{n+1}c(X),$$

$$y(X+2h)-\overset{*}{y}_{2h}(X+2h) = (2h)^{n+1}c(X),$$

if we assume that $c(x)$ varies slowly with x and that terms of order h^{n+2} may be neglected. We then have

$$y(X+2h) = (2^n \overset{*}{y}_h - \overset{*}{y}_{2h})/(2^n-1)$$

and

$$2h^{n+1}c(X)=(\overset{*}{y}_h-\overset{*}{y}_{2h})(2^n-1).$$

Although this quantity is of the order h^{n+1}, it is in general not equal to the h^{n+1}-term in the expansion of $y(X+2h)$. In spite of this, in practice the value of this quantity is used to determine whether the step is accepted or not. [4], [5].

In the following chapters we shall construct a set of Runge-Kutta formulas that give, in addition to the increment of the dependent variable(s), the value(s) of the last term(s) taken into account in the Taylor series, thus controlling the accuracy without performing the integration twice.

CHAPTER 1

Runge-Kutta method

Since the basis of the Runge-Kutta method is described in several books on numerical analysis [1], we shall here give only an outline of the idea.

Suppose we are given the first-order differential equation

$$y'=f(x,y)\,;\; y(X)=Y,$$

and we want to find y(X+h), then the Runge-Kutta idea is the following:

Compute the quantities $k_i=hf(x_i,y_i)$ in succession in the points (x_0,y_0), (x_1,y_1), ..., (x_{m-1},y_{m-1}), where $x_i=X+M_ih$; $M_0=0$ and where

$$y_i=Y+\sum_{l=0}^{i-1} L_{il}k_l.$$

Here M_i and L_{il} are constants of the process, independent of the differential equation under consideration. Form a weighted mean

$$dy=\sum_{i=0}^{m-1} a_ik_i,$$

being an approximation to the increment of y corresponding to the increment h in x. There are a number of parameters: m, the number of points used; M_i; L_{il}; and the weights a_i. These parameters must be determined from the following conditions. Let y(X+h) be expanded in a Taylor series about X and let Y+dy be expanded in a power series in h. We now want to find the minimal number of points (x_i,y_i), the constants M_i and L_{il} and the weights a_i such that the two series agree to a given number of terms. This results in a set of nonlinear equations. In practice the problem is reversed in that the number of points is given and the order of approximation (i.e. the exponent of h in the last term agreeing) is made as high as possible. There exist several Runge-Kutta type formulas given by Runge [6], Heun [7], Kutta [8], Nyström [9], Gill [10], Huťa [11] [12], Zurmühl [13] [14]. For second, third and fourth order approximation respectively two, three and four points are necessary. Fifth order approximation however seems impossible with five points, Lemaitre [15], but six point formulas do exist [9]. Since in several cases there are more unknowns than equations, variants exist for the same order and with the same number of points. Nyström [9] and Zurmühl [13] [14] give Runge-Kutta type formulas for differential equations of orders two, three and four. There are also formulas for the solution of systems of differential equations.

For a system of first-order differential equations

$$dy_j/dx = f_j(x, y_1, \ldots, y_n); \quad y_j(X) = Y_j; \quad j = 1(1)n,$$

the Runge-Kutta scheme is

$$k_{ij} = hf_j(X + M_i h, Y_1 + \sum_{l=0}^{i-1} L_{il} k_{l1}, \ldots, Y_n + \sum_{l=0}^{i-1} L_{il} k_{ln});$$

$$y_j^*(X+h) = Y_j + \sum_{i=0}^{m-1} a_i k_{ij},$$

where the unknowns m, M_i, L_{il} and a_i must be determined in such a way that the expansions of $y_j^*(X+h)$ with respect to h agree with the expansions of $y_j(X+h)$ to a given number of terms. Our object is to compute not only the increment of y_j, corresponding to an increment h of x but also the last term of the Taylor expansions that has been taken into account. If we suppose the above m-point formula to be of the order r and if we denote the term with h^r in $y_j(X+h)$ by $th^r dy_j$, we want to compute

$$th^r dy_j = \sum_{i=0}^{m-1} b_i k_{ij}.$$

As we shall see later it turns out that it is possible to find a_i as well as b_i at the cost of increasing m by one or two.

For a system of second-order equations

$dy_j^2/dx^2 = f_j(x, y_1, \ldots, y_n, dy_1/dx, \ldots, dy_n/dx)$; $y_j(X) = Y_j$; $dy_j(X)/dx = Y_j'$; $j = 1(1)n$,

the computing scheme is

$$k_{ij} = hf_j(X + M_i h, Y_1 + M_i Y_1' h + h \sum_{l=0}^{i-1} K_{il} k_{l1}, \ldots, Y_n + M_i Y_n' h + h \sum_{l=0}^{i-1} K_{il} k_{ln},$$
$$Y_1' + \sum_{l=0}^{i-1} N_{il} k_{l1}, \ldots, Y_n' + \sum_{l=0}^{i-1} N_{il} k_{ln});$$

$$y_j^*(X+h) = Y_j + h(Y_j' + \sum_{i=0}^{m-1} b_i k_{ij});$$

$$y_j'^*(X+h) = Y_j' + \sum_{i=0}^{m-1} c_i k_{ij},$$

where again the unknowns must be determined by equating the power series for $y_j(X+h)$ and $y_j^*(X+h)$, and also those for $y_j'(X+h)$ and $y_j'^*(X+h)$. Here we have some freedom since we can choose the order of approximation in the two series differently. Nyström[9] gives one formula where the order of approximation for y' is four and that for y is five. We think however, that, if unequal orders are to be used, it would perhaps be better to use a higher order for y' than for y on the ground that, an error in y' has, in the long run, a greater influence on y than an error in y itself. We shall see, however, that given an certain amount of computational effort, measured by the number of evaluations of f(x,y) per step of integration, the order obtained in y is automatically equal to the highest one obtainable in y'. To obtain a higher order of approximation in y requires again more evaluations. In the sequel, therefore, we shall deal only with equal orders. Again, letting the above m-point formula be of order r and the r^{th}-order term in $y_j(X+h)$ be $th^r dy_j$, and the r^{th}-order term in $y_j'(X+h)$ be $th^r dy_j'$, we then want to compute:

$$th^r dy_j = h \sum_{i=0}^{m-1} B_i k_{ij};$$

$$th^r dy_j' = \sum_{i=0}^{m-1} c_i k_{ij}.$$

If the functions f_j do not depend on dy_j/dx, the problem is somewhat simpler in that the unknowns N_{il} do not occur.

In the next three chapters we shall derive and solve, for various orders of approximation the equations for the cases

$$dy_j/dx = f_j(x, y_1, \ldots, y_n);$$

$$d^2y_j/dx^2 = f_j(x, y_1, \ldots, y_n, dy_1/dx, \ldots, dy_n/dx);$$

$$d^2y_j/dx^2 = f_j(x, y_1, \ldots, y_n).$$

CHAPTER 2

First-order equations

Suppose we have the set of differential equations

$$dy_j/dx = f_j(x,y_1,\dots,y_n);\; y_j(X)=Y_j;\; j=1(1)n \tag{1}$$

For the sake of simplicity we prefer the system

$$dy_j/dx = f_j(y_0,y_1,\dots,y_n);\; y_j(X)=Y_j;\; j=0(1)n, \tag{2}$$

which is equivalent to (1) if we put $x=y_0$, whence

$f_0(y_0,y_1,\dots,y_n)=1$ and $Y_0=X$

The Runge-Kutta scheme of order r with m points is

$$k_{ij} = hf_j(Y_0+\sum_{l=0}^{i-1} L_{il}k_{l0},\dots,Y_n+\sum_{l=0}^{i-1} L_{il}k_{ln});\; j=0(1)n;\; l=0(1)m-1; \tag{3}$$

$$y_j^*(X+h) = Y_j+\sum_{i=0}^{m-1} a_i k_{ij}; \tag{4}$$

$$th^r dy_j = \sum_{i=0}^{m-1} b_i k_{ij}. \tag{5}$$

We now must develop $y_j^*(X+h)$ and $th^r dy_j$ in power series in h.
To this end we introduce some notations:

i) we do not write the arguments of f_j and of its derivatives, if they are $(Y_0,Y_1,\dots,Y_n)$;

ii) $f_{jkl..}$ shall mean $f_j(y_0,y_1,\dots,y_n)$, differentiated with respect to y_k, y_l etc. in $(Y_0,Y_1,\dots,Y_n)$;

iii) $M_i = \sum_{l=0}^{i-1} L_{il}, M_0=0;$ (6)

iv) L_{il} is defined for $l=0(1)i-1$, otherwise zero;

v) the range of the following summation variables, if not redefined explicitly, is understood to be
$l=1(1)i-1$: $p=1(1)l-1$: $q=1(1)p-1$: $s=0(1)n$: $t=0(1)n$: $u=0(1)n$;

vi) we introduce the differential operator

$$D=\sum_s f_s \frac{\partial}{\partial y_s} \qquad (7)$$

We retain terms of order five and find for $i \geq 4$

$$\begin{aligned}
k_{ij}= {} & hf_j+h^2M_iDf_j+h^3(\tfrac{1}{2}M_i^2D^2f_j+(\sum_l L_{il}M_l)\sum_s Df_sf_{js})+\\
& h^4(\tfrac{1}{6}M_i^3D^3f_j+\tfrac{1}{2}(\sum_l L_{il}M_l^2)\sum_s D^2f_sf_{js}+M_i(\sum_l L_{il}M_l)\sum_s Df_sDf_{js}\\
& +(\sum_l L_{il}\sum_p L_{lp}M_p)\sum_s\sum_t Df_sf_{ts}f_{jt})+h^5(\tfrac{1}{24}M_i^4D^4f_j+\\
& \tfrac{1}{6}(\sum_l L_{il}M_l^3)\sum_s D^3f_sf_{js}+\tfrac{1}{2}M_i(\sum_l L_{il}M_l^{\,2})\sum_s D^2f_sDf_{js}+\\
& \tfrac{1}{2}M_i^2(\sum_l L_{il}M_l)\sum_s Df_sD^2f_{js}+\tfrac{1}{2}(\sum_l L_{il}M_l)^2\sum_s\sum_t Df_sDf_tf_{jst}+\\
& \tfrac{1}{2}(\sum_l L_{il}\sum_p L_{lp}M_p^2)\sum_s\sum_t D^2f_sf_{ts}f_{jt}+(\sum_l L_{il}M_l\sum_p L_{lp}M_p)\\
& \sum_s\sum_t Df_sDf_{ts}f_{jt}+M_i(\sum_l L_{il}\sum_p L_{lp}M_p)\sum_s\sum_t Df_sf_{ts}Df_{jt}+\\
& (\sum_l L_{il}\sum_p L_{lp}\sum_q L_{pq}M_q)\sum_s\sum_t\sum_u Df_sf_{ts}f_{ut}f_{ju}).
\end{aligned} \qquad (8)$$

For $i < 4$ the expression for k_{ij} reduces to:

$$k_{0j}= hf_j; \qquad (9)$$

$$k_{1j}= hf_j+h^2M_1Df_j+\tfrac{1}{2}h^3M_1^2D^2f_j+\tfrac{1}{6}h^4M_1^3D^3f_j+\tfrac{1}{24}h^5M_1^4D^4f_j; \qquad (10)$$

$$\begin{aligned}
k_{2j}= {} & hf_j+h^2M_2Df_j+h^3(\tfrac{1}{2}M_2^2D^2f_j+L_{21}M_1\sum_s Df_sf_{js})+\\
& h^4(\tfrac{1}{6}M_2^3D^3f_j+\tfrac{1}{2}L_{21}M_1^2\sum_s D^2f_sf_{js}+M_2L_{21}M_1\sum_s Df_sDf_{js})+\\
& h^5(\tfrac{1}{24}M_2^4D^4f_j+\tfrac{1}{6}L_{21}M_1^3\sum_s D^3f_sf_{js}+\tfrac{1}{2}M_2L_{21}M_1^2\sum_s D^2f_sDf_{js}+\\
& \tfrac{1}{2}M_2^2L_{21}M_1\sum_s Df_sD^2f_{js}+\tfrac{1}{2}L_{21}^2M_1^2\sum_s\sum_t Df_sDf_tf_{jst});
\end{aligned} \qquad (11)$$

$$k_{3j}= hf_j+h^2M_3Df_j+h^3\Big(\frac{1}{2} M_3^2D^2f_j+\Big(\sum_{l} L_{3l}M_l\Big) \sum_{s} Df_s f_{js}\Big)+$$

$$h^4\Big(\frac{1}{6} M_3^3D^3f_j+\frac{1}{2} \Big(\sum_{l} L_{3l}M_l^2\Big) \sum_{s} D^2f_s f_{js}+M_3\Big(\sum_{l} L_{3l}M_l\Big) \sum_{s} Df_s Df_{js}$$

$$+L_{32}L_{21}M_1 \sum_{s} \sum_{t} Df_s f_{ts} f_{jt}\Big)+h^5\Big(\frac{1}{24} M_3^4D^4f_j+\frac{1}{6} \Big(\sum_{l} L_{3l}M_l^3\Big) \sum_{s} D^3f_s f_{js}$$

$$+\frac{1}{2} M_3 \Big(\sum_{l} L_{3l}M_l^2\Big) \sum_{s} D^2f_s Df_{js}+\frac{1}{2} M_3^2\Big(\sum_{l} L_{3l}M_l\Big) \sum_{s} Df_s D^2f_{js}+$$

$$\frac{1}{2}\Big(\sum_{l} L_{3l}M_l\Big)^2 \sum_{s} \sum_{t} Df_s Df_t f_{jst}+\frac{1}{2} L_{32}L_{21}M_1^2 \sum_{s} \sum_{t} D^2f_s f_{ts} f_{jt}+$$

$$L_{32}M_2L_{21}M_1 \sum_{s} \sum_{t} Df_s Df_{ts} f_{jt}+M_3L_{32}L_{21}M_1 \sum_{s} \sum_{t} Df_s f_{ts} Df_{jt}\Big). \quad (12)$$

Furthermore we have the Taylor series:

$$y_j(X+h)= Y_j+hf_j+\frac{1}{2} h^2Df_j+\frac{1}{6} h^3\Big(D^2f_j+ \sum_{s} Df_s f_{js}\Big)+ \frac{1}{24} h^4\Big(D^3f_j+$$

$$\sum_{s} D^2f_s f_{js}+3 \sum_{s} Df_s Df_{js}+ \sum_{s} \sum_{t} Df_s f_{ts} f_{jt}\Big)+\frac{1}{120} h^5\Big(D^4f_j+$$

$$\sum_{s} D^3f_s f_{js}+4 \sum_{s} D^2f_s Df_{js}+6 \sum_{s} Df_s D^2f_{js}+3 \sum_{s} \sum_{t} Df_s Df_t f_{jst}+$$

$$\sum_{s} \sum_{t} D^2f_s Df_{ts} f_{jt}+3 \sum_{s} \sum_{t} Df_s Df_{ts} f_{jt}+4 \sum_{s} \sum_{t} Df_s f_{ts} Df_{jt}+$$

$$\sum_{s} \sum_{t} \sum_{u} Df_s f_{ts} f_{ut} f_{ju}\Big). \quad (13)$$

Equating coefficients of powers of h in $y_j(X+h)$ and in $y_j^{*}(X+h)= Y_j+\sum_{i=0}^{m-1} a_i k_{ij}$, we find

$$h: \quad \sum_{i} a_i = 1 \quad (14a)$$

$$h^2: \quad \sum_{i} a_i M_i = \frac{1}{2} \quad (15a)$$

$$h^3: \quad \sum_{i} a_i M_i^2 = \frac{1}{3} \quad (16a)$$

$$\sum_i a_i \sum_l L_{il} M_l = \frac{1}{6} \quad (17a)$$

h^4: $$\sum_i a_i M_i^3 = \frac{1}{4} \quad (18a)$$

$$\sum_i a_i \sum_l L_{il} M_l^2 = \frac{1}{12} \quad (19a)$$

$$\sum_i a_i M_i \sum_l L_{il} M_l = \frac{1}{8} \quad (20a)$$

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp} M_p = \frac{1}{24} \quad (21a)$$

h^5: $$\sum_i a_i M_i^4 = \frac{1}{5} \quad (22a)$$

$$\sum_i a_i \sum_l L_{il} M_l^3 = \frac{1}{20} \quad (23a)$$

$$\sum_i a_i M_i \sum_l L_{il} M_l^2 = \frac{1}{15} \quad (24a)$$

$$\sum_i a_i M_i^2 \sum_l L_{il} M_l = \frac{1}{10} \quad (25a)$$

$$\sum_i a_i \left(\sum_l L_{il} M_l \right)^2 = \frac{1}{20} \quad (26a)$$

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp} M_p^2 = \frac{1}{60} \quad (27a)$$

$$\sum_i a_i \sum_l L_{il} M_l \sum_p L_{lp} M_p = \frac{1}{40} \quad (28a)$$

$$\sum_i a_i M_i \sum_l L_{il} \sum_p L_{lp} M_p = \frac{1}{30} \quad (29a)$$

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp} \sum_q L_{pq} M_q = \frac{1}{120} \quad (30a)$$

For b_i, the weights in the expression for $h^r dy_j$, we find a similar set of equations, the lefthand side being the same with b_i replacing a_i, and the righthand side being zero in the equations resulting from terms of h^t, $t < r$ and being identical to the above righthand side if $t=r$. The equations of this set we denote by (14b)-(30b).

Next we solve the equations for various values of r.

r=1

$k_{0j}=hf_j(Y_0,\ldots,Y_n)$;

$y_j^*(X+h)=Y_j+k_{0j}$;

$th^1dy_j=k_{0j}$.

This is Eulers method.

r=2

The equations are

$a_0+a_1=1$; $b_0+b_1=0$; $a_1M_1=\frac{1}{2}$; $b_1M_1=\frac{1}{2}$,

giving

$a_1=b_1=\frac{1}{2M_1}$; $a_0=1-a_1$; $b_0=-b_1$.

With $M_1=1$ we find

$k_{0j}=hf_j(Y_0,\ldots,Y_n)$;

$k_{1j}=hf_j(Y_0+k_{00},\ldots,Y_n+k_{0n})$;

$y_j^*(X+h)=Y_j+(k_{0j}+k_{1j})/2$;

$th^2dy_j=(-k_{0j}+k_{1j})/2$.

r=3

The three point equations having no solution, we use four points:

$a_0+a_1+a_2+a_3=1$; $\qquad b_0+b_1+b_2+b_3=0$;

$a_1M_1+a_2M_2+a_3M_3=\frac{1}{2}$; $\qquad b_1M_1+b_2M_2+b_3M_3=0$;

$a_1M_1^2+a_2M_2^2+a_3M_3^2=\frac{1}{3}$; $\qquad b_1M_1^2+b_2M_2^2+b_3M_3^2=\frac{1}{3}$;

$a_2L_{21}M_1+a_3(L_{31}M_1+L_{32}M_2)=\frac{1}{6}$; $\qquad b_2L_{21}M_1+b_3(L_{31}M_1+L_{32}M_2)=\frac{1}{6}$.

We now try to find a solution with $a_3=0$, i.e. an integration formula with three points and a last term formula with at most four points. We find

$a_1=(\frac{M_2}{2}-\frac{1}{3})/M_2/(M_2-M_1)$; $a_2=(\frac{1}{3}-\frac{M_1}{2})/M_2/(M_2-M_1)$;

$L_{21}=M_2(M_2-M_1)/M_1/(2-3M_1)$; $L_{20}=M_2(-3M_1^2+3M_1-M_2)/M_1/(2-3M_1)$.

If we choose

$$M_1=L_{10}=\frac{1}{3};\; M_2=\frac{2}{3},$$

we find

$$a_1=0;\; a_2=\frac{3}{4};\; a_0=\frac{1}{4};\; L_{20}=0;\; L_{21}=\frac{2}{3},$$

this being Heun's formula. For b_i we have the equations

$$b_0+b_1+b_2+b_3=0;\; b_1+2b_2+3M_3b_3=0;$$
$$b_1+4b_2+9M_3^2b_3=3;\; 2b_2+b_3(3L_{31}+6L_{32})=\frac{3}{2}.$$

We choose

$$L_{30}=a_0;\; L_{31}=a_1;\; L_{32}=a_2;\; M_3=1.$$

We then find the formula

$$k_{0j}=hf_j(Y_0,\ldots,Y_n);$$
$$k_{1j}=hf_j(Y_0+k_{00}/3,\ldots,Y_n+k_{0n}/3);$$
$$k_{2j}=hf_j(Y_0+2k_{10}/3,\ldots,Y_n+2k_{1n}/3);$$
$$y_j^*(X+h)=Y_j+(k_{0j}+3k_{2j})/4;$$
$$k_{3j}=hf_j(y_0^*(X+h),\ldots,y_n^*(X+h));$$
$$th^3dy_j=(k_{0j}-3k_{2j}+2k_{3j})/2.$$

As extra point for the computation of th^3dy_j we use $y_0^*(X+h),\ldots,y_n^*(X+h)$, this being the starting point of the next integration step, and which has to be computed anyway. In case of acceptance one has to do only three evaluations of f_j per step. A formula of this type is ideal because (since, with a good strategy for extrapolation of the step length, rejections are rare) one has not do any extra work for determining the last term taken into account.

For r=4 and r=5 we did not find formulas using the starting point of the next integration step as "extra" point.

r=4

The equations are

$$\sum_i a_i=1 \quad ; \quad \sum_i b_i=0 \quad ;$$

$$\sum_i a_iM_i=\frac{1}{2} \quad ; \quad \sum_i b_iM_i=0 \quad ;$$

$$\sum_i a_iM_i^2=\frac{1}{3} \quad ; \quad \sum_i b_iM_i^2=0 \quad ;$$

$$\sum_i a_i \sum_l L_{il}M_l=\frac{1}{6} \quad ; \quad \sum_i b_i \sum_l L_{il}M_l=0 \quad ;$$

$$\sum_i a_iM_i^3=\frac{1}{4} \quad ; \quad \sum_i b_iM_i^3=\frac{1}{4} \quad ;$$

$$\sum_i a_i \sum_l L_{il}M_l^2=\frac{1}{12} \quad ; \quad \sum_i b_i \sum_l L_{il}M_l^2=\frac{1}{12} \quad ;$$

$$\sum_i a_iM_i \sum_l L_{il}M_l=\frac{1}{8} \quad ; \quad \sum_i b_iM_i \sum_l L_{il}M_l=\frac{1}{8} \quad ;$$

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp}M_p=\frac{1}{24}; \quad \sum_i b_i \sum_l L_{il} \sum_p L_{lp}M_p=\frac{1}{24}.$$

To obtain the four point Runge-Kutta integration formula with a five point formula for th^4dy_j we put

$a_0=a_3=\frac{1}{6}$; $a_1=a_2=\frac{1}{3}$; $a_4=0$; $M_1=M_2=\frac{1}{2}$; $M_3=1$;

$L_{20}=L_{30}=L_{31}=0$; $L_{10}=L_{21}=\frac{1}{2}$; $L_{32}=1$.

The equations for b_i are

$(b_1+b_2)/2+b_3+b_4M_4=0$; $(b_1+b_2)/4+b_3+b_4M_4^2=0$;

$(b_1+b_2)/8+b_3+b_4M_4^3=\frac{1}{4}$; $b_0+b_1+b_2+b_3+b_4=0$;

$b_4(L_{41}+L_{42})/2+b_4L_{43}=-b_2/4-b_3/2$;

$b_4(L_{41}+L_{42})/4+b_4L_{43}=-b_2/8-b_3/4+\frac{1}{12}$;

$b_4(L_{41}+L_{42})/8+b_4L_{43}M_4=-b_2/8-b_3/2+\frac{1}{8}$;

$b_4L_{42}/2+b_4L_{43}=-b_3/2+\frac{1}{12}$.

We have 8 equations with 9 unknowns. Choosing $M_4=\frac{3}{4}$ we find

$b_0=-\frac{2}{3}$; $b_1=b_2=b_3=2$; $b_4=-\frac{16}{3}$;

$L_{40}=\frac{5}{32}$; $L_{41}=\frac{7}{32}$; $L_{42}=\frac{13}{32}$; $L_{43}=-\frac{1}{32}$.

The fourth order formula is

$k_{0j}= hf_j(Y_0,\ldots,Y_n)$;

$k_{1j}= hf_j(Y_0+k_{00}/2,\ldots,Y_n+k_{0n}/2)$;

$k_{2j}= hf_j(Y_0+k_{10}/2,\ldots,Y_n+k_{1n}/2)$;

$k_{3j}= hf_j(Y_0+k_{20},\ldots,Y_n+k_{2n})$;

$k_{4j}= hf_j(Y_0+(5k_{00}+7k_{10}+13k_{20}-k_{30})/32,\ldots,Y_n+(5k_{0n}+7k_{1n}+13k_{2n}$
$-k_{3n})/32)$;

$y_j^*(X+h)= Y_j+(k_{0j}+2(k_{1j}+k_{2j})+k_{3j})/6$;

$th^4dy_j= \frac{2}{3}(-k_{0j}+3(k_{1j}+k_{2j}+k_{3j})-8k_{4j})$.

<u>r=5</u>

The equations to be solved are (14a)-(30a) and the set (14b)-(30b) that is obtained in the following way: replace a_i by b_i in (14a)-(30a) and, to obtain (14b)-(21b) replace the righthand side by zero. To solve these 34 equations we try to use seven points, or 35 unknowns. We begin by adding to the set the following 12 equations

$a_1= 0$;	(31a)	$b_1= 0$;	(31b)
$\sum_1 L_{21}M_1= M_2^2/2$;	(32)	$\sum_1 L_{21}M_1^2= M_2^3/3$;	(37)
$\sum_1 L_{31}M_1= M_3^2/2$;	(33)	$\sum_1 L_{31}M_1^2= M_3^3/3$;	(38)
$\sum_1 L_{41}M_1= M_4^2/2$;	(34)	$\sum_1 L_{41}M_1^2= M_4^3/3$;	(39)

$$\sum_l L_{5l}M_l = M_5^2/2; \qquad (35)$$

$$\sum_l L_{6l}M_l = M_6^2/2; \qquad (36)$$

$$\sum_l L_{5l}M_l^2 = M_5^3/3; \qquad (40)$$

$$\sum_l L_{6l}M_l^2 = M_6^3/3. \qquad (41)$$

Substitution of these additional equations into the original ones shows that a number of them reduce to identities. Using (32)-(36) and (16a), the lefthand side of (17a) is

$$\sum_i a_i \sum_l L_{il}M_l = \sum_i a_i M_i^2/2 = \frac{1}{6}.$$

Thus (17a) is satisfied.

Similarly, using (37)-(41) and (18a), the lefthand side of (19a) is

$$\sum_i a_i \sum_l L_{il}M_l^2 = \sum_i a_i M_i^3/3 = \frac{1}{12}.$$

Thus (19a) is satisfied.

Again, using (32)-(36) and (18a) we have

$$\sum_i a_i M_i \sum_l L_{il}M_l = \sum_i a_i M_i^3/2 = \frac{1}{8}.$$

Thus (20a) is satisfied.

Using (37)-(41) and (22a) we have

$$\sum_i a_i M_i \sum_l L_{il}M_l^2 = \sum_i a_i M_i^4/3 = \frac{1}{15},$$

showing that (24a) is satisfied.

And

$$\sum_i a_i M_i^2 \sum_l L_{il}M_l = \sum_i a_i M_i^4/2 = \frac{1}{10}.$$

Thus (25a) is satisfied.

Again,

$$\sum_i a_i \left(\sum_l L_{il}M_l \right)^2 = \sum_i a_i M_i^4/4 = \frac{1}{20},$$

showing that (26a) is satisfied.

Using (32)-(36) and (23a) we have

$$\sum_i a_i \sum_l L_{il}M_l \quad \sum_p L_{lp}M_p = \sum_i a_i \sum_l L_{il}M_l^3/2 = \frac{1}{40}.$$

Thus (28a) is satisfied.

Using (32)-(36), (37)-(41) and (18a) we have

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp}M_p = \sum_i a_i \sum_{l=2}^{i-1} L_{il}M_l^2/2 = \sum_i a_i \sum_{l=1}^{i-1} L_{il}M_l^2/2 - \sum_i a_i L_{i1}M_1^2/2 =$$

$$\sum_i a_i M_i^3/6 - M_1^2/2 \sum_i a_i L_{i1} = \frac{1}{24} - M_1^2/2 \sum_i a_i L_{i1}.$$

Thus (18a) may be replaced by $\sum_i a_i L_{i1} = 0$.

Again using (32)-(36), (37)-(41) and (22a) we have

$$\sum_i a_i M_i \sum_l L_{il} \sum_p L_{lp}M_p = \sum_i a_i M_i \sum_{l=2}^{i-1} L_{il}M_l^2/2 = \sum_i a_i M_i \sum_{l=1}^{i-1} L_{il}M_l^2/2 -$$

$$\sum_i a_i M_i L_{i1}M_1^2/2 = \sum_i a_i M_i^4/6 - M_1^2/2 \sum_i a_i M_i L_{i1} = \frac{1}{30} - M_1^2/2 \sum_i a_i M_i L_{i1}.$$

Thus (29a) may be replaced by $\sum_i a_i M_i L_{i1} = 0$.

Finally, using (37)-(41), (23a) we have

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp}M_p^2 = \sum_i a_i \sum_{l=2}^{i-1} L_{il}M_l^3/3 = \sum_i a_i \sum_{l=1}^{i-1} L_{il}M_l^3/3 - \sum_i a_i L_{i1}M_1^3/3$$

$$= \frac{1}{60},$$

showing that (27a) is satisfied.

All results obtained above remain unchanged if we replace a_i by b_i.

The set of equations now is (31)-(41) and

$$\sum_i a_i = 1; \qquad (42a) \qquad \sum_i b_i = 0; \qquad (42b)$$

$$\sum_i a_i M_i = \frac{1}{2}; \qquad (43a) \qquad \sum_i b_i M_i = 0; \qquad (43b)$$

$$\sum_i a_i M_i^2 = \frac{1}{3}; \qquad (44a) \qquad \sum_i b_i M_i^2 = 0 \qquad (44b)$$

$$\sum_i a_i M_i^3 = \frac{1}{4}; \quad (45a) \qquad \sum_i b_i M_i^3 = 0; \quad (45b)$$

$$\sum_i a_i L_{il} = 0; \quad (46a) \qquad \sum_i b_i L_{il} = 0; \quad (46b)$$

$$\sum_i a_i M_i^4 = \frac{1}{5}; \quad (47a) \qquad \sum_i b_i M_i^4 = \frac{1}{5}; \quad (47b)$$

$$\sum_i a_i \sum_l L_{il} M_l^3 = \frac{1}{20}; \quad (48a) \qquad \sum_i b_i \sum_l L_{il} M_l^3 = \frac{1}{20}; \quad (48b)$$

$$\sum_i a_i M_i L_{il} = 0; \quad (49a) \qquad \sum_i b_i M_i L_{il} = 0; \quad (49b)$$

$$\sum_i a_i \sum_l L_{il} \sum_p L_{lp} \sum_q L_{pq} M_q = \frac{1}{120}; \quad (50a) \qquad \sum_i b_i \sum_l L_{il} \sum_p L_{lp} \sum_q L_{pq} M_q = \frac{1}{120} \quad (50b)$$

being 30 equations with 35 unknowns. To simplify this set we introduce

$$\sum_i a_i L_{i2} = a_2(1-M_2); \quad (51) \qquad \sum_i b_i L_{i2} = -b_2(1-M_2)/4; \quad (54)$$

$$\sum_i a_i L_{i3} = a_3(1-M_3); \quad (52) \qquad \sum_i b_i L_{i3} = -b_3(1-M_3)/4; \quad (55)$$

$$\sum_i a_i L_{i4} = a_4(1-M_4); \quad (53) \qquad \sum_i b_i L_{i4} = -b_4(1-M_4)/4; \quad (56)$$

$$M_5 = 1; \quad (57)$$

$$M_6 = 1; \quad (58)$$

$$L_{65} = 0. \quad (59)$$

(59) means that the computation of k_6 is independent of k_5.

For the lefthand side of (48a) we find

$$\sum_{i=2}^{6} a_i \sum_{l=1}^{i-1} L_{il} M_l^3 = \sum_{l=1}^{5} M_l^3 \sum_{i=l+1}^{6} a_i L_{il} = \sum_{l=2}^{4} M_l^3 \sum_{i=l+1}^{6} a_i L_{il} =$$

$$\sum_l M_l^3 a_l (1-M_l) = \frac{1}{20}.$$

Thus (48a) is satisfied. The same holds for (48b).

For the lefthand side of (50a) we find

$$\sum_{i=4}^{6} a_i \sum_{l=3}^{i-1} L_{il} \sum_{p=2}^{l-1} L_{lp} \sum_{q=1}^{p-1} L_{pq} M_q = \frac{1}{2} \sum_{i=4}^{6} a_i \sum_{l=3}^{i-1} L_{il} \sum_{p=2}^{l-2} L_{lp} M_p^2 =$$

$$\frac{1}{6} \sum_{i=4}^{6} a_i \sum_{l=3}^{i-1} L_{il} M_l^2 - \frac{1}{2} M_1^2 \sum_{i=4}^{6} a_i \sum_{l=3}^{i-1} L_{il} L_{l1} =$$

$$\frac{1}{6} \sum_{l=3}^{5} M_l^3 \sum_{i=l+1}^{6} a_i L_{il} - \frac{1}{2} M_1^2 \sum_{l=3}^{5} L_{l1} \sum_{i=l+1}^{6} a_i L_{il} =$$

$$\frac{1}{6} \sum_{l=2}^{6} M_l^3 a_l (1-M_l) - \frac{1}{6} M_2^3 a_2 (1-M_2) - \frac{1}{2} M_1^2 \sum_{l=3}^{5} L_{l1} a_l (1-M_l) =$$

$$\frac{1}{120} - \frac{1}{2} M_1^2 \sum_{l=2}^{6} L_{l1} a_l (1-M_l) = \frac{1}{120}.$$

Thus (50a) and (50b) are satisfied.

We shall see that four more equations are dependent. Writing

$$\Sigma L_{5i} M_i = pM_5^2; \quad \Sigma L_{6i} M_i = qM_6^2,$$

we determine p and q. We have

$$\sum_{i=2}^{6} a_i \sum_{l=1}^{i-1} L_{il} M_l = \sum_{l=1}^{5} M_l \sum_{i=l+1}^{6} a_i L_{il} = \sum_{l=1}^{4} M_l a_l (1-M_l) + M_5 a_5 L_{65} = \frac{1}{6}.$$

On the other hand,

$$\sum_{i=2}^{6} a_i \sum_{l=1}^{i-1} L_{il} M_l = \frac{1}{2} \sum_{i=2}^{4} a_i M_i^2 + pa_5 + qa_6 = \frac{1}{6} + (p-\frac{1}{2}) a_5 + (q-\frac{1}{2}) a_6.$$

Furthermore,

$$\sum_{i=2}^{6} b_i \sum_{l=1}^{i-1} L_{il} M_l = 0 \quad \text{and} \quad \sum_{i=2}^{6} b_i \sum_{l=1}^{i-1} L_{il} M_l = \frac{1}{2} \sum_{i=2}^{4} b_i M_i^2 + pb_5 + qb_6 =$$

$$(p-\frac{1}{2}) b_5 + (q-\frac{1}{2}) b_6.$$

Thus, $p=\frac{1}{2}$ and $q=\frac{1}{2}$ and equations (35) and (36) are satisfied.

In the same way we find that (40) and (41) may be omitted.

So finally we have 31 equations with 35 unknowns:

$$\sum_i a_i = 1; \quad (57a) \qquad \sum_i b_i = 0; \quad (57b)$$

$$\sum_i a_i M_i = \frac{1}{2}; \quad (58a) \qquad \sum_i b_i M_i = 0; \quad (58b)$$

$\sum_i a_i M_i^2 = \frac{1}{3};$ (59a) $\sum_i b_i M_i^2 = 0;$ (59b)

$\sum_i a_i M_i^3 = \frac{1}{4};$ (60a) $\sum_i b_i M_i^3 = 0;$ (60b)

$\sum_i a_i M_i^4 = \frac{1}{5};$ (61a) $\sum_i b_i M_i^4 = \frac{1}{5};$ (61b)

$\sum_i a_i L_{i1} = 0;$ (62) $\sum_i b_i L_{i1} = 0;$ (66)

$\sum_i a_i L_{i2} = a_2(1-M_2);$ (63) $\sum_i b_i L_{i2} = -b_2(1-M_2)/4;$ (67)

$\sum_i a_i L_{i3} = a_3(1-M_3);$ (64) $\sum_i b_i L_{i3} = -b_3(1-M_3)/4;$ (68)

$\sum_i a_i L_{i4} = a_4(1-M_4);$ (65) $\sum_i b_i L_{i4} = -b_4(1-M_4)/4;$ (69)

$\sum_i a_i L_{i1} M_i = 0;$ (70) $\sum_i b_i L_{i1} M_1 = 0;$ (71)

$\sum_l L_{21} M_1 = M_2^2/2;$ (72) $\sum_l L_{21} M_1^2 = M_2^3/3;$ (75)

$\sum_l L_{31} M_1 = M_3^2/2;$ (73) $\sum_l L_{31} M_1^2 = M_3^3/3;$ (76)

$\sum_l L_{41} M_1 = M_4^2/2;$ (74) $\sum_l L_{41} M_1^2 = M_4^3/3;$ (77)

$a_1 = 0;$ (78a) $b_1 = 0;$ (78b)

$M_5 = 1;$ (79) $M_6 = 1;$ (80)

$L_{65} = 0.$ (81)

To solve these we try $a_5=b_6=0$. If we succeed, then, with $L_{65}=0$, we have found a six point integration formula and a six point formula for $th^5 dy_j$, with the advantage that in order to find whether the step is acceptable, one has to compute six points, and only if the step is accepted, need one compute another point.

From (72) and (75) we have

$M_1 = \frac{3}{2} M_2;\ L_{21} = \frac{3}{4} M_2.$

From (73) and (76) we find

$$L_{31}=\frac{3}{4}\, M_3^2(3M_2-2M_3)/M_2^2;\ L_{32}=-M_3^2(M_2-M_3)/M_2^2,$$

and from (74) and (77)

$$L_{41}=(\tfrac{3}{4}M_4^2(3M_2-2M_4)-\tfrac{9}{2}\, M_3(M_2-M_3)L_{43})/M_2^2;$$

$$L_{42}=(-M_4^2(M_2-M_4)+3M_3(M_1-M_3)L_{43})/M_2^2.$$

Substituting the expressions found for M_1, L_{21}, L_{31}, L_{32}, L_{41} and L_{42} into (70) and (71), and using (62) and (66) we find

$$M_2^3(1-M_2)a_2+M_3^2(3M_2-2M_3)(1-M_3)a_3+(M_4^2(3M_2-2M_4)-6L_{43}M_3(M_2-M_3))(1-M_4)a_4=0$$

and a similar equation with a_i replaced by b_i. This equation, together with (58a), (59a), (60a) and (61a) forms five linear equations for a_2, a_3, a_4 and a_5, thus giving the condition

$$L_{43}= -(M_2-M_4)(M_3-M_4)(2-5M_2)M_4/(M_2-M_3)/(10M_2M_3-5(M_2+M_3)+3)/M_3/2.$$

From the corresponding b-equations it follows that

$$L_{43}= -M_4(M_2-M_4)(M_3-M_4)/M_3/(M_2-M_3)/3,$$

or $M_2=2M_3(1+4M_3)$.

We still have two free parameters. Our aim has been to use them to make at least the a_i positive (This is the reason why we did not use, as integration formula, the one given by Kutta, as corrected by Nyström[9]; this formula has negative weights). This condition can be satisfied and even leaves some freedom of choice. That is why, using the Electrologica X1 computer, we varied the two free parameters to find a formula which, in addition, has only positive L_{il}. In this we did not succeed. The most acceptable choice was

$$M_3=\frac{1}{2};\ M_4=\frac{4}{5},$$

resulting in

$$M_1=\frac{2}{9};\ M_2=\frac{1}{3};$$

$a_0=\frac{35}{336}$; $a_1=0$; $a_2=\frac{162}{336}$; $a_3=0$; $a_4=\frac{125}{336}$; $a_5=\frac{14}{336}$; $a_6=0$;
$b_0=\frac{21}{14}$; $b_1=0$; $b_2=-\frac{162}{14}$; $b_3=\frac{224}{14}$; $b_4=-\frac{125}{14}$; $b_5=0$; $b_6=\frac{42}{14}$;
$L_{20}=\frac{1}{12}$; $L_{21}=\frac{3}{12}$; $L_{30}=\frac{1}{8}$; $L_{31}=0$; $L_{32}=\frac{3}{8}$; $L_{40}=\frac{53}{125}$; $L_{41}=-\frac{135}{125}$; $L_{42}=\frac{126}{125}$;
$L_{43}=\frac{56}{125}$; $L_{50}=-\frac{63}{28}$; $L_{51}=\frac{189}{28}$; $L_{52}=-\frac{36}{28}$; $L_{53}=-\frac{112}{28}$; $L_{54}=\frac{50}{28}$; $L_{60}=\frac{133}{168}$;
$L_{61}=-\frac{378}{168}$; $L_{62}=\frac{276}{168}$; $L_{63}=\frac{112}{168}$; $L_{64}=\frac{25}{168}$; $L_{65}=0$.

Our fifth order formula is

$$k_{0j}=hf_j(Y_0,\ldots,Y_n);$$
$$k_{1j}=hf_j(Y_0+\tfrac{2}{9}k_{00},\ldots,Y_n+\tfrac{2}{9}k_{0n});$$
$$k_{2j}=hf_j(Y_0+(k_{00}+3k_{10})/12,\ldots,Y_n+(k_{0n}+3k_{1n})/12);$$
$$k_{3j}=hf_j(Y_0+(k_{00}+3k_{20})/8,\ldots,Y_n+(k_{0n}+3k_{2n})/8);$$
$$k_{4j}=hf_j(Y_0+(53k_{00}-135k_{10}+126k_{20}+56k_{30})/125,\ldots,$$
$$Y_n+(53k_{0n}-135k_{1n}+126k_{2n}+56k_{3n})/125);$$
$$k_{5j}=hf_j(Y_0+(-63k_{00}+189k_{10}-36k_{20}-112k_{30}+50k_{40})/28,\ldots,$$
$$Y_n+(-63k_{0n}+189k_{1n}-36k_{2n}-112k_{3n}+50k_{4n})/28);$$
$$k_{6j}=hf_j(Y_0+(133k_{00}-378k_{10}+276k_{20}+112k_{30}+25k_{40})/168,\ldots,$$
$$Y_n+(133k_{0n}-378k_{1n}+276k_{2n}+112k_{3n}+25k_{4n})/168);$$
$$y_j^*(X+h)=Y_j+(35k_{0j}+162k_{2j}+125k_{4j}+14k_{5j})/336;$$
$$th^5dy_j=(21k_{0j}-162k_{2j}+224k_{3j}-125k_{4j}+42k_{6j})/14.$$

In order to determine whether an integration step is acceptable, one has to compute th^5dy_j which necessitates the computation of k_0, k_1, k_2, k_3, k_4 and k_6 but not that of k_5. Only in case of acceptance has k_5 to be calculated.

The weights a_0, a_2, a_3, a_4 and a_5 in the integration formula are all positive. The influence of rounding errors on dy_j is therefore minimized. Only some of the L_{il} are negative. This does not matter very much since rounding errors influence the result only in terms which are multiplied by h before entering in the rest of the computation.

CHAPTER 3

Second order equations with first derivatives

Suppose we have the set of differential equations

$$d^2y_j/dx^2=f_j(x,y_1,\dots,y_n,dy_1/dx,\dots,dy_n/dx);\ y_j(X)=Y_j;$$

$$dy_j(X)/dx=Y'_j;\ j=1(1)n. \tag{1}$$

For the sake of simplicity we prefer

$$d^2y_j/dx^2=f_j(y_0,y_1,\dots,y_n,dy_0/dx,dy_1/dx,\dots,dy_n/dx);\ y_j(X)=Y_j;$$

$$dy_j(X)/dx=Y'_j;\ j=0(1)n, \tag{2}$$

which is equivalent to (1) if we put $x=y_0$, whence

$f_0(y_0,\dots,y_1,dy_0/dx,\dots,dy_n/dx)=0$, $Y_0=X$ and $Y'_0=1$.

The Runge-Kutta scheme of order r with m points is

$$k_{ij}=hf_j(Y_0+h(M_iY'_0+\sum_{l=0}^{i-1}K_{il}k_{l0}),\dots,Y_n+h(M_iY'_n+\sum_{l=0}^{i-1}K_{il}k_{ln},$$

$$Y'_0+\sum_{l=0}^{i-1}N_{il}k_{l0},\dots,Y'_n+\sum_{l=0}^{i-1}N_{il}k_{ln});\ j=0(1)n;\ i=0(1)m-1; \tag{3}$$

$$y_j^*(X+h)=Y_j+hY'_j+h\sum_{i=0}^{m-1}A_ik_{ij} \tag{4}$$

$$y_j'^*(X+h)=Y'_j+\sum_{i=0}^{m-1}a_ik_{ij}; \tag{5}$$

$$th^rdy_j=h\sum_{i=0}^{m-1}B_ik_{ij}; \tag{6}$$

$$th^rdy'_j=\sum_{i=0}^{m-1}b_ik_{ij}. \tag{7}$$

We now must develop $y_j^*(X+h)$, $y_j'^*(X+h)$, th^rdy_j and $th^rdy'_j$ in power series in h.

To this end, we introduce some notations:

i) we do not write the arguments of f_j and of its derivatives, if they are

$(Y_0,\ldots,Y_n,Y'_0,\ldots,Y'_n)$;

ii) $f_{jklm'\ldots}$ will mean $f_j(y_0,\ldots,y_n,dy_0/dx,\ldots,dy_n/dx)$, differentiated with respect to $y_k, y_l, dy_m/dx$, etc, in

$(Y_0,\ldots,Y_n,Y'_0,\ldots,Y'_n)$;

iii) K_{il} and N_{il} are defined for $l=0(1)i-1$, and are otherwise zero;

iv) the range of the following summation variables, if not redefined explicitly, is understood to be

$l=1(1)i-1$; $p=1(1)l-1$; $q=1(1)p-1$; $s=0(1)n$; $t=0(1)n$; $u=0(1)n$;

v) we introduce the differential operators

$$D=\sum_s \left(y'_s \frac{\partial}{\partial y_s}+f_s \frac{\partial}{\partial y'_s}\right) \text{ and } E=\sum_s f_s \frac{\partial}{\partial y_s}. \tag{7}$$

To simplify the results a great deal, we introduce the equations:

$$\sum_l K_{il}=\frac{1}{2} M_i^2; \tag{8}$$

$$\sum_l N_{il}=M_i. \tag{9}$$

We retain terms of order five and find, for $i \geq 4$

$$\begin{aligned}
k_{ij}=&hf_j+h^2M_iDf_j+h^3\left(\tfrac{1}{2}M_i^2(D^2+E)f_j+\left(\sum_l N_{il}M_l\right)\sum_s Df_sf_{js'}\right)+\\
&h^4\left(\left(\sum_l K_{il}M_l\right)\sum_s Df_sf_{js}+\tfrac{1}{2}\left(\sum_l N_{il}M_l^2\right)\sum_s (D^2+E)f_sf_{js'}+\right.\\
&\left(\sum_l N_{il}\sum_p N_{lp}M_p\right)\sum_s\sum_t Df_sf_{ts'}f_{jt'}+\tfrac{1}{6}M_i^3(D^3+3DE)f_j+M_i\left(\sum_l N_{il}M_l\right)\\
&\left.\sum_s Df_sDf_{js'}\right)+h^5\left(\tfrac{1}{2}\left(\sum_l K_{il}M_l^2\right)\sum_s (D^2+E)f_sf_{js}+\left(\sum_l K_{il}\sum_p N_{lp}M_p\right)\right.\\
&\sum_s\sum_t Df_sf_{ts'}f_{jt}+\left(\sum_l N_{il}\sum_p K_{lp}M_p\right)\sum_s\sum_t Df_sf_{ts}f_{jt'}+
\end{aligned}$$

$\frac{1}{2}(\sum_l N_{il}\sum_p N_{lp}M_p^2)\sum_s\sum_t (D^2+E)f_s f_{ts}, f_{jt}, +\frac{1}{6}(\sum_l N_{il}M_l^3)$

$\sum_s (D^3+3DE)f_s f_{js}, +(\sum_l N_{il}M_l \sum_p N_{lp}M_p)\sum_s\sum_t Df_s Df_{ts}, f_{jt}, +$

$(\sum_l N_{il}\sum_p N_{lp}\sum_q N_{pq}M_q)\sum_s\sum_t\sum_u Df_s f_{ts}, f_{ut}, f_{ju}, +M_i(\sum_l K_{il}M_l)$

$\sum_s Df_s Df_{js}+\frac{1}{2}M_i(\sum_l N_{il}M_l^2)\sum_s (D^2+E)f_s Df_{js}, +M_i(\sum_l N_{il}\sum_p N_{lp}M_p)$

$\sum_s\sum_t Df_s f_{ts}, Df_{jt}, +\frac{1}{24}M_i^4(D^4+6D^2E+3E^2)f_j+\frac{1}{2}M_i^2(\sum_l N_{il}M_l)$

$\sum_s Df_s(D^2+E)f_{js}, +\frac{1}{2}(\sum_l N_{il}M_l)^2\sum_s\sum_t Df_s Df_t f_{js't'})$. (10)

For i <4, the expression for k_{ij} reduces to

$k_{0j}= hf_j;$ (11)

$k_{1j}= hf_j+h^2M_1Df_j+\frac{1}{2}h^3M_1^2(D^2+E)f_j+\frac{1}{6}h^4M_1^3(D^3+3DE)f_j+\frac{1}{24}h^5M_1^4$
$(D^4+6D^2E+3E^2)f_j;$ (12)

$k_{2j}= hf_j+h^2M_2Df_j+h^3(\frac{1}{2}M_2^2(D^2+E)f_j+N_{21}M_1\sum_s Df_s f_{js},)+$

$h^4(K_{21}M_1\sum_s Df_s f_{js}+\frac{1}{2}N_{21}M_1^2\sum_s (D^2+E)f_s f_{js}, +\frac{1}{6}M_2^3(D^3+3DE)f_j+$

$M_2N_{21}M_1\sum_s Df_s Df_{js},)+h^5(\frac{1}{2}K_{21}M_1^2\sum_s (D^2+E)f_s f_{js}+\frac{1}{6}N_{21}M_1^3\sum_s (D^3+3DE)$

$f_s f_{js}, +M_2K_{21}M_1\sum_s Df_s Df_{js}+\frac{1}{2}M_2N_{21}M_1^2\sum_s (D^2+E)f_s Df_{js}, +$

$\frac{1}{24}M_2^4(D^4+6D^2E+3E^2)f_j+\frac{1}{2}M_2^2N_{21}M_1\sum_s Df_s(D^2+E)f_{js}, +\frac{1}{2}N_{21}^2M_1^2$

$\sum_s\sum_t Df_s Df_t f_{js't'});$ (13)

$k_{3j}= hf_j+h^2M_3Df_j+h^3(\frac{1}{2}M_3^2(D^2+E)f_j+(\sum_l N_{31}M_l)\sum_s Df_s f_{js},)+$

$h^4((\sum_l K_{31}M_l)\sum_s Df_s f_{js}+\frac{1}{2}(\sum_l N_{31}M_l^2)\sum_s (D^2+E)f_s f_{js}, +$

$$N_{32}N_{21}M_1 \sum_s \sum_t Df_s f_{ts'} f_{jt'} + \frac{1}{6}M_3^3(D^3+3DE)f_j + M_3(\sum_l N_{31}M_1)$$

$$\sum_s Df_s Df_{js'}) + h^5(\frac{1}{2}(\sum_l K_{31}M_1^2) \sum_s (D^2+E)f_s f_{js} + K_{32}N_{21}M_1 \sum_s \sum_t$$

$$Df_s f_{ts'} f_{jt} + N_{32}K_{21}M_1 \sum_s \sum_t Df_s f_{ts} f_{jt'} + \frac{1}{2}N_{32}N_{21}M_1^2 \sum_s \sum_t (D^2+E)$$

$$f_s f_{ts'} f_{jt'} + \frac{1}{6}(\sum_l N_{31}M_1^3) \sum_s (D^3+3DE)f_s f_{js'} + N_{32}M_2N_{21}M_1 \sum_s \sum_t$$

$$Df_s Df_{ts'} f_{jt'} + M_3(\sum_l K_{31}M_1) \sum_s Df_s Df_{js} + \frac{1}{2}M_3(\sum_l N_{31}M_1^2) \sum_s (D^2+E)$$

$$f_s Df_{js'} + M_3N_{32}N_{21}M_1 \sum_s \sum_t Df_s f_{ts'} Df_{jt'} + \frac{1}{24}M_3^4(D^4+6D^2E+3E^2)f_j +$$

$$\frac{1}{2}M_3^2(\sum_l N_{31}M_1) \sum_s Df_s(D^2+E)f_{js'} + \frac{1}{2}(\sum_l N_{31}M_1)^2 \sum_s \sum_t Df_s Df_t f_{js't'}). \tag{14}$$

Furthermore, we have the Taylor series

$$y_j(X+h) = Y_j + hY'_j + \frac{1}{2}h^2 f_j + \frac{1}{6}h^3 Df_j + \frac{1}{24}h^4((D^2+E)f_j + \sum_s Df_s f_{js'}) +$$

$$\frac{1}{120}h^5((D^3+3DE)f_j + \sum_s (D^2+E)f_s f_{js'} + 3\sum_s Df_s Df_{js'} + \sum_s Df_s f_{js} +$$

$$\sum_s \sum_t Df_s f_{ts'} f_{jt'}); \tag{15}$$

$$y'_j(X+h) = Y'_j + hf_j + \frac{1}{2}h^2 Df_j + \frac{1}{6}h^3((D^2+E)f_j + \sum_s Df_s f_{js'}) +$$

$$\frac{1}{24}h^4((D^3+3DE)f_j + \sum_s (D^2+E)f_s f_{js'} + 3\sum_s Df_s Df_{js'} + \sum_s Df_s f_{js} +$$

$$\sum_s \sum_t Df_s f_{ts'} f_{jt'}) + \frac{1}{120}h^5((D^4+6D^2E+3E^2)f_j +$$

$$\sum_s (D^3+3DE)f_s f_{js'} + 4\sum_s (D^2+E)f_s Df_{js'} + \sum_s (D^2+E)f_s f_{js} +$$

$$\sum_s \sum_t (D^2+E)f_s f_{ts'} f_{jt'} + \sum_s \sum_t Df_s f_{ts'} f_{jt} + \sum_s \sum_t Df_s f_{ts} f_{jt'} +$$

$$3\sum_s \sum_t Df_s Df_{ts'} f_{jt'} + \sum_s \sum_t \sum_u Df_s f_{ts'} f_{ut'} f_{ju'} + 4\sum_s Df_s Df_{js} +$$

$$4\sum_s \sum_t Df_s f_{ts'} Df_{jt'} + 6\sum_s Df_s Ef_{js'} + 3\sum_s \sum_t Df_s Df_t f_{js't'} +$$

$$6\sum_s Df_s D^2 f_{js'}). \tag{16}$$

Equating coefficients of powers of h in

$y_j(X+h)$ and $y_j^{*}(X+h) = Y_j + hY_j' + h\sum_{i=0}^{m-1} A_i k_{ij}$,

and

$y_j'(X+h)$ and $y_j'^{*}(X+h) = Y_j' + \sum_{i=0}^{m-1} a_i k_{ij}$,

we find

$$h^2: \quad \sum_i A_i = \frac{1}{2}; \tag{17a}$$

$$h^3: \quad \sum_i A_i M_i = \frac{1}{6}; \tag{18a}$$

$$h^4: \quad \sum_i A_i M_i^2 = \frac{1}{12}; \tag{19a}$$

$$\sum_i A_i \sum_l N_{il} M_l = \frac{1}{24}; \tag{20a}$$

$$h^5: \quad \sum_i A_i \sum_l K_{il} M_l = \frac{1}{120}; \tag{21a}$$

$$\sum_i A_i \sum_l N_{il} M_l^2 = \frac{1}{60}; \tag{22a}$$

$$\sum_i A_i M_i^3 = \frac{1}{20}; \tag{23a}$$

$$\sum_i A_i M_i \sum_l N_{il} M_l = \frac{1}{40}; \tag{24a}$$

$$\sum_i A_i \sum_l N_{il} \sum_p N_{lp} M_p = \frac{1}{120}; \tag{25a}$$

$$h: \quad \sum_i a_i = 1; \tag{26a}$$

$$h^2: \quad \sum_i a_i M_i = \frac{1}{2}; \tag{27a}$$

h^3: $\sum_i a_i M_i^2 =$ $\frac{1}{3}$; (28a)

$\sum_i a_i \sum_l N_{il} M_l =$ $\frac{1}{6}$; (29a)

h^4: $\sum_i a_i M_i^3 =$ $\frac{1}{4}$; (30a)

$\sum_i a_i \sum_l N_{il} M_l^2 =$ $\frac{1}{12}$; (31a)

$\sum_i a_i M_i \sum_l N_{il} M_l =$ $\frac{1}{8}$; (32a)

$\sum_i a_i \sum_l N_{il} \sum_p N_{lp} M_p =$ $\frac{1}{24}$; (33a)

$\sum_i a_i \sum_l K_{il} M_l =$ $\frac{1}{24}$; (34a)

h^5: $\sum_i a_i M_i^4 =$ $\frac{1}{5}$; (35a)

$\sum_i a_i \sum_l N_{il} M_l^3 =$ $\frac{1}{20}$; (36a)

$\sum_i a_i M_i \sum_l N_{il} M_l^2 =$ $\frac{1}{15}$; (37a)

$\sum_i a_i M_i^2 \sum_l N_{il} M_l =$ $\frac{1}{10}$; (38a)

$\sum_i a_i (\sum_l N_{il} M_l)^2 =$ $\frac{1}{20}$; (39a)

$\sum_i a_i \sum_l N_{il} \sum_p N_{lp} M_p^2 =$ $\frac{1}{60}$; (40a)

$\sum_i a_i \sum_l N_{il} M_l \sum_p N_{lp} M_p =$ $\frac{1}{40}$; (41a)

$\sum_i a_i M_i \sum_l N_{il} \sum_p N_{lp} M_p =$ $\frac{1}{30}$; (42a)

$\sum_i a_i \sum_l N_{il} \sum_p N_{lp} \sum_q N_{pq} M_q =$ $\frac{1}{120}$; (43a)

$\sum_i a_i \sum_l K_{il} M_l^2 =$ $\frac{1}{60}$; (44a)

$$\sum_i a_i M_i \sum_l K_{il} M_l = \frac{1}{30}; \tag{45a}$$

$$\sum_i a_i \sum_l K_{il} \sum_p N_{lp} M_p = \frac{1}{120}; \tag{46a}$$

$$\sum_i a_i \sum_l N_{il} \sum_p K_{lp} M_p = \frac{1}{120}. \tag{47a}$$

For B_i and b_i, the weights in the expressions for $th^r dy_j$ and $th^r dy'_j$, we find a similar set of equations, the lefthand side being the same with B_i replacing a_i, and the righthand side being zero in the equations resulting from terms of h^t, $t < r$ and being identical to the above righthand side if $t=r$. The equations of this set we denote by (17b)-(47b).

Next we solve the equations for various values of r.

<u>r=1</u>

$k_{0j} = hf_j(Y_0, \ldots, Y_n, Y'_0, \ldots, Y'_n)$;

$y^*_j(X+h) = Y_j + hY'_j$;

$y'^*_j(X+h) = Y'_j + k_{0j}$;

$th^1 dy_j = hY'_j$;

$th^1 dy'_j = k_{0j}$.

<u>r=2</u>

The equations are

$a_0 + a_1 = 1$;	$b_0 + b_1 = 0$
$a_1 M_1 = \frac{1}{2}$;	$b_1 M_1 = \frac{1}{2}$;
$A_0 + A_1 = \frac{1}{2}$;	$B_0 + B_1 = \frac{1}{2}$.

with $M_1 = 1$ we find

$k_{0j} = hf_j(Y_0, \ldots, Y_n, Y'_0, \ldots, Y'_n)$;

$k_{1j} = hf_j(Y_0 + hY'_0, \ldots, Y_n + hY'_n, Y'_0 + k_{00}, \ldots, Y'_n + k_{0n})$;

$y^*_j(X+h) = Y_j + h(Y'_j + k_{0j}/2)$;

$y_j'^{*}(X+h)= Y_j'+(k_{0j}+k_{1j})/2;$

$th^2dy_j= h\ k_{0j}/2;$

$th^2dy_j'= (-k_{0j}+k_{1j})/2.$

r=3

We try four points. The equations are

$A_0+A_1+A_2+A_3= \frac{1}{2};$

$A_1M_1+A_2M_2+A_3M_3= \frac{1}{6};$

$a_0+a_1+a_2+a_3= 1;$

$a_1M_1+a_2M_2+a_3M_3= \frac{1}{2};$

$a_1M_1^2+a_2M_2^2+a_3M_3^2= \frac{1}{3};$

$a_2N_{21}M_1+a_3(N_{31}M_1+N_{32}M_2)= \frac{1}{6};$

$B_0+B_1+B_2+B_3= 0;$

$B_1M_1+B_2M_2+B_3M_3= \frac{1}{6};$

$b_0+b_1+b_2+b_3= 0;$

$b_1M_1+b_2M_2+b_3M_3= 0;$

$b_1M_1^2+b_2M_2^2+b_3M_3^2= \frac{1}{3};$

$b_2N_{21}M_1+b_3(N_{31}M_1+N_{32}M_2)= \frac{1}{6}.$

If we put $A_i= a_i(1-M_i)$ then the equations for A_i automatically are fulfilled if we use the values obtained by solving the equations for a_i. The same holds for the B-equations if we put $B_i=-b_i(1-M_i)/2$. As solution for a_i and b_i we use the solution given in chapter 2 (r=3).

Thus we have

$k_{0j}= hf_j(Y_0,\ldots,Y_n,Y_0',\ldots,Y_n');$

$k_{1j}= hf_j(Y_0+h(6Y_0'+k_{00})/18,\ldots,Y_n+h(6Y_n'+k_{0n})/18,Y_0'+k_{00}/3,\ldots,$
$Y_n'+k_{0n}/3);$

$k_{2j}= hf_j(Y_0+h(6Y_0'+2k_{00})/9,\ldots,Y_n+h(6Y_n'+2k_{0n})/9,Y_0'+2k_{10}/3,\ldots,$
$Y_n'+2k_{1n}/3);$

$y_j^{*}(X+h)= Y_j+h(Y_j'+(k_{0j}+k_{2j})/4);$

$y_j'^{*}(X+h)= Y_j'+(k_{0j}+3k_{2j})/4;$

$k_{3j}= hf_j(y_0^{*}(X+h),\ldots,y_n^{*}(X+h),y_0'^{*}(X+h),\ldots,y_n'^{*}(X+h));$

$th^3dy_j= h(-k_{0j}+k_{2j})/4;$

$th^3 dy'_j = (k_{0j} - 3k_{2j} + 2k_{3j})/2,$

where, in case of acceptance, only three evaluations of f_j are necessary, and no extra work is needed to compute $th^3 dy_j$ and $th^3 dy'_j$.

r=4

We use five points. Writing this time

$A_i = a_i(1-M_i)$ and $B_i = -b_i(1-M_i)/3,$

we again need not consider the equations for A_i and B_i. Of the remaining set (26a,b)-(34a,b) all equations but (34a,b) are solved by the solution of the first order equations, r=4. (chapter 2). Insertion of this solution in (34a,b) gives

$4K_{21} + 2(K_{31} + K_{32}) = 1;$

$24(K_{21} + K_{31} + K_{32}) - 64K_{41} - 128K_{42} - 96K_{43} = 1.$

A solution is

$K_{21} = K_{30} = K_{31} = K_{42} = K_{43} = 0;\ K_{10} = \frac{1}{8};\ K_{20} = \frac{1}{8};\ K_{32} = \frac{1}{2};\ K_{40} = \frac{7}{64};\ K_{41} = \frac{11}{64}.$

The formula is

$k_{0j} = hf_j(Y_0, \ldots, Y_n, Y'_0, \ldots, Y'_n);$

$k_{1j} = hf_j(Y_0 + h(4Y'_0 + k_{00})/8, \ldots, Y_n + h(4Y'_n + k_{0n})/8, Y'_0 + k_{00}/2, \ldots, Y'_n + k_{0n}/2);$

$k_{2j} = hf_j(Y_0 + h(4Y'_0 + k_{00})/8, \ldots, Y_n + h(4Y'_n + k_{0n})/8, Y'_0 + k_{10}/2, \ldots, Y'_n + k_{1n}/2);$

$k_{3j} = hf_j(Y_0 + h(2Y'_0 + k_{20})/2, \ldots, Y_n + h(2Y'_n + k_{2n})/2, Y'_0 + k_{20}, \ldots, Y'_n + k_{2n});$

$k_{4j} = hf_j(Y_0 + h(48Y'_0 + 7k_{00} + 11k_{10})/64, \ldots, Y_n + h(48Y'_n + 7k_{0n} + 11k_{1n})/64,$
$\quad Y'_0 + (5k_{00} + 7k_{10} + 13k_{20} - k_{30})/32, \ldots, Y'_n + (5k_{0n} + 7k_{1n} + 13k_{2n} - k_{3n})/32);$

$y^*_j(X+h) = Y_j + h(Y'_j + (k_{0j} + k_{1j} + k_{2j})/6);$

$y'^*_j(X+h) = Y'_j + (k_{0j} + 2(k_{1j} + k_{2j}) + k_{3j})/6;$

$th^4 dy_j = h(2k_{0j} - 3(k_{1j} + k_{2j}) + 4k_{3j})/9;$

$th^4 dy'_j = \frac{2}{3}(-k_{0j} + 3(k_{1j} + k_{2j} + k_{3j}) - 8k_{4j}).$

<u>r=5</u>

The set of equations to be solved is (71a,b)-(47a,b). If we put

$A_i = a_i(1-M_i)$ and $B_i = -b_i(1-M_i)/4$,

we need not consider the equations for A_i and B_i.

We use the seven point solution for first order equations with r=5. The remaining equations are (34,a,b),(44a,b)-(47a,b). After substitution of the values for a_i, b_i, M_i, N_{il} they reduce to ten linear equations in the fifteen unknowns K_{il}; $i=2(1)6$; $l=1(1)i-1$. We tried to make use of the free parameters to find K_{il} that are positive. We nearly succeeded. The best solution we found was

$K_{21}=0$; $K_{31}=0$; $K_{32}=\frac{1}{16}$; $K_{41}=0$; $K_{42}=0$; $K_{43}=\frac{28}{125}$; $K_{51}=0$; $K_{52}=\frac{36}{56}$;
$K_{53}=0$; $K_{54}=-\frac{15}{56}$; $K_{61}=0$; $K_{62}=\frac{92}{336}$; $K_{63}=0$; $K_{64}=\frac{55}{336}$; $K_{65}=0$.

We chose $K_{65}=0$ in order that, with $N_{65}=0$, the computation of k_{6j} is independent of k_{5j}. The fifth order formula is

$$k_{0j} = hf_j(Y_0,\ldots,Y_n,Y_0',\ldots,Y_n');$$

$$k_{1j} = hf_j(Y_0+h(18Y_0'+2k_{00})/81,\ldots,Y_n+h(18Y_n'+2k_{0n})/81,Y_0'+\tfrac{2}{9}k_{00},\ldots,$$
$$Y_n'+\tfrac{2}{9}k_{0n});$$

$$k_{2j} = hf_j(Y_0+h(6Y_0'+k_{00})/18,\ldots,Y_n+h(6Y_n'+k_{0n})/18,Y_0'+(k_{00}+3k_{10})/12,\ldots,$$
$$Y_n'+(k_{0n}+3k_{1n})/12);$$

$$k_{3j} = hf_j(Y_0+h(8Y_0'+k_{00}+k_{20})/16,\ldots,Y_n+h(8Y_n'+k_{0n}+k_{2n})/16,Y_0'+$$
$$(k_{00}+3k_{20})/8,\ldots,Y_n'+(k_{0n}+3k_{2n})/8);$$

$$k_{4j} = hf_j(Y_0+h(100Y_0'+12k_{00}+28k_{30})/125,\ldots,Y_n+h(100Y_n'+12k_{0n}+28k_{3n})/125,$$
$$Y_0'+(53k_{00}-135k_{10}+126k_{20}+56k_{30})/125,\ldots,$$
$$Y_n'+(53k_{0n}-135k_{1n}+126k_{2n}+56k_{3n})/125);$$

$$k_{5j} = hf_j(Y_0+h(56Y_0'+7k_{00}+36k_{20}-15k_{40})/56,\ldots,$$
$$Y_n+h(56Y_n'+7k_{0n}+36k_{2n}-15k_{4n})/56,$$
$$Y_0'+(-63k_{00}+189k_{10}-36k_{20}-112k_{30}+50k_{40})/28,\ldots,$$

$$Y_n'+(-63k_{0n}+189k_{1n}-36k_{2n}-112k_{3n}+50k_{4n})/28);$$

$$k_{6j}= hf_j(Y_0+h(336Y_0'+21k_{00}+92k_{20}+55k_{40})/336,\ldots,$$

$$h(336Y_n'+21k_{0n}+92k_{2n}+55k_{4n})/336,$$

$$Y_0'+(133k_{00}-378k_{10}+276k_{20}+112k_{30}+25k_{40})/168,\ldots,$$

$$Y_n'+(133k_{0n}-378k_{1n}+276k_{2n}+112k_{3n}+25k_{4n})/168);$$

$$y_j^*(X+h)= Y_j+h(Y_j'+(35k_{0j}+108k_{2j}+25k_{4j})/336;$$

$$y_j'^*(X+h)= Y_j'+(35k_{0j}+162k_{2j}+125k_{4j}+14k_{5j})/336;$$

$$th^5dy_j= h(-21k_{0j}+108k_{2j}-112k_{3j}+25k_{4j})/56;$$

$$th^5dy_j'= (21k_{0j}-162k_{2j}+224k_{3j}-125k_{4j}+42k_{6j})/14.$$

CHAPTER 4

Second order equations without first derivatives

For the set of equations

$$d^2y_j/dx^2 = f_j(y_0,\ldots,y_n);\ y_j(X)=Y_j;\ dy_j(X)/dx=Y_j';\ j=0(1)n, \tag{1}$$

we have, for the m-point formula of order r

$$k_{ij} = hf_j(Y_0+h(M_iY_0'+\sum_{l=0}^{i-1}K_{il}k_{l0}),\ldots,Y_n+h(M_iY_n'+\sum_{l=0}^{i-1}K_{il}k_{ln})); \tag{2}$$

$$y_j^*(X+h) = Y_j+h(Y_j'+\sum_{i=0}^{m-1}A_ik_{ij}); \tag{3}$$

$$y_j'^*(X+h) = Y_j'+\sum_{i=0}^{m-1}a_ik_{ij}; \tag{4}$$

$$th^5dy_j = h\sum_{i=0}^{m-1}B_ik_{ij}; \tag{5}$$

$$th^5dy_j' = \sum_{i=0}^{m-1}b_ik_{ij}. \tag{6}$$

The set of equations is the same as that in the previous chapter, with $N_{il}=0$.

$$h^2:\quad \sum_i A_i = \frac{1}{2}; \tag{7a}$$

$$h^3:\quad \sum_i A_iM_i = \frac{1}{6}; \tag{8a}$$

$$h^4:\quad \sum_i A_iM_i^2 = \frac{1}{12}; \tag{9a}$$

$$h^5:\quad \sum_i A_i\sum_l K_{il}M_l = \frac{1}{120}; \tag{10a}$$

$$\sum_i A_iM_i^3 = \frac{1}{20}; \tag{11a}$$

$$h:\quad \sum_i a_i = 1; \tag{12a}$$

h^2: $\sum_i a_i M_i =$ $\frac{1}{2}$; (13a)

h^3: $\sum_i a_i M_i^2 =$ $\frac{1}{3}$; (14a)

h^4: $\sum_i a_i M_i^3 =$ $\frac{1}{4}$; (16a)

$\sum_i a_i \sum_l K_{il} M_l =$ $\frac{1}{24}$; (17a)

h^5: $\sum_i a_i M_i^4 =$ $\frac{1}{5}$; (18a)

$\sum_i a_i \sum_l K_{il} M_l^2 =$ $\frac{1}{60}$; (19a)

$\sum_i a_i M_i \sum_l K_{il} M_l =$ $\frac{1}{30}$; (20a)

For B_i and b_i a similar set of equations exists; the lefthand side is obtained from the above lefthand side, by replacing A_i by B_i and a_i by b_i; the righthand side is zero for equations resulting from terms of h^t, $t < r$ and identical to the above if $t=r$. The equations of this set we shall denote by (7b)-(20b).

Next we solve the equations for various values of r.

<u>r=1</u>

$k_{0j} = hf_j(Y_0, \ldots, Y_n)$;

$y_j^*(X+h) = Y_j + hY_j'$;

$y_j'^*(X+h) = Y_j' + k_{0j}$;

$th^1 dy_j = hY_j'$;

$th^1 dy_j' = k_{0j}$.

<u>r=2</u>

$k_{0j} = hf_j(Y_0, \ldots, Y_n)$;

$k_{1j} = hf_j(Y_0 + hY_0', \ldots, Y_n + hY_n')$;

$y_j^*(X+h) = Y_j + h(Y_j' + k_{0j}/2)$;

$y_j'^{*}(X+h) = Y_j' + (k_{0j}+k_{1j})/2;$

$th^2 dy_j = hk_{0j}/2;$

$th^2 dy_j' = (-k_{0j}+k_{1j})/2.$

r=3

The equations are

$$a_0+a_1+a_2 = 1; \qquad b_0+b_1+b_2 = 0;$$
$$a_1M_1+a_2M_2 = \frac{1}{2}; \qquad b_1M_1+b_2M_2 = 0;$$
$$a_1M_1^2+a_2M_2^2 = \frac{1}{3}; \qquad b_1M_1^2+b_2M_2^2 = \frac{1}{3},$$

with $A_i = a_i(1-M_i)$ and $B_i = -b_i(1-M_i)/2;\ i=0(1)2$.

We try to find a two point integration formula; thus, $a_2=0$. Furthermore we want to use as third point the starting point of the next integration step, thus $M_2 = 1$ and $K_{20} = A_0$ and $K_{21} = A_1$.

We find

$$a_0 = \frac{1}{4};\ a_1 = \frac{3}{4};\ a_2 = 0;\ M_1 = \frac{2}{3};$$
$$A_0 = \frac{1}{4};\ A_1 = \frac{1}{4};\ A_2 = 0;\ M_2 = 1;$$
$$b_0 = \frac{1}{2};\ b_1 = -\frac{3}{2};\ b_2 = 1;$$
$$B_0 = \frac{1}{2};\ B_1 = -\frac{1}{2};\ B_2 = 0;$$
$$K_{10} = \frac{2}{9};\ K_{20} = \frac{1}{4};\ K_{21} = \frac{1}{4},$$

using $\sum_{1} K_{i1} = \frac{1}{2}M_i^2$.

Thus the formula is

$k_{0j} = hf_j(Y_0,\ldots,Y_n);$

$k_{1j} = hf_j(Y_0+h(6Y_0'+2k_{00})/9,\ldots,Y_n+h(6Y_n'+2k_{0n})/9);$

$y_j^{*}(X+h) = Y_j+h(Y_j'+(k_{0j}+k_{1j})/4);$

$y_j'^{*}(X+h) = Y_j'+(k_{0j}+3k_{1j})/4;$

$k_{2j} = hf_j(y_0^{*}(X+h),\ldots,y_n^{*}(X+h));$

$th^3 dy_j = h(k_{0j}-k_{1j})/2;$

$th^3 dy_j' = (k_{0j} - 3k_{1j} + 2k_{2j})/2.$

Therefore, in case the steps are accepted, the computation of two function values gives third order accuracy; i.e. the work to be done is the same as for r=2.

r=4

The solution for r=4 is easily obtained from the corresponding one derived for second order equations with first derivative occuring, by observing that the expressions, obtained for k_{1j} and k_{2j}, only differ in the actual parameters substituted for the formal parameters dy_i/dx in the functions $f_j(y_0, \ldots, y_n, dy_0/dx, \ldots, dy_n/dx)$.
Hence,

$k_{0j} = hf_j(Y_0, \ldots, Y_n);$

$k_{1j} = hf_j(Y_0 + h(4Y_0' + k_{00})/8, \ldots, Y_n + h(4Y_n' + k_{0n})/8);$

$k_{2j} = hf_j(Y_0 + h(2Y_0' + k_{10})/2, \ldots, Y_n + h(2Y_n' + k_{1n})/2);$

$k_{3j} = hf_j(Y_0 + h(48Y_0' + 7k_{00} + 11k_{10})/64, \ldots, Y_n + h(48Y_n' + 7k_{0n} + 11k_{1n})/64);$

$y_j^*(X+h) = Y_j + h(Y_j' + (k_{0j} + 2k_{1j})/6);$

$y_j'^*(X+h) = Y_j' + (k_{0j} + 4k_{1j} + k_{2j})/6;$

$th^4 dy_j = h(2k_{0j} - 6k_{1j} + 4k_{3j})/9;$

$th^5 dy_j' = \frac{2}{3}(-k_{0j} + 6k_{1j} + 3k_{2j} - 8k_{3j}).$

r=5

To find a formula for this case we firstly construct a four point integration formula with positive coefficients and weights.
The equations are

$\sum_l K_{il} = M_i^2/2;\ A_i = a_i(1-M_i);\ i = 1(1)3;\ A_0 = a_0;$

$\sum_i a_i = 1; \qquad \sum_i a_i M_i^4 = \frac{1}{5};$

$\sum_i a_i M_i = \frac{1}{2}; \qquad \sum_i a_i \sum_l K_{il} M_l = \frac{1}{24};$

$\sum_i a_i M_i^2 = \frac{1}{3}$; $\sum_i a_i \sum_l K_{il} M_l^2 = \frac{1}{60}$;

$\sum_i a_i M_i^3 = \frac{1}{4}$; $\sum_i a_i M_i \sum_l K_{il} M_l = \frac{1}{30}$.

These are 15 equations in 17 unknowns. We solve them, choosing $M_3=1$. We find

$M_1 = (3-5M_2)/(5-10M_2)$; $a_0 = (10M_1M_2-1)/M_1/M_2$;

$a_1 = (-5+10M_2)/60/M_1/(1-M_1)/(M_2-M_1)$;

$a_2 = (5+10M_2)/60/M_2/(1-M_2)/(M_2-M_1)$;

$a_3 = (15-20(M_1+M_2)+30M_1M_2)/60/(1-M_2)/(1-M_1)$;

$K_{21} = \frac{1}{120}/M_1/(1-M_2)/a_2$; $K_{32} = (2-5M_1)/120/M_2/(M_2-M_1)/a_3$;

$K_{31} = ((4-5M_2)(M_2-M_1)-(2-5M_1)(1-M_2))/120/M_1/(1-M_2)/(M_2-M_1)/a_3$.

Letting $M_1 \leq M_2$, then we have the conditions:

$a_0 \geq 0$ or $M_1M_2 \geq \frac{1}{10}$;

$a_1 \geq 0$ or $M_2 \geq \frac{1}{2}$;

$a_2 \geq 0$ or $M_1 \leq \frac{1}{2}$;

$a_3 \geq 0$ or $(6-\mathrm{sqrt}(6))/10 \leq M_2 \leq (6+\mathrm{sqrt}(6))/10$;

$K_{32} \geq 0$ or $M_1 \leq \frac{2}{5}$;

$K_{31} \geq 0$ or $\frac{7}{10} \leq M_2 \leq (5+\mathrm{sqrt}(5))/10$;

$K_{20} \geq 0$ or $(5+\mathrm{sqrt}(5))/10 \leq M_2 \leq \frac{3}{4}$.

Thus $M_2 = (5+\mathrm{sqrt}(5))/10$.

The integration formula is

p= sqrt(5);

$k_{0j} = hf_j(Y_0, \ldots, Y_n)$;

$k_{1j} = hf_j(Y_0+h((10-2p)Y_0'+(3-p)k_{00})/20, \ldots, Y_n+h((10-2p)Y_n'+(3-p)k_{0n})/20)$;

$k_{2j}= hf_j(Y_0+h((10+2p)Y_0'+(3+p)k_{10})/20,\ldots,Y_n+h((10+2p)Y_n'+(3+p)$
$k_{1n})/20)$;

$k_{3j}= hf_j(Y_0+h(4Y_0'+(-1+p)k_{00}+(3-p)k_{20})/4,\ldots,Y_n+h(4Y_n'+(-1+p)k_{0n}+$
$(3-p)k_{2n})/4)$;

$y_j^*(X+h)= Y_j+h(Y_j'+(2k_{0j}+(5+p)k_{1j}+(5-p)k_{2j})/24)$;

$y_j'^*(X+h)= Y_j'+(k_{0j}+5(k_{1j}+k_{2j})+k_{3j})/12$.

To find the expressions for th^5dy_j and th^5dy_j', we must solve:

$$\sum_i b_i= 0; \qquad \sum_i b_iM_i^4= \frac{1}{5};$$

$$\sum_i b_iM_i= 0; \qquad \sum_i b_i\sum_l K_{il}M_l= 0;$$

$$\sum_i b_iM_i^2= 0; \qquad \sum_i b_i\sum_l K_{il}M_l^2= \frac{1}{60};$$

$$\sum_i b_iM_i^3= 0; \qquad \sum_i b_iM_i\sum_l K_{il}M_l= \frac{1}{30}.$$

We tried in vain, using one extra point, to solve this system. So, we had to use two extra points in this case. As second point we use the starting point of the next integration step. The formulas for th^5dy_j and th^5dy_j' are

$k_{4j}= hf_j(Y_0+h(192Y_0'+18k_{00}+p(3p+7)k_{10}+p(3p-7)k_{20})/384,\ldots,$
$Y_n+h(192Y_n'+18k_{0n}+p(3p+7)k_{1n}+p(3p-7)k_{2n})/384)$;

$k_{5j}= hf_j(y_0^*(X+h),\ldots,y_n^*(X+h))$;

$th^5dy_j= h(-2k_{0j}+(5+p)k_{1j}+(5-p)k_{2j}-8k_{4j})/4$;

$th^5dy_j'= 2k_{0j}-10k_{1j}-10k_{2j}-2k_{3j}+16k_{4j}+4k_{5j}$.

CHAPTER 5

Summary of formulas

We now give a summary of the formulas obtained in the preceding chapters. In the derivations, we assumed that, in the righthand side of the differential equations, the independent variable did not occur. Here, however, we give the formulas with the independent variable in the righthand side.

$$dy_j/dx = f_j(x, y_1, \ldots, y_n); \quad y_j(X) = Y_j; \quad j = 1(1)n.$$

<u>r=2</u>

$k_{0j} = hf_j(X, Y_1, \ldots, Y_n);$

$k_{1j} = hf_j(X+h, Y_1+k_{01}, \ldots, Y_n+k_{0n});$

$y_j^{*}(X+h) = Y_j + (k_{0j}+k_{1j})/2;$

$th^2 dy_j = (-k_{0j}+k_{1j})/2.$

<u>r=3</u>

$k_{0j} = hf_j(X, Y_1, \ldots, Y_n);$

$k_{1j} = hf_j(X+h/3, Y_1+k_{01}/3, \ldots, Y_n+k_{0n}/3);$

$k_{2j} = hf_j(X+\frac{2}{3}h, Y_1+\frac{2}{3}k_{01}, \ldots, Y_n+\frac{2}{3}k_{0n});$

$y_j^{*}(X+h) = Y_j + (k_{0j}+3k_{2j})/4;$

$k_{3j} = hf_j(X+h, y_1^{*}(X+h), \ldots, y_n^{*}(X+h));$

$th^3 dy_j = (k_{0j}-3k_{2j}+2k_{3j})/2.$

r=4

$k_{0j} = hf_j(X, Y_1, \ldots, Y_n);$

$k_{1j} = hf_j(X+h/2, Y_1+k_{01}/2, \ldots, Y_n+k_{0n}/2);$

$k_{2j} = hf_j(X+h/2, Y_1+k_{11}/2, \ldots, Y_n+k_{1n}/2);$

$k_{3j} = hf_j(X+h, Y_1+k_{21}, \ldots, Y_n+k_{2n});$

$y_j^*(X+h) = Y_j+(k_{0j}+2(k_{1j}+k_{2j})+k_{3j})/6;$

$k_{4j} = hf_j(X+\frac{3}{4}h, Y_0+(5k_{01}+7k_{11}+13k_{21}-k_{31})/32, \ldots,$

$Y_n+(5k_{0n}+7k_{1n}+13k_{2n}-k_{3n})/32);$

$th^4dy_j = \frac{2}{3}(-k_{0j}+3(k_{1j}+k_{2j}+k_{3j})-8k_{4j}).$

r=5

$k_{0j} = hf_j(X, Y_1, \ldots, Y_n);$

$k_{1j} = hf_j(X+\frac{2}{9}h, Y_1+\frac{2}{9}k_{01}, \ldots, Y_n+\frac{2}{9}k_{0n});$

$k_{2j} = hf_j(X+h/3, Y_1+(k_{01}+3k_{11})/12, \ldots, Y_n+(k_{0n}+3k_{1n})/12);$

$k_{3j} = hf_j(X+h/2, Y_1+(k_{01}+3k_{21})/8, \ldots, Y_n+(k_{0n}+3k_{2n})/8);$

$k_{4j} = hf_j(X+\frac{4}{5}h, Y_1+(53k_{01}-135k_{11}+126k_{21}+56k_{31})/125, \ldots,$

$Y_n+(53k_{0n}-135k_{1n}+126k_{2n}+56k_{3n})/125);$

$k_{5j} = hf_j(X+h, Y_1+(-63k_{01}+189k_{11}-36k_{21}-112k_{31}+50k_{41})/28, \ldots,$

$Y_n+(-63k_{0n}+189k_{1n}-36k_{2n}-112k_{34}+50k_{4n})/28);$

$y_j^*(X+h) = Y_j+(35k_{0j}+162k_{2j}+125k_{4j}+14k_{5j})/336;$

$k_{6j} = hf_j(X+h, Y_1+(133k_{01}-378k_{11}+276k_{21}+112k_{31}+25k_{41})/168, \ldots,$

$Y_n+(133k_{0n}-378k_{1n}+276k_{2n}+112k_{3n}+25k_{4n})/168);$

$th^5dy_j = (21k_{0j}-162k_{2j}+224k_{3j}-125k_{4j}+42k_{6j})/14.$

$$d^2y_j/dx^2 = f_j(x,y_1,\ldots,y_n,dy_1/dx,\ldots,dy_n/dx);\ y_j(X) = Y_j;$$
$$dy_j(X)/dx = Y_j';\ j = 1(1)n.$$

r=2

$k_{0j} = hf_j(X,Y_1,\ldots,Y_n,Y_1',\ldots,Y_n');$

$k_{1j} = hf_j(X+h,Y_1+hY_1',\ldots,Y_n+hY_n',Y_1'+k_{01},\ldots,Y_n'+k_{0n});$

$y_j^*(X+h) = Y_j+h(Y_j'+k_{0j}/2);$

$y_j'^*(X+h) = Y_j'+(k_{0j}+k_{1j})/2;$

$th^2dy_j = hk_{0j}/2;$

$th^2dy_j' = (-k_{0j}+k_{1j})/2.$

r=3

$k_{0j} = hf_j(X,Y_1,\ldots,Y_n,Y_1',\ldots,Y_n');$

$k_{1j} = hf_j(X+h/3,Y_1+h(6Y_1'+k_{01})/18,\ldots,Y_n+h(6Y_n'+k_{0n})/18,$
$Y_1'+k_{01}/3,\ldots,Y_n'+k_{0n}/3);$

$k_{2j} = hf_j(X+\frac{2}{3}h,Y_1+h(6Y_1'+2k_{01})/9,\ldots,Y_n+h(6Y_n'+2k_{0n})/9,$
$Y_1'+\frac{2}{3}k_{11},\ldots,Y_n'+\frac{2}{3}k_{1n});$

$y_j^*(X+h) = Y_j+h(Y_j'+(k_{0j}+k_{2j})/4);$

$y_j'^*(X+h) = Y_j'+(k_{0j}+3k_{2j})/4;$

$k_{3j} = hf_j(X+h,y_1^*(X+h),\ldots,y_n^*(X+h),y_1'^*(X+h),\ldots,y_n'^*(X+h));$

$th^3dy_j = h(-k_{0j}+k_{2j})/4;$

$th^3dy_j' = (k_{0j}-3k_{2j}+2k_{3j})/2.$

<u>r=4</u>

$k_{0j}= hf_j(X,Y_1,\ldots,Y_n,Y_1',\ldots,Y_n');$

$k_{1j}= hf_j(X+h/2,Y_1+h(4Y_1'+k_{01})/8,\ldots,Y_n+h(4Y_n'+k_{0n})/8,$
$\quad Y_1'+k_{01}/2,\ldots,Y_n'+k_{0n}/2);$

$k_{2j}= hf_j(X+h/2,Y_1+h(4Y_1'+k_{01})/8,\ldots,Y_n+h(4Y_n'+k_{0n})/8,$
$\quad Y_1'+k_{11}/2,\ldots,Y_n'+k_{1n}/2);$

$k_{3j}= hf_j(X+h,Y_1+h(2Y_1'+k_{21})/2,\ldots,Y_n+h(2Y_n'+k_{2n})/2,Y_1'+k_{21},\ldots,Y_n'+k_{2n});$

$y_j^*(X+h)= Y_j+h(Y_j'+(k_{0j}+k_{1j}+k_{2j})/6);$

$y_j'^*(X+h)= Y_j'+(k_{0j}+2(k_{1j}+k_{2j})+k_{3j})/6;$

$k_{4j}= hf_j(X+\frac{2}{3}h,Y_1'+h(48Y_1'+7k_{01}+11k_{11})/64,\ldots,Y_n'+h(48Y_n'+7k_{0n}+11k_{1n})/64,$
$\quad Y_1'+(5k_{01}+7k_{11}+13k_{21}-k_{31})/32,\ldots,Y_n'+(5k_{0n}+7k_{1n}+13k_{2n}-k_{3n})$
$\quad /32);$

$th^4dy_j= h(2k_{0j}-3(k_{1j}+k_{2j})+4k_{3j})/9;$

$th^4dy_j'= \frac{2}{3}(-k_{0j}+3(k_{1j}+k_{2j}+k_{3j})-8k_{4j}).$

<u>r=5</u>

$k_{0j}= hf_j(X,Y_1,\ldots,Y_n,Y_1',\ldots,Y_n');$

$k_{1j}= hf_j(X+\frac{2}{9}h,Y_1+h(18Y_1'+2k_{01})/81,\ldots,Y_n+h(18Y_n'+2k_{0n})/81,Y_1'+\frac{2}{9}k_{01},\ldots,$
$\quad Y_n'+\frac{2}{9}k_{0n});$

$k_{2j}= hf_j(X+h/3,Y_1+h(6Y_1'+k_{01})/18,\ldots,Y_n+h(6Y_n'+k_{0n})/18,Y_1'+$
$\quad (k_{01}+3k_{11})/12,\ldots,Y_n'+(k_{0n}+3k_{1n})/12);$

$k_{3j}= hf_j(X+h)2,Y_1+h(8Y_1'+k_{01}+k_{21})/16,\ldots,Y_n+h(8Y_n'+k_{0n}+k_{2n})/16,$
$\quad Y_1'+(k_{01}+3k_{21})/8,\ldots,Y_n'+(k_{0n}+3k_{2n})/8);$

$k_{4j}= hf_j(X+\frac{4}{5}h,Y_1+h(100Y_1'+12k_{01}+28k_{31})/125,\ldots,Y_n+$
$\quad h(100Y_n'+12k_{0n}+28k_{3n})/125,Y_1'+(53k_{01}-135k_{11}+126k_{21}+56k_{31})$
$\quad /125,\ldots,Y_n'+(53k_{0n}-135k_{1n}+126k_{2n}+56k_{3n})/125);$

$$k_{5j} = hf_j(X+h, Y_1+h(56Y_1'+7k_{01}+36k_{21}-15k_{41})/56, \ldots, Y_n$$
$$+h(56Y_n'+7k_{0n}+36k_{2n}-15k_{4n})/56,$$
$$Y_1'+(-63k_{01}+189k_{11}-36k_{21}-112k_{31}+50k_{41})/28, \ldots,$$
$$Y_n'+(-63k_{0n}+189k_{1n}-36k_{2n}-112k_{3n}+50k_{4n})/28);$$
$$y_j^*(x+h) = Y_j+h(Y_j'+(35k_{0j}+108k_{2j}+25k_{4j})/336);$$
$$y_j'^*(X+h) = Y_j'+(35k_{0j}+162k_{2j}+125k_{4j}+14k_{5j})/336;$$
$$k_{6j} = hf_j(X+h, Y_1+h(336Y_1'+21k_{01}+92k_{21}+55k_{41}), \ldots, Y_n+$$
$$h(336Y_n'+21k_{0n}+92k_{2n}+55k_{4n}), Y_1'+(133k_{01}-378k_{11}+276k_{21}+112k_{31}+$$
$$25k_{41})/168, \ldots, Y_n'+(133k_{0n}-378k_{1n}+276k_{2n}+112k_{3n}+25k_{4n})/168);$$
$$th^5dy_j = h(-21k_{0j}+108k_{2j}-112k_{3j}+25k_{4j})/56;$$
$$th^5dy_j' = (21k_{0j}-162k_{2j}+224k_{3j}-125k_{4j}+42k_{6j})/14.$$

$d^2y_j/dx^2 = f_j(x, y_1, ..., y_n)$; $y_j(X) = Y_j$; $dy_j(X)/dx = Y'_j$; $j = 1(1)n$;

r=2

$k_{0j} = hf_j(X, Y_1, ..., Y_n)$;

$k_{1j} = hf_j(X+h, Y_1+hY'_1, ..., Y_n+hY'_n)$;

$y^*_j(X+h) = Y_j+h(Y'_j+k_{0j}/2)$;

$y'^*_j(X+h) = Y'_j+(k_{0j}+k_{1j})/2$;

$th^2dy_j = hk_{0j}/2$;

$th^2dy'_j = (-k_{0j}+k_{1j})/2$.

r=3

$k_{0j} = hf_j(X, Y_1, ..., Y_n)$;

$k_{1j} = hf_j(X+\frac{2}{3}h, Y_1+h(6Y'_1+2k_{01})/9, ..., Y_n+h(6Y'_n+2k_{0n})/9)$;

$y^*_j(X+h) = Y_j+h(Y'_j+(k_{0j}+k_{1j})/4)$;

$y'^*_j(X+h) = Y'_j+(k_{0j}+3k_{1j})/4$;

$k_{2j} = hf_j(X+h, y^*_1(X+h), ..., y^*_n(X+h))$;

$th^3dy_j = h(k_{0j}-k_{1j})/2$;

$th^3dy'_j = (k_{0j}-3k_{1j}+2k_{2j})/2$.

r=4

$k_{0j} = hf_j(X, Y_1, ..., Y_n)$;

$k_{1j} = hf_j(X+h/2, Y_1+h(4Y'_1+k_{01})/8, ..., Y_n+h(4Y'_n+k_{0n})/8$;

$k_{2j} = hf_j(X+h, Y_1+h(2Y'_1+k_{11})/2, ..., Y_n+h(2Y'_n+k_{1n})/2)$;

$y^*_j(X+h) = Y_j+h(Y'_j+(k_{0j}+2k_{1j})/6)$;

$y'^*_j(X+h) = Y'_j+(k_{0j}+4k_{1j}+k_{2j})/6$;

$k_{3j}= hf_j(X+\frac{3}{4}h, Y_1+h(48Y_1'+7k_{01}+11k_{11})/64, \ldots,$

$\quad Y_n+h(48Y_n'+7k_{0n}+11k_{1n})/64);$

$th^4dy_j= h(2k_{0j}-6k_{1j}+4k_{3j})/9;$

$th^4dy_j'= \frac{2}{3}(-k_{0j}+6k_{1j}+3k_{2j}-8k_{3j}).$

<u>r=5</u>

p= sqrt(5);

$k_{0j}= hf_j(X, Y_1, \ldots, Y_n);$

$k_{1j}= hf_j(X+(5-p)h/10, Y_1+h((10-2p)Y_1'+(3-p)k_{01})/20, \ldots,$

$\quad Y_n+h((10-2p)Y_n'+(3-p)k_{0n})/20);$

$k_{2j}= hf_j(X+(5+p)h/10, Y_1+h((10+2p)Y_1'+(3+p)k_{11})/20, \ldots,$

$\quad Y_n+h((10+2p)Y_n'+(3-p)k_{0n})/20);$

$k_{3j}= hf_j(X+h, Y_1+h(4Y_1'+(p-1)k_{01}+(3-p)k_{21})/4, \ldots,$

$\quad Y_n+h(4Y_n'+(p-1)k_{0n}+(3-p)k_{2n})/4);$

$y_j^{**}(X+h)= Y_j+h(Y_j'+(2k_{0j}+(5+p)k_{1j}+(5-p)k_{2j})/24);$

$y_j'^{**}(X+h)= Y_j'+(k_{0j}+5(k_{1j}+k_{2j})+k_{3j})/12;$

$k_{4j}= hf_j(X+h/2, Y_1+h(192Y_1'+18k_{01}+p(3p+7)k_{11}+p(3p-7)k_{21})/384, \ldots,$

$\quad Y_n+h(192Y_n'+18k_{0n}+p(3p+7)k_{1n}+p(3p-7)k_{2n})/384);$

$k_{5j}= hf_j(X+h, y_1^{**}(X+h), \ldots, y_n^{**}(X+h));$

$th^5dy_j= h(-2k_{0j}+(5+p)k_{1j}+(5-p)k_{2j}-8k_{4j})/4;$

$th^5dy_j'= 2k_{0j}-10k_{1j}-10k_{2j}-2k_{3j}+16k_{4j}+4k_{5j}.$

CHAPTER 6

Choice of interval and integration variable

In the preceding chapters we derived a set of integration formulas with explicit expressions for the last term taken into account. The value of this last term we want to use to determine whether a step is acceptable with regard to the tolerances specified and to extrapolate a new step length. As criterion for acceptance of a step we shall use this: a step is acceptable if the absolute value of the last term taken into account is smaller than tol, the absolute value of the product of the step length h and some positive tolerance, which latter is independent of h. If we write

$$\text{discr} = \text{abs}(th^r dy), \tag{1}$$

where r is the order of the formula used, then we would like to use as a step length the solution h of

$$\text{discr} = \text{tol}, \tag{2}$$

where discr is approximately proportional to h^r and tol is proportional to h. A process we can use is the following: compute, for a given step h_1, discr and tol. If discr >tol, then the step h_1 is rejected. However, rejection or not, we compute a new step h_2 using the relation

$$h_2 = h_1(\text{tol}/\text{discr})^{1/(r-1)}, \tag{3}$$

that holds approximately. In case of rejection h_2 is used instead of h_1; in case of acceptance we use h_2 to continue the integration. There are some objections against the method just outlined. In the first place it is very dangerous to use h_2 without introducing a safety margin; otherwise, the number of rejections will be larger than necessary. It is better to use

$$h_2 = .95h_1(tol/discr)^{1/(r-1)}, \tag{4}$$

thus giving a safety margin of 20% in discr.
This is still insufficient, however, to guarantee efficient integration. In case of acceptance we have $abs(h_2) \geq .95abs(h_1)$. Suppose that the differential equation is such that the acceptable step length decreases more than 5% at each step. In this case, every step accepted is followed by a step rejected. To overcome this difficulty, we propose to use a two-point extrapolation formula for the step length. Let the maximal acceptable step length for integrating a certain differential equation, as a function of the independent variable x, be

$$H = H_0 + H_1(x-X) + H_2(x-X)^2 + \dots \ . \tag{5}$$

Assume we did a step h_0, starting at X, followed by a step h_1. Writing

$$p(x) = discr(x)/tol(x) \tag{6}$$

we have, neglecting quadratic and higher terms in (5), approximately

$$p(x_1) = p(X+h_0) = c(h_0/H_0)^{r-1}; \tag{7}$$

$$p(x_2) = p(X+h_0+h_1) = c(h_1/(H_0+H_1h_0))^{r-1}. \tag{8}$$

We now want to find a next step h_2 such that for $x = X+h_0+h_1+h_2$ we have

$$p(x) = c(h_2/(H_0+H_1(h_0+h_1)))^{r-1} \approx 1. \tag{9}$$

Writing $s = -1/(r-1)$ we find

$$h_2/h_1 = (1+h_1/h_0)p^s(x_2) - p^s(x_1). \tag{10}$$

Introducing a safety margin, we have

$$h_2/h_1 = .95((1+h_1/h_0)p^s(x_2) - p^s(x_1)). \tag{11}$$

The extrapolation of h is now as follows: in starting we use a trial h_1 and use (4) until we accept two steps; in case of acceptance we use (11), otherwise (4). Although this is a useful scheme it has a drawback if we integrate functions that become infinite. Consider

$$dy/dx = 1/\mathrm{sqrt}(1-x), \tag{12}$$

and suppose that we want to integrate, from x=0 to x=1. First, we try $h_1=1$; this is rejected since discr, being a linear combination of values of the righthand side including that for x=1, is infinite. Using (4) we find $h_2=0$. If now, in the integration procedure, there is a minimal step length hmin, below which we do not reduce the interval and which, if rejected, is skipped without integration at all, then h_2 will be made equal to hmin. If hmin is not too large, this step is accepted and a next step is computed, using (4). This h_2 may well be greater than (1-hmin), and therefore we next try h_2=1-hmin, which is, of course, rejected. We again put h_2=hmin and so on. In this way we proceed with steps hmin, rejecting every other step until we are so close to x=1 that further steps have to be skipped. Integrals of this type can still be computed, if we limit $p^s(x)$, by means of an approximation to it. If, e.g., we have r=5, we use

$$p^s(x) = (\mathrm{discr}(x)/\mathrm{tol}(x))^{-\frac{1}{4}} \approx \mathrm{tol}(x)/(\mathrm{tol}(x)+\mathrm{discr}(x))+.5, \tag{13}$$

which yields exact function value and derivative if tol(x)= discr(x), but varies between 0.5 and 1.5 instead of between 0 and ∞. If we replace 0.5 by 0.45 we have our safety margin. The extrapolation scheme for the step length that we actually use is as follows:

defining $mu_i = \mathrm{tol}(x_i)/(\mathrm{tol}(x_i)+\mathrm{discr}(x_i))+.45;$ (14)

then, in starting or after a rejection we use

$$h_2 = mu_1 * h_1; \tag{15}$$

otherwise,

$$h_2 = h_1((h_1/h_0 + 1)mu_1 - mu_0). \tag{16}$$

In the case of a set of differential equations, we compute, for each equation, a value of h_2. The smallest of them all is used as next step.

Another problem which presents itself is that of the choice of the independent variable. If we want to solve dy/dx= f(x,y), where f(x,y) becomes infinite, it is, if one can not skip intervals as described above, impossible to use x as independent variable in that neighbourhood. We employ two schemes to overcome this difficulty.

One of these is the interchange of x and y. Whenever abs(f(x,y)) > 1 we use y as independent variable. In the case of a set of differential equations, we choose from amongst x and y_j a variable such that the derivatives of the other variables with respect to the one chosen are in absolute value < 1.

The second scheme is to introduce the arc length s as independent variable. In this case, we write the system as follows:

$$dx_j/dx_0 = f_j(x_0,\dots,x_n)/f_0(x_0,\dots,x_n);\ j= 1(1)n, \qquad (17)$$

where f_0 and f_j remain finite. The system we solve is

$$dx_j/ds= f_j/\mathrm{sqrt}(\sum_{i=0}^{n} f_i^2);\ j= 0(1)n. \qquad (18)$$

In order to reach a certain end point of integration, x_0= b, we must, of course, use some kind of a zero finding procedure because, if x_j, rather than x_0 is the independent variable in the neighbourhood of x_0= b, then we want to find a zero of x_0-b, where x_0 is a function of x_j. This circumstance gives us, on the other hand the possibility of specifying as the end of the integration not simply a zero of x_0-b but a zero of an arbitrary function b of $x_0, x_1, \dots, x_n$.

In the next chapters we present a set of ALGOL 60 procedures to be used for the integration of differential equations. All of them use fifth order integration formulas and the extrapolation of the step length as described above.

RK1, RK1n, RK2, RK2n, RK3, RK3n are procedures that use, as integration variable, the variable specified; that is to say, they do not interchange variables. They compute a minimal step hmin, and steps of length $|h| \leq$ hmin are skipped; i.e., only the independent variable is increased by sign (h) * hmin; the dependent variables are left unchanged. Thus, the integration will at least proceed until the end point. The numerical consequences of this we explain with the following example. Suppose we want to integrate

$$dy/dx = f(x,y) = 1/sqrt(1-x), \qquad (18)$$

from somewhere to x= 1.
Using our fifth order formula, we find, with

$$x = 1-eps, \qquad (19)$$

approximately

$$discr(x) = \frac{7}{128} h^5 eps^{-4.5}. \qquad (20)$$

For the tolerance per step h we use

$$tol(x) = h \times re \times f(x,y) = h \times re \times eps^{-.5}, \qquad (21)$$

where re is a relative tolerance.
From

$$discr(x) = tol(x), \qquad (22)$$

we find

$$eps = h(\frac{128}{7} re)^{-\frac{1}{4}}. \qquad (23)$$

If, furthermore, we put hmin= re, (24)
we find that the integration stops at

$$x = 1-eps = 1-(\frac{7}{128})^{\frac{1}{4}} re^{\frac{3}{4}}, \qquad (25)$$

giving an error in y of

$$\int_{1-eps}^{1} 1/sqrt(1-x)\,dx = (14re^3)^{\frac{1}{8}}. \qquad (26)$$

We could, in this case, reduce this error by a factor 2 if, instead of skipping, we had kept f constant. In view of the fact that this gives difficulties in starting in a singularity we did not do so.

RK4 and RK4n are procedures that interchange dependent and independent variables. Here, instead of skipping, we integrate keeping f constant; this may be done since the interchanging prevents f from becoming infinite.

RK5n is a procedure using the arc length as integration variable. In this procedure no hmin is used.

CHAPTER 7

ALGOL 60 procedures with fixed integration variable

We now give six ALGOL 60 procedures for the integration of first and second order differential equations. The independent variable is used as integration variable.
The texts of the procedures given in this and the next two chapters have been printed from the same tapes that have been used for testing and for the examples given in chapter 10.

```
procedure RK1(x,a,b,y,ya,fxy,e,d,fi); value b,fi; real x,a,b,y,ya,fxy;
Boolean fi; array e,d;
begin real e1,e2,xl,yl,h,int,hmin,absh,k0,k1,k2,k3,k4,k5,discr,tol,mu,mu1,fh,
  hl; Boolean last,first,reject;
  if fi then begin d[3]:= a; d[4]:= ya end; d[1]:= 0; xl:= d[3]; yl:= d[4]; if
  fi then d[2]:= b - d[3]; absh:= h:= abs(d[2]); if b - xl < 0 then h:= - h;
  int:= abs(b - xl); hmin:= int × e[1] + e[2]; e1:= e[1]/int; e2:= e[2]/int;
  first:= true; if fi then begin last:= true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
  else - hmin; absh:= hmin end; if h ≥ b - xl ≡ h ≥ 0 then begin d[2]:=
  h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  x:= xl; y:= yl; k0:= fxy × h; x:= xl + h × 2/9; y:= yl + k0 × 2/9;
  k1:= fxy × h; x:= xl + h/3; y:= yl + (k0 + k1 × 3)/12; k2:= fxy × h; x:=
  xl + h × .5; y:= yl + (k0 + k2 × 3)/8; k3:= fxy × h; x:= xl + h × .8;
  y:= yl + (k0 × 53 - k1 × 135 + k2 × 126 + k3 × 56)/125; k4:= fxy × h;
  x:= if last then b else xl + h; y:= yl + (k0 × 133 - k1 × 378 + k2 × 276
  + k3 × 112 + k4 × 25)/168; k5:= fxy × h; discr:= abs(k0 × 21 - k2 ×
  162 + k3 × 224 - k4 × 125 + k5 × 42)/14; tol:= abs(k0) × e1 + absh ×
  e2; reject:= discr > tol; mu:= tol/(tol + discr) + .45; if reject then
  begin if absh ≤ hmin then begin d[1]:= d[1] + 1; y:= yl; first:= true;
  goto next end; h:= mu × h; goto test end; if first then begin first:=
  false; hl:= h; h:= mu × h; goto acc end; fh:= mu × h/hl + mu - mu1;
  hl:= h; h:= fh × h;
acc:  mu1:= mu; y:= yl + (- k0 × 63 + k1 × 189 - k2 × 36 - k3 × 112 +
  k4 × 50)/28; k5:= fxy × hl; y:= yl + (k0 × 35 + k2 × 162 + k4 × 125 +
  k5 × 14)/336;
next:  if b ≠ x then begin xl:= x; yl:= y; goto test end; if ¬ last then d[2]
  := h; d[3]:= x; d[4]:= y end RK1;
```

RK1 can be used to integrate the equation dy/dx= f(x,y).
First we explain the actual parameters corresponding to the formal parameters:

x: the independent variable; upon completion of a call of RK1, it is equal to b;

a: the starting value of x;

b: a value parameter, giving the end value of x;

y: the dependent variable;

ya: the value of y at x=a;

fxy: an expression, depending on x and y, giving the value of dy/dx;

e: an array of positive tolerances, consisting of e[1] and e[2]; e[1] is used as a relative tolerance, e[2] as an absolute one; the tolerance in a quantity z is defined, here and in the sequel, as

tolerance(z)= abs(z) * e[1]+e[2]; (1)

d: an array with elements d[1],...,d[4]; upon completion of a call of RK1:
entier (d[1]+.5) is the number of steps skipped;
d[2] is the step length;
d[3] is equal to b;
d[4] is equal to y(b);

fi: a Boolean; if fi then the integration starts at a, with trial step b-a; if ¬fi then the integration is continued with x=d[3], y=d[4], h=d[2] * sign (b-d[3]) as initial conditions; a and ya are ignored.

RK1 integrates dy/dx= fxy to x=b, with if fi then x=a, y(a)=ya; if ¬fi then x=d[3], y(d[3])= d[4].
Upon completion of a call of RK1 we have x=d[3]=b, y=d[4]=y(b).
RK1 uses as its minimal absolute step length
hmin= e[1] * int+e[2],
where int= abs(b-(<u>if</u> fi <u>then</u> a <u>else</u> d[3])).

If a step of length abs(h) $\leq$ hmin is rejected, a step sign(h) * hmin is skipped; a step is rejected if

$th^5 dy > (abs(fxy) * e[1] + e[2]) * abs(h)/int.$

```
procedure RK1n(x,a,b,y,ya,fxyj,j,e,d,fi,n); value b,fi,n; integer j,n;
real x,a,b,fxyj; Boolean fi; array y,ya,e,d;
begin integer jj; real xl,h,hmin,int,hl,absh,fhm,discr,tol,mu,mu1,fh;
   Boolean last,first,reject; array yl,k0,k1,k2,k3,k4,k5[1:n],ee[1:2×n];
   if fi then begin d[3]:= a; for jj:= 1 step 1 until n do d[jj+3]:= ya[jj]
   end; d[1]:= 0; xl:= d[3]; for jj:= 1 step 1 until n do yl[jj]:= d[jj+3];
   if fi then d[2]:= b - d[3]; absh:= h:= abs(d[2]); if b - xl < 0 then
   h:= - h; int:= abs(b - xl); hmin:= int × e[1] + e[2]; for jj:= 2 step 1
   until n do begin hl:= int × e[2×jj-1] + e[2×jj]; if hl < hmin then
   hmin:= hl end; for jj:= 1 step 1 until 2×n do ee[jj]:= e[jj]/int; first:=
   true; if fi then begin last:= true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
   else - hmin; absh:= hmin end; if h ≥ b - xl ≡ h ≥ 0 then begin d[2]
   := h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  x:= xl; for jj:= 1 step 1 until n do y[jj]:= yl[jj]; for j:= 1 step 1
   until n do k0[j]:= fxyj × h; x:= xl + h × 2/9; for jj:= 1 step 1 until
   n do y[jj]:= yl[jj] + k0[jj] × 2/9; for j:= 1 step 1 until n do k1[j]:=
   fxyj × h; x:= xl + h/3; for jj:= 1 step 1 until n do y[jj]:= yl[jj] +
   (k0[jj] + k1[jj] × 3)/12; for j:= 1 step 1 until n do k2[j]:= fxyj × h;
   x:= xl + h × .5; for jj:= 1 step 1 until n do y[jj]:= yl[jj] + (k0[jj] +
   k2[jj] × 3)/8; for j:= 1 step 1 until n do k3[j]:= fxyj × h; x:= xl +
   h × .8; for jj:= 1 step 1 until n do y[jj]:= yl[jj] + (k0[jj] × 53 - k1[jj]
   × 135 + k2[jj] × 126 + k3[jj] × 56)/125; for j:= 1 step 1 until n do
   k4[j]:= fxyj × h; x:= if last then b else xl + h; for jj:= 1 step 1 until
   n do y[jj]:= yl[jj] + (k0[jj] × 133 - k1[jj] × 378 + k2[jj] × 276 + k3[jj]
   × 112 + k4[jj] × 25)/168; for j:= 1 step 1 until n do k5[j]:= fxyj × h;
   reject:= false; fhm:= 0; for jj:= 1 step 1 until n do begin discr:=
   abs(k0[jj] × 21 - k2[jj] × 162 + k3[jj] × 224 - k4[jj] × 125 + k5[jj] ×
   42)/14; tol:= abs(k0[jj]) × ee[2×jj-1] + absh × ee[2×jj]; reject:= discr
   > tol ∨ reject; fh:= discr/tol; if fh > fhm then fhm:= fh end; mu:=
   1/(1 + fhm) + .45; if reject then begin if absh ≤ hmin then begin d[1]
   := d[1] + 1; for jj:= 1 step 1 until n do y[jj]:= yl[jj]; first:= true;
   goto next end; h:= mu × h; goto test end rej; if first then begin first
```

```
:= false; hl:= h; h:= mu × h; goto acc end; fh:= mu × h/hl + mu -
  mul; hl:= h; h:= fh × h;
acc:  mul:= mu; for jj:= 1 step 1 until n do y[jj]:= yl[jj] + (- k0[jj] ×
  63 + k1[jj] × 189 - k2[jj] × 36 - k3[jj] × 112 + k4[jj] × 50)/28; for j
  := 1 step 1 until n do k5[j]:= fxyj × hl; for jj:= 1 step 1 until n do
  y[jj]:= yl[jj] + (k0[jj] × 35 + k2[jj] × 162 + k4[jj] × 125 + k5[jj] × 14)/
  336;
next:  if b ≠ x then begin xl:= x; for jj:= 1 step 1 until n do yl[jj]:=
  y[jj]; goto test end; if ¬ last then d[2]:= h; d[3]:= x; for jj:= 1 step 1
  until n do d[jj+3]:= y[jj] end RK1n;
```

RK1n is used to integrate a set of equations $dy_j/dx = f_j(x, y_1, \ldots y_n)$. The actual parameters corresponding to the formal parameters are:

x,a,b: the same as for RK1;

y: an array with elements y[1],...,y[n], the dependent variables;

ya: an array with elements ya[1],...,ya[n], the starting values of y[j];

fxyj: an expression, depending on x,y[j], j, giving the value of dy_j/dx;

j: a variable of type integer used, in the actual parameter corresponding to fxyj, to denote the number of the equation required;

e: an array with elements e[1],...,e[2*n]; e[2*j-1] is a relative and e[2*j] is an absolute tolerance associated with y[j], cf(1);

d: an array with elements d[1],...,d[n+3]; upon completion of a call of RK1n we have:
entier (d[1]+.5) is the number of steps skipped;
d[2] is the step length;
d[3] is equal to b;
d[4],...,d[n+3] are equal to y[1],...,y[n] for x=b;

fi: a Boolean; if fi then the integration starts at a, with a trial step b-a; if ¬fi then the integration is continued with, as initial conditions, x=d[3], y[j]=d[j+3], h=d[2] * sign(b-d[3]); a and ya are ignored;

n: the number of equations.

RK1n integrates dy[j]/dx=fxyj to x=b, with if fi then x=a, y[j]=ya[j]else x=d[3], y[j]=d[j+3].

Upon completion of a call of RK1n we have x=d[3]=b, y[j]=d[j+3]; i.e. the value of the j^{th} dependent variable for x=b.

RK1n uses as its minimal absolute step length

$$hmin= \min_{1 \leq j \leq n} (e[2*j-1]*int+e[2*j]);$$

where int=abs(b-(<u>if</u> fi <u>then</u> a <u>else</u> d[3])).

If a step of length abs(h) $\leq$ hmin is rejected, a step sign(h)* hmin is skipped; a step is rejected if

$$th^5dy_j > (abs(fxyj)*e[2*j-1]+e[2*j])*abs(h)/int,$$

for any value of j, $1 \leq j \leq n$.

```
procedure RK2(x,a,b,y,ya,z,za,fxyz,e,d,fi); value b,fi; real x,a,b,y,ya,z,za,
fxyz; Boolean fi; array e,d;
begin real e1,e2,e3,e4,xl,yl,zl,h,int,hmin,hl,absh,k0,k1,k2,k3,k4,k5,discry,
   discrz,toly,tolz,mu,mu1,fhy,fhz; Boolean last,first,reject;
   if fi then begin d[3]:= a; d[4]:= ya; d[5]:= za end; d[1]:= 0; xl:= d[3];
   yl:= d[4]; zl:= d[5]; if fi then d[2]:= b - d[3]; absh:= h:= abs(d[2]);
   if b - xl < 0 then h:= - h; int:= abs(b - xl); hmin:= int × e[1] + e[2];
   hl:= int × e[3] + e[4]; if hl < hmin then hmin:= hl; e1:= e[1]/int; e2:=
   e[2]/int; e3:= e[3]/int; e4:= e[4]/int; first:= true; if fi then begin last
   := true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
   else - hmin; absh:= hmin end; if h ≥ b - xl ≡ h ≥ 0 then begin d[2]:=
   h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  x:= xl; y:= yl; z:= zl; k0:= fxyz × h; x:= xl + h × 2/9; y:= yl +
   (zl × 18 + k0 × 2)/81 × h; z:= zl + k0 × 2/9; k1:= fxyz × h; x:= xl +
   h/3; y:= yl + (zl × 6 + k0)/18 × h; z:= zl + (k0 + k1 × 3)/12; k2:=
   fxyz × h; x:= xl + h × .5; y:= yl + (zl × 8 + k0 + k2)/16 × h; z:= zl
   + (k0 + k2 × 3)/8; k3:= fxyz × h; x:= xl + h × .8; y:= yl + (zl × 100
   + k0 × 12 + k3 × 28)/125 × h; z:= zl + (k0 × 53 - k1 × 135 + k2 ×
   126 + k3 × 56)/125; k4:= fxyz × h; x:= if last then b else xl + h; y:=
   yl + (zl × 336 + k0 × 21 + k2 × 92 + k4 × 55)/336 × h; z:= zl + (k0 ×
   133 - k1 × 378 + k2 × 276 + k3 × 112 + k4 × 25)/168; k5:= fxyz × h;
   discry:= abs((- k0 × 21 + k2 × 108 - k3 × 112 + k4 × 25)/56 × h);
   discrz:= abs(k0 × 21 - k2 × 162 + k3 × 224 - k4 × 125 + k5 × 42)/14;
   toly:= absh × (abs(zl) × e1 + e2); tolz:= abs(k0) × e3 + absh × e4;
   reject:= discry > toly ∨ discrz > tolz; fhy:= discry/toly; fhz:= discrz/
   tolz; if fhz > fhy then fhy:= fhz; mu:= 1/(1 + fhy) + .45; if reject then
   begin if absh ≤ hmin then begin d[1]:= d[1] + 1; y:= yl; z:= zl; first:=
   true; goto next end; h:= mu × h; goto test end; if first then begin first
   := false; hl:= h; h:= mu × h; goto acc end; fhy:= mu × h/hl + mu -
   mu1; hl:= h; h:= fhy × h;
acc:  mu1:= mu; y:= yl + (zl × 56 + k0 × 7 + k2 × 36 - k4 × 15)/56 × hl;
   z:= zl + (-k0 × 63 + k1 × 189 - k2 × 36 - k3 × 112 + k4 × 50)/28;
```

```
  k5:= fxyz × hl; y:= yl + (zl × 336 + k0 × 35 + k2 × 108 + k4 × 25)/336
  × hl; z:= zl + (k0 × 35 + k2 × 162 + k4 × 125 + k5 × 14)/336;
next: if b≠x then begin xl:= x; yl:= y; zl:= z; goto test end; if ¬ last
  then d[2]:= h; d[3]:= x; d[4]:= y; d[5]:= z end RK2;
```

RK2 can be used to integrate $d^2y/dx^2=f(x,y,dy/dx)$.

The actual parameters, corresponding to the formal parameters are:

x,a,b,y,ya: the same as in RK1;

z: the derivative dy/dx;

za: the value of dy/dx for x=a;

fxyz: an expression, depending on x,y,z, giving the value of d^2y/dx^2;

e: an array of tolerances consisting of the elements e[1],...,e[4]; e[1] and e[3] are used as relative, e[2] and e[4] as absolute tolerances for y and dy/dx respectively, cf.(1);

d: an array with elements d[1],...,d[5]; upon completion of a call of RK2
entier (d[1]+.5) is the number of steps skipped;
d[2] is the step length;
d[3] is equal to b;
d[4] is equal to y(b);
d[5] is equal to dy/dx for x=b;

fi: a Boolean: if fi then the integration starts at a with a trial step b-a, if ¬fi then the integration is continued with, as initial conditions, x=d[3], y=d[4], z=d[5], h=d[2] sign(b-d[2]); a, ya, za are ignored.

RK2 integrates d^2y/dx^2=fxyz to x=b, with if fi then x=a, y=ya, dy/dx=za else x=d[3], y=d[4], z=d[5].
Upon completion of a call of RK2 we have x=d[3]=b, y=d[4]=y[b], z=d[5], i.e. the value of dy/dx for x=b.

RK2 uses as its minimal absolute step length

$$hmin= \min_{j=1,2} (e[2*j-1] * int+e[2*j]),$$

where int=abs(b-(<u>if</u> fi <u>then</u> a <u>else</u> d[3])).

If a step of length abs(h) ≤ hmin is rejected, a step sign(h)* hmin is skipped; a step is rejected if

th^5dy >(abs(dy/dx)* e[1]+e[2])* abs(h)/int ∨

th^5dy' >(abs(fxyz)* e[3]+e[4])* abs(h)/int.

```
procedure RK2n(x,a,b,y,ya,z,za,fxyzj,j,e,d,fi,n); value b,fi,n; integer j,n;
real x,a,b,fxyzj; Boolean fi; array y,ya,z,za,e,d;
begin integer jj; real xl,h,int,hmin,hl,absh,fhm,discry,discrz,toly,tolz,mu,
   mul,fhy,fhz; Boolean last,first,reject; array yl,zl,k0,k1,k2,k3,k4,k5[1:n],
   ee[1:4×n];
   if fi then begin d[3]:= a; for jj:= 1 step 1 until n do begin d[jj+3]:=
   ya[jj]; d[n+jj+3]:= za[jj] end end; d[1]:= 0; xl:= d[3]; for jj:= 1 step 1
   until n do begin yl[jj]:= d[jj+3]; zl[jj]:= d[n+jj+3] end; if fi then d[2]:=
   b - d[3]; absh:= h:= abs(d[2]); if b - xl < 0 then h:= - h; int:= abs(b
   - xl); hmin:= int × e[1] + e[2]; for jj:= 2 step 1 until 2×n do begin
   hl:= int × e[2×jj-1] + e[2×jj]; if hl < hmin then hmin:= hl end; for jj:=
   1 step 1 until 4×n do ee[jj]:= e[jj]/int; first:= true; if fi then begin
   last:= true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
   else - hmin; absh:= abs(h) end; if h ≥ b - xl = h ≥ 0 then begin d[2]
   := h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  x:= xl; for jj:= 1 step 1 until n do begin y[jj]:= yl[jj]; z[jj]:= zl[jj]
   end; for j:= 1 step 1 until n do k0[j]:= fxyzj × h; x:= xl + h × 2/9;
   for jj:= 1 step 1 until n do begin y[jj]:= yl[jj] + (zl[jj] × 18 + k0[jj] ×
   2)/81 × h; z[jj]:= zl[jj] + k0[jj] × 2/9 end; for j:= 1 step 1 until n do
   k1[j]:= fxyzj × h; x:= xl + h/3; for jj:= 1 step 1 until n do begin y[jj]
   := yl[jj] + (zl[jj] × 6 + k0[jj])/18 × h; z[jj]:= zl[jj] + (k0[jj] + k1[jj] ×
   3)/12 end; for j:= 1 step 1 until n do k2[j]:= fxyzj × h; x:= xl + h ×
   .5; for jj:= 1 step 1 until n do begin y[jj]:= yl[jj] + (zl[jj] × 8 + k0[jj]
   + k2[jj])/16 × h; z[jj]:= zl[jj] + (k0[jj] + k2[jj] × 3)/8 end; for j:= 1
   step 1 until n do k3[j]:= fxyzj × h; x:= xl + h × .8; for jj:= 1 step 1
   until n do begin y[jj]:= yl[jj] + (zl[jj] × 100 + k0[jj] × 12 + k3[jj] × 28)
   /125 × h; z[jj]:= zl[jj] + (k0[jj] × 53 - k1[jj] × 135 + k2[jj] × 126 +
   k3[jj] × 56)/125 end; for j:= 1 step 1 until n do k4[j]:= fxyzj × h; x:=
   if last then b else xl + h; for jj:= 1 step 1 until n do begin y[jj]:=
   yl[jj] + (zl[jj] × 336 + k0[jj] × 21 + k2[jj] × 92 + k4[jj] × 55)/336 × h;
   z[jj]:= zl[jj] + (k0[jj] × 133 - k1[jj] × 378 + k2[jj] × 276 + k3[jj] × 112
   + k4[jj] × 25)/168 end; for j:= 1 step 1 until n do k5[j]:= fxyzj × h;
```

```
  reject:= false; fhm:= 0; for jj:= 1 step 1 until n do begin discry:=
  abs((- k0[jj] × 21 + k2[jj] × 108 - k3[jj] × 112 + k4[jj] × 25)/56 × h);
  discrz:= abs(k0[jj] × 21 - k2[jj] × 162 + k3[jj] × 224 - k4[jj] × 125 +
  k5[jj] × 42)/14; toly:= absh × (abs(zl[jj]) × ee[2×jj-1] + ee[2×jj]); tolz:=
  abs(k0[jj]) × ee[2×(jj+n)-1] + absh × ee[2×(jj+n)]; reject:= discry > toly
  ∨ discrz > tolz ∨ reject; fhy:= discry/toly; fhz:= discrz/tolz; if fhz >
  fhy then fhy:= fhz; if fhy > fhm then fhm:= fhy end; mu:= 1/(1 + fhm)
  + .45; if reject then begin if absh ≤ hmin then begin d[1]:= d[1] + 1;
  for jj:= 1 step 1 until n do begin y[jj]:= yl[jj]; z[jj]:= zl[jj] end; first
  := true; goto next end; h:= mu × h; goto test end; if first then begin
  first:= false; hl:= h; h:= mu × h; goto acc end; fhm:= mu × h/hl + mu
  - mul; hl:= h; h:= fhm × h;
acc: mul:= mu; for jj:= 1 step 1 until n do begin y[jj]:= yl[jj] + (zl[jj] ×
  56 + k0[jj] × 7 + k2[jj] × 36 - k4[jj] × 15)/56 × hl; z[jj]:= zl[jj] +
  (- k0[jj] × 63 + k1[jj] × 189 - k2[jj] × 36 - k3[jj] × 112 + k4[jj] × 50)
  /28 end; for j:= 1 step 1 until n do k5[j]:= fxyzj × hl; for jj:= 1 step
  1 until n do begin y[jj]:= yl[jj] + (zl[jj] × 336 + k0[jj] × 35 + k2[jj] ×
  108 + k4[jj] × 25)/336 × hl; z[jj]:= zl[jj] + (k0[jj] × 35 + k2[jj] × 162
  + k4[jj] × 125 + k5[jj] × 14)/336 end;
next: if b≠x then begin xl:= x; for jj:= 1 step 1 until n do begin yl[jj]:=
  y[jj]; zl[jj]:= z[jj] end; goto test end; if ¬ last then d[2]:= h; d[3]:= x;
  for jj:= 1 step 1 until n do begin d[jj+3]:= y[jj]; d[n+jj+3]:= z[jj] end
  end RK2n;
```

RK2n is used to integrate the equations $d^2y_j/dx^2=f_j(x,y_1,\dots,y_n, dy_1/dx,\dots,dy_n/dx)$.

The actual parameters, corresponding to the formal parameters are:

x,a,b: the same as for RK1;

y,ya: the same as for RK1n;

z: an array with elements z[1],...,z[n], the derivatives of the dependent variables;

za: an array with elements za[1],...,za[n], the values of z[j] at x=a;

fxyzj: an expression depending on x, y[j], z[j], j, giving the values of d^2y_j/dx^2;

j: a variable of type integer used, in the actual parameter corresponding to fxyzj, to denote the number of the equation required;

e: an array with elements e[1],...e[4*n]; e[2*j-1] is a relative and e[2*j] is an absolute tolerance associated with y[j]; e[2*(n+j)-1] is a relative and e[2*(n+j)] is an absolute tolerance associated with z[j], cf.(1);

d: an array with elements d[1],...,d[2*n+3]; upon completion of a call of RK2n we have
entier(d[1]+.5) is the number of steps skipped;
d[2] is the step length;
d[3] is equal to b;
d[4],...,d[n+3] are equal to y[1],...,y[n] for x=b;
d[n+4],...,d[2*n+3] are equal to the derivatives z[1],...,z[n] for x=b;

fi: Boolean; if fi then the integration starts at a, with a trial step b-a; if ¬fi then the integration is continued with, as initial conditions x=d[3], y[j]=d[j+3], z[j]=d[n+j+3], h=d[2]*sign(b-d[3]); a, ya, za are ignored;

n: the number of equations.

RK2 integrates d^2y_j/dx^2=fxyzj to x=b, with, if fi then x=a, y[j]=ya[j], z[j]=za[j]; if ¬fi then x=d[3], y[j]=d[j+3], z[j]=d[n+j+3].
Upon completion of a call of RK2n we have x=d[3]=b, y[j]=d[j+3] the value of the dependent variables for x=b, z[j]= d[n+j+3], the value of the derivatives of y[j] at x=b.
RK2n uses as its minimal absolute step length

$$\text{hmin} = \min_{1 \le j \le 2n} (e[2*j-1]*\text{int}+e[2*j]),$$

where int=abs(b-(<u>if</u> fi <u>then</u> a <u>else</u> d[3])).

If a step of length abs(h) $\leq$ hmin is rejected, a step sign(h) * hmin is skipped; a step is rejected if

$th^5 dy_j$ > (abs(z[j]) * e[2 * j-1]+e[2 * j]) * abs(h)/int $\vee$

$th^5 dy'_j$ > (abs(fxyzj) * e[2 * (j+n)-1]+e[2 * (j+n)]) * abs(h)/int,

for any value of j, $1 \leq j \leq n$.

```
procedure RK3(x,a,b,y,ya,z,za,fxy,e,d,fi); value b,fi; real x,a,b,y,ya,z,za,
fxy; Boolean fi; array e,d;
begin real e1,e2,e3,e4,xl,yl,zl,h,int,hmin,hl,absh,k0,k1,k2,k3,k4,k5,discry,
   discrz,toly,tolz,mu,mu1,fhy,fhz; Boolean last,first,reject;
   if fi then begin d[3]:= a; d[4]:= ya; d[5]:= za end; d[1]:= 0; xl:= d[3];
   yl:= d[4]; zl:= d[5]; if fi then d[2]:= b - d[3]; absh:= h:= abs(d[2]);
   if b - xl < 0 then h:= - h; int:= abs(b - xl); hmin:= int × e[1] + e[2];
   hl:= int × e[3] + e[4]; if hl < hmin then hmin:= hl; e1:= e[1]/int; e2:=
   e[2]/int; e3:= e[3]/int; e4:= e[4]/int; first:= reject:= true; if fi then
   begin last:= true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
   else - hmin; absh:= hmin end; if h ≥ b - xl ≡ h ≥ 0 then begin d[2]:=
   h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  if reject then begin x:= xl; y:= yl; k0:= fxy × h end else k0:= k5
   × h/hl; x:= xl + .27639 32022 50021 × h; y:= yl + (zl × .27639 32022
   50021 + k0 × .03819 66011 25011) × h; k1:= fxy × h; x:= xl + .72360
   67977 49979 × h; y:= yl + (zl × .72360 67977 49979 + k1 × .26180
   33988 74989) × h; k2:= fxy × h; x:= xl + h × .5; y:= yl + (zl × .5 +
   k0 × .04687 5 + k1 × .07982 41558 39840 - k2 × .00169 91558 39840)
   × h; k4:= fxy × h; x:= if last then b else xl + h; y:= yl + (zl + k0 ×
   .30901 69943 74947 + k2 × .19098 30056 25053) × h; k3:= fxy × h;
   y:= yl + (zl + k0 × .08333 33333 33333 + k1 × .30150 28323 95825 +
   k2 × .11516 38342 70842) × h; k5:= fxy × h; discry:= abs((- k0 ×
   .5 + k1 × 1.80901 69943 74947 + k2 × .69098 30056 25053 - k4 × 2)
   × h); discrz:= abs((k0 - k3) × 2 - (k1 + k2) × 10 + k4 × 16 + k5 × 4);
   toly:= absh × (abs(zl) × e1 + e2); tolz:= abs(k0) × e3 + absh × e4;
   reject:= discry > toly ∨ discrz > tolz; fhy:= discry/toly; fhz:= discrz/
   tolz; if fhz > fhy then fhy:= fhz; mu:= 1/(1 + fhy) + .45; if reject
   then begin if absh ≤ hmin then begin d[1]:= d[1] + 1; y:= yl; z:= zl;
   first:= true; goto next end; h:= mu × h; goto test end; if first then
   begin first:= false; hl:= h; h:= mu × h; goto acc end; fhy:= mu × h/hl
   + mu - mu1; hl:= h; h:= fhy × h;
acc:  mu1:= mu; z:= zl + (k0 + k3) × .08333 33333 33333 + (k1 + k2) ×
```

```
    .41666 66666 66667;
next: if b ≠ x then begin x1:= x; y1:= y; z1:= z; goto test end; if ¬ last
    then d[2]:= h; d[3]:= x; d[4]:= y; d[5]:= z end RK3;
```

RK3 is used to integrate the equation $d^2y/dx^2=f(x,y)$.
The description of the parameters and of the effect of a call of RK3 on them is the same as for RK2 except for the fact that fxy, unlike fxyz, does not depend on z.

```
procedure RK3n(x,a,b,y,ya,z,za,fxyj,j,e,d,fi,n); value b,fi,n; integer j,n;
real x,a,b,fxyj; Boolean fi; array y,ya,z,za,e,d;
begin integer jj; real xl,h,hmin,int,hl,absh,fhm,discry,discrz,toly,tolz,mu,
  mul,fhy,fhz; Boolean last,first,reject; array yl,zl,k0,k1,k2,k3,k4,k5[1:n],
  ee[1:4×n];
  if fi then begin d[3]:= a; for jj:= 1 step 1 until n do begin d[jj+3]:=
  ya[jj]; d[n+jj+3]:= za[jj] end end; d[1]:= 0; xl:= d[3]; for jj:= 1 step 1
  until n do begin yl[jj]:= d[jj+3]; zl[jj]:= d[n+jj+3] end; if fi then d[2]:=
  b - d[3]; absh:= h:= abs(d[2]); if b - xl < 0 then h:= - h; int:= abs(b -
  xl); hmin:= int × e[1] + e[2]; for jj:= 2 step 1 until 2×n do begin hl:=
  int × e[2×jj-1] + e[2×jj]; if hl < hmin then hmin:= hl end; for jj:= 1
  step 1 until 4×n do ee[jj]:= e[jj]/int; first:= reject:= true; if fi then
  begin last:= true; goto step end;
test:  absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
  else - hmin; absh:= hmin end; if h ≥ b - xl = h ≥ 0 then begin d[2]:=
  h; last:= true; h:= b - xl; absh:= abs(h) end else last:= false;
step:  if reject then begin x:= xl; for jj:= 1 step 1 until n do y[jj]:= yl[jj];
  for j:= 1 step 1 until n do k0[j]:= fxyj × h end else begin fhy:= h/hl;
  for jj:= 1 step 1 until n do k0[jj]:= k5[jj] × fhy end; x:= xl + .27639
  32022 50021 × h;  for jj:= 1 step 1 until n do y[jj]:= yl[jj] + (zl[jj] ×
  .27639 32022 50021 + k0[jj] × .03819 66011 25011) × h; for j:= 1 step 1
  until n do k1[j]:= fxyj × h; x:= xl + .72360 67977 49979 × h; for jj:= 1
  step 1 until n do y[jj]:= yl[jj] + (zl[jj] × .72360 67977 49979 + k1[jj] ×
  .26180 33988 74989) × h; for j:= 1 step 1 until n do k2[j]:= fxyj × h;
  x:= xl + h × .5; for jj:= 1 step 1 until n do y[jj]:= yl[jj] + (zl[jj] × .5
  + k0[jj] × .04687 5 + k1[jj] × .07982 41558 39840 - k2[jj] × .00169
  91558 39840) × h; for j:= 1 step 1 until n do k4[j]:= fxyj × h; x:= if
  last then b else xl + h; for jj:= 1 step 1 until n do y[jj]:= yl[jj] +
  (zl[jj] + k0[jj] × .30901 69943 74947 + k2[jj] × .19098 30056 25053) ×
  h; for j:= 1 step 1 until n do k3[j]:= fxyj × h; for jj:= 1 step 1 until n
  do y[jj]:= yl[jj] + (zl[jj] + k0[jj] × .08333 33333 33333 + k1[jj] × .30150
  28323 95825 + k2[jj] × .11516 38342 70842) × h; for j:= 1 step 1 until
  n do k5[j]:= fxyj × h; reject:= false; fhm:= 0; for jj:= 1 step 1 until n
```

```
  do begin discry:= abs((- k0[jj] × .5 + k1[jj] × 1.80901 69943 74947 +
  k2[jj] × .69098 30056 25053 - k4[jj] × 2) × h); discrz:= abs((k0[jj] -
  k3[jj]) × 2 - (k1[jj] + k2[jj]) × 10 + k4[jj] × 16 + k5[jj] × 4); toly:=
  absh × (abs(zl[jj]) × ee[2×jj-1] + ee[2×jj]); tolz:= abs(k0[jj]) × ee[2×(jj+n)
  -1] + absh × ee[2×(jj+n)]; reject:= discry > toly ∨ discrz > tolz ∨ reject;
  fhy:= discry/toly; fhz:= discrz/tolz; if fhz > fhy then fhy:= fhz; if fhy >
  fhm then fhm:= fhy end; mu:= 1/(1 + fhm) + .45; if reject then begin
  if absh ≤ hmin then begin d[1]:= d[1] + 1; for jj:= 1 step 1 until n do
  begin y[jj]:= yl[jj]; z[jj]:= zl[jj] end; first:= true; goto next end; h:= mu
  × h; goto test end rej; if first then begin first:= false; hl:= h; h:= mu
  × h; goto acc end; fhy:= mu × h/hl + mu - mu1; hl:= h; h:= fhy × h;
acc:  mu1:= mu; for jj:= 1 step 1 until n do z[jj]:= zl[jj] + (k0[jj] + k3[jj])
  × .08333 33333 33333 + (k1[jj] + k2[jj]) × .41666 66666 66667;
next:  if b ≠ x then begin xl:= x; for jj:= 1 step 1 until n do begin yl[jj]:=
  y[jj]; zl[jj]:= z[jj] end; goto test end; if ¬ last then d[2]:= h; d[3]:= x;
  for jj:= 1 step 1 until n do begin d[jj+3]:= y[jj]; d[n+jj+3]:= z[jj] end
  end RK3n;
```

RK3n is used to integrate the system $d^2y_j/dx^2=f(x,y_1,...,y_n)$. The description of the parameters and of the effect of a call of RK3n on them is the same as for RK2n except for the fact that fxyj, unlike fxyzj does not depend on z.

CHAPTER 8

ALGOL 60 procedures with discretely changing integration variable

In this chapter we give two ALGOL 60 procedures for the integration of first order differential equations, where the integration variable to be used, is determined by the procedure.
RK4, to be used for the integration of $dy/dx=f(x,y)$, uses x as integration variable if $abs(f(x,y)) \leq 1$, otherwise y.
RK4n, to be used for the integration of $dy_j/dx=f_j(x,y_1,...,y_n)$, chooses from amongst x and y_j, as integration variable, a variable such that the absolute value of the derivatives of the other variables, with respect to the one chosen, are ≤ 1.

```
procedure RK4(x,xa,b,y,ya,fxy,e,d,fi,xdir,pos); value fi,xdir,pos; Boolean
fi,xdir,pos; real x,xa,b,y,ya,fxy; array e,d;
begin integer i; Boolean iv,first,fir,rej; real k0,k1,k2,k3,k4,k5,fhm,absh,
  discr,s,xl,cond0,s1,cond1,yl,hmin,h,zl,tol,hl,mu,mu1; array e1[1:2];
  procedure RKstep(x,xl,h,y,yl,zl,fxy,d); value xl,yl,zl,h; real x,xl,h,y,yl,
  zl,fxy; integer d;
  begin if d = 2 then goto integrate; if d = 3 then begin x:= xl; y:= yl;
    k0:= fxy × h end else if d = 1 then k0:= zl × h else k0:= k0 × mu;
    x:= xl + h/4.5; y:= yl + k0/4.5; k1:= fxy × h; x:= xl + h/3; y:= yl
    + (k0 + k1 × 3)/12; k2:= fxy × h; x:= xl + h × .5; y:= yl + (k0 + k2
    × 3)/8; k3:= h × fxy; x:= xl + h × .8; y:= yl + (k0 × 53 - k1 × 135
    + k2 × 126 + k3 × 56)/125; k4:= fxy × h; if d ≤ 1 then begin x:= xl
    + h; y:= yl + (k0 × 133 - k1 × 378 + k2 × 276 + k3 × 112 + k4 × 25)
    /168; k5:= fxy × h; discr:= abs(k0 × 21 - k2 × 162 + k3 × 224 - k4
    × 125 + k5 × 42)/14; goto end end;
  integrate:  x:= xl + h; y:= yl + (- k0 × 63 + k1 × 189 - k2 × 36 - k3
    × 112 + k4 × 50)/28; k5:= fxy × h; y:= yl + (k0 × 35 + k2 × 162 +
    k4 × 125 + k5 × 14)/336;
  end:  end RKstep;
  real procedure fzero;
  begin if iv then begin if s = xl then fzero:= cond0 else if s = s1 then
    fzero:= cond1 else begin RKstep(x,xl,s-xl,y,yl,zl,fxy,3); fzero:= b end
    end else begin if s = yl then fzero:= cond0 else if s = s1 then fzero
    := cond1 else begin RKstep(y,yl,s-yl,x,xl,zl,1/fxy,3); fzero:= b end
    end end fzero;
  if fi then begin d[3]:= xa; d[4]:= ya; d[0]:= 1 end; d[1]:= 0; x:= xl:= d[3];
  y:= yl:= d[4]; iv:= d[0] > 0; first:= fir:= true; hmin:= e[0] + e[1]; h:=
  e[2] + e[3]; if h < hmin then hmin:= h;
change:  zl:= fxy; if abs(zl) ≤ 1 then begin if ¬ iv then begin d[2]:= h:=
  h/zl; d[0]:= 1; iv:= first:= true end; if fir then goto A; i:= 1; goto
  again end else begin if iv then begin if ¬ fir then d[2]:= h:= h × zl;
  d[0]:= - 1; iv:= false; first:= true end; if fir then begin h:= e[0] + e[1];
A:  if (if fi then (if iv ≡ xdir then h else h × zl) < 0 ≡ pos else h ×
```

```
  d[2] < 0) then h:= - h end; i:= 1 end;
again:  absh:= abs(h); if absh < hmin then begin h:= sign(h) × hmin; absh
  := hmin end; if iv then begin RKstep(x,xl,h,y,yl,zl,fxy,i); tol:= e[2] ×
  abs(k0) + e[3] × absh end else begin RKstep(y,yl,h,x,xl,1/zl,1/fxy,i);
  tol:= e[0] × abs(k0) + e[1] × absh end; rej:= discr > tol; mu:= tol/(tol
  + discr) + .45; if rej then begin if absh ≤ hmin then begin if iv then
  begin x:= xl + h; y:= yl + k0 end else begin x:= xl + k0; y:= yl + h
  end; d[1]:= d[1] + 1; first:= true; goto next end; h:= h × mu; i:= 0;
  goto again end rej; if first then begin first:= fir; hl:= h; h:= mu × h;
  goto accept end; fhm:= mu × h/hl + mu - mul; hl:= h; h:= fhm × h;
accept:  if iv then RKstep(x,xl,hl,y,yl,zl,fxy,2) else RKstep(y,yl,hl,x,xl,zl,
  1/fxy,2); mul:= mu;
next:  if fir then begin fir:= false; cond0:= b; if ¬ (fi ∨ rej) then h:= d[2]
  end else begin d[2]:= h; cond1:= b; if cond0 × cond1 ≤ 0 then goto
  zero; cond0:= cond1 end; d[3]:= xl:= x; d[4]:= yl:= y; goto change;
zero:  el[1]:= e[4]; el[2]:= e[5]; s1:= if iv then x else y; s:= ZERO(s,if
  iv then xl else yl,s1,fzero,el); if iv then RKstep(x,xl,s-xl,y,yl,zl,fxy,3)
  else RKstep(y,yl,s-yl,x,xl,zl,1/fxy,3); d[3]:= x; d[4]:= y end RK4;
```

The actual parameters corresponding to the formal parameters are:

x: the independent variable; upon completion of a call of RK4 its value is the last value of x reached by the integration;

xa: the starting value of x;

b: an expression depending on x and y; the equation b=0, fulfilled within certain tolerances, specifies the end of the integration; b is evaluated and is tested for change of sign at the end of each step;

y: the dependent variable;

ya: the value of y at x=xa;

fxy: an expression depending on x and y, giving the value of dy/dx;

e: an array with elements e[0],...,e[5]; e[0] and e[2] are relative tolerances, e[1] and e[3] are absolute tolerances associated with x and y respectively; e[4] and e[5] are tolerances used in the determination of the zero of b.

d: an array with elements d[0],...,d[4];
after completion of each step of integration we have
if d[0] > 0 then x is the integration variable; if d[0] < 0 then y is the integration variable;
d[1] is the number of steps skipped;
d[2] is the step length;
d[3] is equal to the last value of x;
d[4] is equal to the last value of y;

fi: a Boolean; if fi then the integration is started with initial conditions x=xa,y=ya; if ¬fi then the integration is started with x=d[3], y=d[4];

xdir, pos: two Booleans; if fi then the integration starts in such a way that if xdir then x else y if pos then increases else decreases;
if ¬fi then xdir and pos are ignored;

RK4 is used to integrate dy/dx=fxy.
The effect of a call of RK4 is the following. First the integration variable is selected and if fi then in accordance with xdir and pos, a step h is done, the absolute value of which is, if x is chosen as integration variable, e[0]+e[1]. If y is chosen as integration variable then the absolute value of the first step is e[2]+e[3].
If ¬fi then the first step done has the same absolute value as above, its sign being in accordance with d[2].
The test, whether b changed sign during this first step is suppressed. This enables us to start at a zero of b.
At each step RK4 determines whether to use x or y as integration variable. After completion of each further step, change of sign of b is tested; if a change of sign is found, then the zero of b is determined using a non-local procedure ZERO(x,a,b,fx,e), the

value of which is the value of a zero of the expression fx, provided that for x=a and x=b, fx has different sign. The array e, with elements e[1] and e[2] is the array of tolerances. Using these tolerances ZERO defines as zero of fx a value of x, $a \leq x \leq b$, such that there exist an x_1 and x_2, $x-(e[1] * x+e[2]) \leq x_1, x_2 \leq x+(e[1] * x+e[2])$, such that the product of the signs of fx for $x=x_1$ and $x=x_2$ differs from 1.

ZERO as called by RK4 uses e[4] and e[5] for its e[1] and e[2].

RK4 uses as its minimal absolute step

$$hmin = \min_{j=0,1}(e[2 * j]+e[2 * j+1]) .$$

If a step of length $abs(h) \leq hmin$ is rejected the integration will be replaced by, if x is the integration variable then

dx=sign(h)* hmin;

dy=sign(h)* hmin * fxy.

If, however, y is then integration variable then

dx= sign(h) * hmin/fxy;

dy=sign(h) * hmin,

where the arguments of fxy are the values of x and y at the beginning of the step.

A step is rejected if x is the integration variable and if

$$th^5dy > (e[2] * abs(fxy)+e[3]) * abs(h),$$

or if y is the integration variable if

$$th^5dx > (e[0]/abs(fxy)+e[1]) * abs(h) .$$

```
procedure RK4n(x,xa,b,fxj,j,e,d,fi,n,l,pos); value fi,n,l,pos; integer j,n,l;
Boolean fi,pos; real b,fxj; array x,xa,e,d;
begin integer i,iv,iv0; Boolean fir,first,rej; real h,cond0,cond1,fhm,absh,
  tol,fh,max,x0,x1,s,hmin,hl,mu,mu1; array xl,discr,y[0:n],k[0:5,0:n],el[1:2];
  procedure RKstep(h,d); value h,d; integer d; real h;
  begin integer i;
    procedure F(t); value t; integer t;
    begin integer i; real p; for j:= 1 step 1 until n do y[j]:= fxj; p:=
      h/y[iv]; for i:= 0 step 1 until n do begin if i ≠ iv then k[t,i]:=
      y[i] × p end end F;
    if d = 2 then goto integrate; if d = 3 then begin for i:= 0 step 1
    until n do x[i]:= xl[i]; F(0) end else if d = 1 then
    begin real p; p:= h/y[iv]; for i:= 0 step 1 until n do begin if i ≠ iv
      then k[0,i]:= p × y[i] end end
    else for i:= 0 step 1 until n do begin if i ≠ iv then k[0,i]:= k[0,i] ×
    mu end; for i:= 0 step 1 until n do x[i]:= xl[i] + (if i = iv then h
    else k[0,i])/4.5; F(1); for i:= 0 step 1 until n do x[i]:= xl[i] + (if
    i = iv then h × 4 else (k[0,i] + k[1,i] × 3))/12; F(2); for i:= 0 step
    1 until n do x[i]:= xl[i] + (if i = iv then h × .5 else (k[0,i] + k[2,i]
    × 3)/8); F(3); for i:= 0 step 1 until n do x[i]:= xl[i] + (if i = iv
    then h × .8 else (k[0,i] × 53 - k[1,i] × 135 + k[2,i] × 126 + k[3,i]
    × 56)/125); F(4); if d ≤ 1 then begin for i:= 0 step 1 until n do
    x[i]:= xl[i] + (if i = iv then h else (k[0,i] × 133 - k[1,i] × 378 +
    k[2,i] × 276 + k[3,i] × 112 + k[4,i] × 25)/168); F(5); for i:= 0 step
    1 until n do begin if i ≠ iv then discr[i]:= abs(k[0,i] × 21 - k[2,i] ×
    162 + k[3,i] × 224 - k[4,i] × 125 + k[5,i] × 42)/14 end; goto end end;
  integrate: for i:= 0 step 1 until n do x[i]:= xl[i] + (if i = iv then h
    else (- k[0,i] × 63 + k[1,i] × 189 - k[2,i] × 36 - k[3,i] × 112 + k[4,i]
    × 50)/28); F(5); for i:= 0 step 1 until n do begin if i ≠ iv then x[i]
    := xl[i] + (k[0,i] × 35 + k[2,i] × 162 + k[4,i] × 125 + k[5,i] × 14)/336
    end;
  end: end RKstep;
```

```
real procedure fzero;
begin if s = x0 then fzero:= cond0 else if s = x1 then fzero:= cond1
  else begin RKstep(s-xl[iv],3); fzero:= b end end fzero;
if fi then begin for i:= 0 step 1 until n do d[i+3]:= xa[i]; d[0]:= d[2]:=
0 end; d[1]:= 0; for i:= 0 step 1 until n do x[i]:= xl[i]:= d[i+3]; iv:=
d[0]; h:= d[2]; first:= fir:= true; y[0]:= 1; goto change;
again: absh:= abs(h); if absh < hmin then begin h:= if h > 0 then hmin
  else - hmin; absh:= abs(h) end; RKstep(h,i); rej:= false; fhm:= 0;
  for i:= 0 step 1 until n do begin if i≠iv then begin tol:= e[2×i] ×
  abs(k[0,i]) + e[2×i+1] × absh; rej:= tol < discr[i] ∨ rej; fh:= discr[i]/
  tol; if fh > fhm then fhm:= fh end end; mu:= 1/(1 + fhm) + .45; if rej
  then begin if absh ≤ hmin then begin for i:= 0 step 1 until n do begin
  if i≠iv then x[i]:= xl[i] + k[0,i] else x[i]:= xl[i] + h end; d[1]:= d[1] + 1;
  first:= true; goto next end; h:= h × mu; i:= 0; goto again end; if first
  then begin first:= fir; hl:= h; h:= mu × h; goto accept end; fh:= mu ×
  h/hl + mu - mu1; hl:= h; h:= fh × h;
accept: RKstep(hl,2); mu1:= mu;
next: if fir then begin fir:= false; cond0:= b; if ¬ (fi ∨ rej) then h:= d[2]
  end else begin d[2]:= h; cond1:= b; if cond0 × cond1 ≤ 0 then goto
  zero; cond0:= cond1 end; for i:= 0 step 1 until n do d[i+3]:= xl[i]:= x[i];
change: iv0:= iv; for j:= 1 step 1 until n do y[j]:= fxj; max:= abs(y[iv]);
  for i:= 0 step 1 until n do begin if abs(y[i]) > max then begin max:=
  abs(y[i]); iv:= i end end; if iv0≠iv then begin first:= true; d[0]:= iv;
  d[2]:= h:= y[iv]/y[iv0] × h end; x0:= xl[iv]; if fir then begin hmin:=
  e[0] + e[1]; for i:= 1 step 1 until n do begin h:= e[2×i] + e[2×i+1]; if
  h < hmin then hmin:= h end; h:= e[2×iv] + e[2×iv+1]; if (fi ∧ (y[1]/y[iv]
  × h < 0 ≡ pos)) ∨ (¬ fi ∧ d[2] × h < 0) then h:= - h end; i:= 1; goto
  again;
zero: el[1]:= e[2×n+2]; el[2]:= e[2×n+3]; x1:= x[iv]; x0:= ZERO(s,x0,x1,
  fzero,el); RKstep(x0-xl[iv],3); for i:= 0 step 1 until n do d[i+3]:= x[i]
  end RK4n;
```

The actual parameters corresponding to the formal parameters are:

x: an array with elements x[0],...,x[n]; x[0] is the independent variable, x[j], $1 \leq j \leq n$, are the dependent variables;

xa: an array with elements xa[0],...,xa[n], the starting values of x[j];

b: an expression depending on x[0],...,x[n]; the equation b=0, fulfilled within certain tolerances specifies the end of the integration; b is evaluated and tested for change of sign at the end of each step;

fxj: an expression depending on x[0],...,x[n],j, giving the value of dx_j/dx_0;

j: a variable of type integer used, in the actual parameter corresponding to fxj, to denote the number of the equation required;

e: an array with elements e[0],...,e[2*n+3]; e[2*j] and e[2*j+1], $0 \leq j \leq n$, are the tolerances associated with x[j]; e[2*n+2] and e[2*n+3] are tolerances used in the determination of the zero of b;

d: an array with elements d[0],...,d[n+3]; after completion of each step we have:
entier (d[0]+.5) denotes the index of the variable used as integration variable;
entier (d[1]+.5) is the number of steps skipped;
d[2] is the step length;
d[j+3] is equal to the last value of x[j];

fi: a Boolean; if fi then the integration is started with initial conditions x[j]=xa[j]; if ¬fi then the integration is started with x[j]=d[j+3];

n: the number of equations;

l: an integer variable;

pos: a Boolean; if fi then the integration starts in such a way that x[1] is if pos then increasing else decreasing.

RK4n is used to integrate dx_j/dx_0=fxj, $1 \leq j \leq n$.
The effect of a call of RK4n is the following. First the integration variable is selected and if fi then in accordance with l and pos, a step h is done, the absolute value of which is e[2*i]+e[2*i-1] where i is the index of the variable chosen as integration variable. If ¬fi then the first step done has the same absolute value as above, the sign being in accordance with d[2].
The test, whether b changed sign during this first step is suppressed. This enables us to start at a zero of b.
At each step RK4n determines which x[j] to use as integration variable. After each further step, change of sign of b is tested; if a change of sign is found, then the zero of b is determined using a non-local procedure ZERO(x,a,b,fx,e), cf. RK4. ZERO as called by RK4n uses e[2*n+2] and e[2*n+3] for its e[1] and e[2].
RK4n uses as its minimal absolute step

$$hmin = \min_{0 \leq j \leq n} (e[2*j]+e[2*j+1]).$$

If a step of length abs(h) ≤ hmin is rejected the integration is replaced by if i is the index of the integration variable
dx[j]=sign(h)*hmin*dx[j]/dx[i]; $0 \leq j \leq n$,
where the arguments occurring in the derivatives are the values of x[j] at the beginning of the step.
A step is rejected if, i being the index of the integration variable

$$th^5 dx_j > (e[2*j]*abs(dx_j/dx_i)+e[2*j+1])*abs(h),$$

for any j, $j \neq i$.

CHAPTER 9

ALGOL 60 procedure with the arc length as integration variable.

In the procedure RK5n, used to integrate the system of equations
$dx_j/dx_0=f_j(x_0,...,x_n)/f_0(x_0,...,x_n)$; $1 \leq j \leq n$,
where f_j and f_0 remain finite, the arc length s is introduced as integration variable. The system solved is

$$dx_j/ds=f_j(x_0,...,x_n)/sqrt(\sum_{i=0}^{n}(f_i(x_0,...,x_n))^2).$$

```
procedure RK5n(x,xa,b,fxj,j,e,d,fi,n,l,pos); value fi,n,l,pos; integer j,n,l;
Boolean fi,pos; real b,fxj; array x,xa,e,d;
begin integer i; Boolean first,fir,rej; real fhm,s,s0,cond0,s1,cond1,h,absh,
  tol,fh,hl,mu,mu1; array y,xl,discr[0:n],k[0:5,0:n],e1[1:2];
  procedure RKstep(h,d); value h,d; integer d; real h;
  begin integer i;
    procedure F(t); value t; integer t;
    begin integer i; real p; for j:= 0 step 1 until n do y[j]:= fxj; p:=
      h/sqrt(SUM(i,0,n,y[i]↑2)); for i:= 0 step 1 until n do k[t,i]:= y[i] × p
      end F;
    if d = 2 then goto integrate; if d = 1 then begin for i:= 0 step 1
    until n do k[0,i]:= k[0,i] × mu; goto A end; for i:= 0 step 1 until n
    do x[i]:= xl[i]; F(0);
  A: for i:= 0 step 1 until n do x[i]:= xl[i] + k[0,i]/4.5; F(1); for i:= 0
    step 1 until n do x[i]:= xl[i] + (k[0,i] + k[1,i] × 3)/12; F(2); for i:= 0
    step 1 until n do x[i]:= xl[i] + (k[0,i] + k[2,i] × 3)/8; F(3); for i:= 0
    step 1 until n do x[i]:= xl[i] + (k[0,i] × 53 - k[1,i] × 135 + k[2,i] ×
    126 + k[3,i] × 56)/125; F(4); if d ≤ 1 then begin for i:= 0 step 1
    until n do x[i]:= xl[i] + (k[0,i] × 133 - k[1,i] × 378 + k[2,i] × 276 +
    k[3,i] × 112 + k[4,i] × 25)/168; F(5); for i:= 0 step 1 until n do
    discr[i]:= abs(k[0,i] × 21 - k[2,i] × 162 + k[3,i] × 224 - k[4,i] × 125
    + k[5,i] × 42)/14; goto end end;
  integrate: for i:= 0 step 1 until n do x[i]:= xl[i] + (- k[0,i] × 63 +
    k[1,i] × 189 - k[2,i] × 36 - k[3,i] × 112 + k[4,i] × 50)/28; F(5); for
    i:= 0 step 1 until n do x[i]:= xl[i] + (k[0,i] × 35 + k[2,i] × 162 + k[4,i]
    × 125 + k[5,i] × 14)/336;
  end: end RKstep;
  real procedure fzero;
  begin if s = s0 then fzero:= cond0 else if s = s1 then fzero:= cond1
    else begin RKstep(s-s0,3); fzero:= b end end fzero;
  if fi then begin for i:= 0 step 1 until n do d[i+3]:= xa[i]; d[1]:= d[2]:= 0
  end; for i:= 0 step 1 until n do x[i]:= xl[i]:= d[i+3]; s:= d[1]; first:= fir
  := true; h:= e[0] + e[1]; for i:= 1 step 1 until n do begin absh:= e[2×i]
```

```
  + e[2×i+1]; if h > absh then h:= absh end; if fi then begin j:= 1; if fxj
  × h < 0 ≡ pos then h:= - h end else if d[2] × h < 0 then h:= - h; i:= 0;
again: RKstep(h,i); rej:= false; fhm:= 0; absh:= abs(h); for i:= 0 step 1
  until n do begin tol:= e[2×i] × abs(k[0,i]) + e[2×i+1] × absh; rej:= tol <
  discr[i] ∨ rej; fh:= discr[i]/tol; if fh > fhm then fhm:= fh end; mu:= 1/
  (1 + fhm) + .45; if rej then begin h:= h × mu; i:= 1; goto again end;
  if first then begin first:= fir; hl:= h; h:= mu × h end else begin fh:=
  mu × h/hl + mu - mu1; hl:= h; h:= fh × h end;
accept: RKstep(hl,2); mu1:= mu; s:= s + hl; if fir then begin cond0:= b;
  fir:= false; if ¬ fi then h:= d[2] end else begin d[2]:= h; cond1:= b; if
  cond0 × cond1 ≤ 0 then goto zero; cond0:= cond1 end; for i:= 0 step 1
  until n do d[i+3]:= x1[i]:= x[i]; d[1]:= s0:= s; i:= 0; goto again;
zero: e1[1]:= e[2×n+2]; e1[2]:= e[2×n+3]; s1:= s; s:= ZERO(s,s0,s1,fzero,
  e1); RKstep(s-s0,3); for i:= 0 step 1 until n do d[i+3]:= x[i]; d[1]:= s
  end RK5n;
```

The actual parameters corresponding to the formal parameters are:

x,xa,b: the same as for RK4n;

fxj: an expression depending on x[0],...,x[n],j; the ratio of two values of fxj, with the same arguments x[0],...,x[n], but with j=i and j=0 respectively, gives the derivative of of x[i] with respect to x[0];

j: a variable of type integer used, in the actual parameter corresponding to fxj, to denote the number of the function required;

e: the same as for RK4n;

d: an array with elements d[1],...,d[n+3]; after completion of each step we have:
abs(d[1]) is the arc length;
d[2] is the step length;
d[i+3] is equal to the last value of x[i];

fi: a Boolean; if fi then the integration is started with initial conditions x[i]=xa[i]; if ¬fi then the integration is started with x[i]=d[i+3];

n: the number of equations;

l: an integer variable;

pos: a Boolean; if fi then the integration starts in such a way that x[l] is if pos then increasing else decreasing.

The real procedure SUM used is a standard procdure in the Electrologica X1 ALGOL 60 system. The declaration would be

```
real procedure SUM (i,a,b,c); value b; integer i,a,b; real c;
begin real s; s:=0; for i:=a step 1 until b do s:=s+c; SUM:=s end
```

RK5n is used to integrate $dx_j/dx_0=f_j(x_0,...,x_n)/f_0(x_0,...,x_n)$. The effect of a call of RK5n is the following. First the arc length is made equal to zero. Then a first step of absolute value $\min_{0\le j\le n}$ (e[2*j]+e[2*j+1]) is done where the sign is chosen to be in accordance with if fi then l and pos else d[2]. The test change of sign of b is suppressed after this first step. After each further step, change of sign of b is tested; if a change of sign is found, then the zero of b is determined using a non-local procedure ZERO(x,a,b,fx,e),cf.RK4. ZERO as called by RK5n uses e[2*n+2] and e[2*n+3] for its e[1] and e[2].
A step is rejected if

$th^5 dx_j$ >(e[2*j]* abs(dx_j/ds)+e[2*j+1])* abs(h),

for any j, $0\le j\le n$.
RK5n uses no minimal step hmin and does not skip steps.

CHAPTER 10

Examples

In this chapter we give five examples of the use of the procedures described above.

I The first program, JAZ162, is given to illustrate the fact that it is possible, given any strategy, to construct an example where this strategy (although virtually foolproof in practice) breaks down completely. In JAZ162 we solve dy/dx=-y.

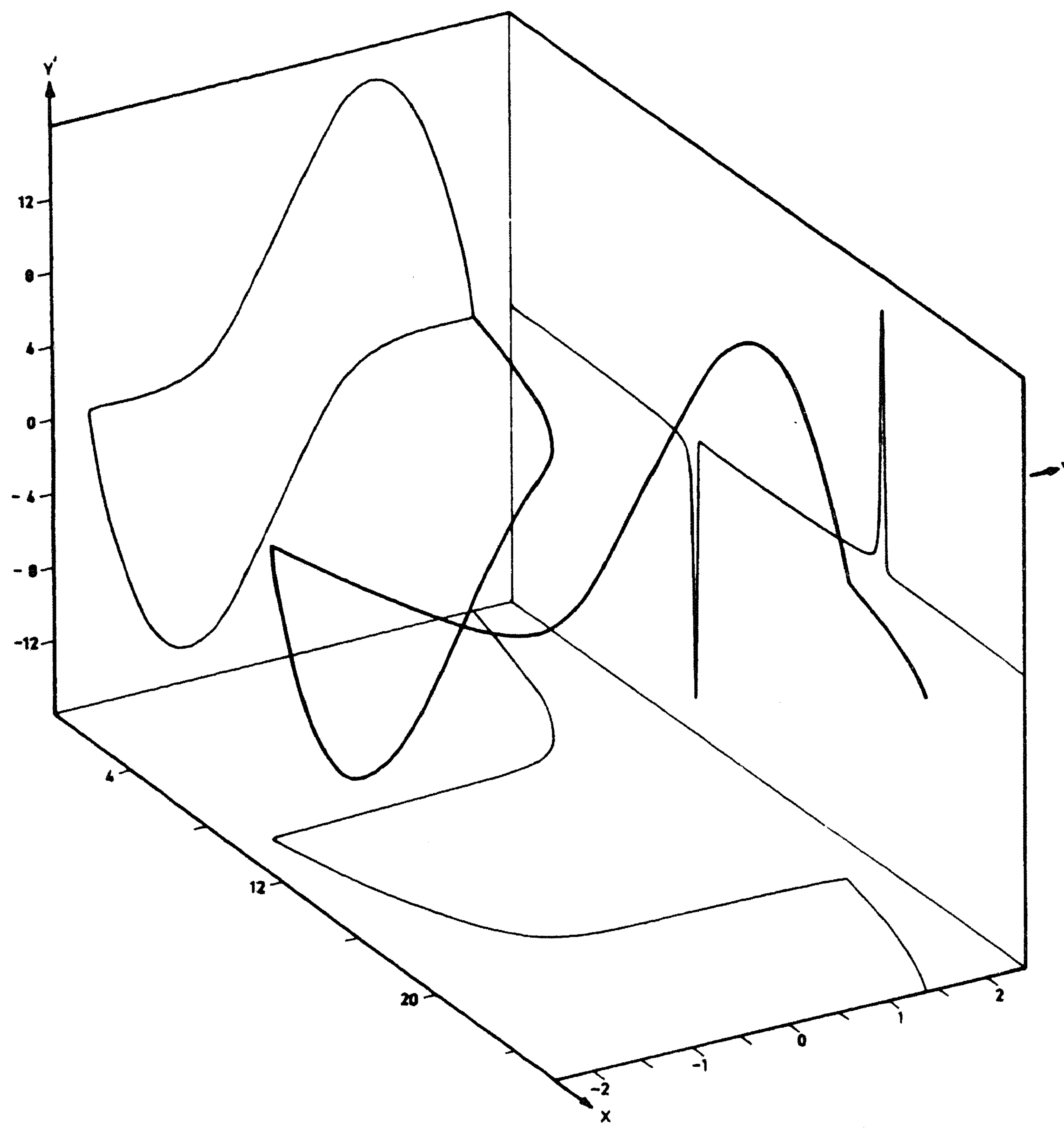
II JAZ161, when executed, solves the van der Pol equation [16]
$d^2y/dx^2-mu(1-y^2)dy/dx+y=0$.
Solving this equation presents great numerical difficulties for large values of mu, although the solution has no singularities; this can be guessed from the graph, drawn for mu=10.

III JAZ163 is used to integrate the van der Pol equation in the phase plane, i.e. written in the form
$dy'/dy-(mu(1-y^2)y'-y)/y'=0$
where y'=dy/dx. As this equation has a limit cycle, it is necessary to switch integration variables. We integrated the equation using the arc length s as integration variable.

IV JAZ164 is given, not because there are numerical difficulties, but rather because it is a big system of coupled equations. It is a system of 15 non-linear second order equations, describing the motion of the five outer planets, the masses of the four inner planets being added to that of the sun.

V In the last example, JAZ165, we integrate a singular function
dy/dx=1/sqrt(1-x)
to show that the integration is possible with the procedures we gave, although the error is greater than in non-singular cases.

In the examples, some procedures available in the Electrologica X1 ALGOL 60 system, are used:



$$y''-10(1-y^2)y'+y=0$$

read: a function designator, that takes the value of the next number on the input tape;

FIXP(m,n,E): when called this procedure punches on paper tape the value of E in the following way: sign, m decimal digits (leading zeroes being replaced by spaces), decimal point, n decimal digits, one space;

ABSFIXP(m,n,E): as FIXP, the sign being replaced by one space;

FLOP(m,n,E): when called this procedure punches the value of E in floating point notation: sign, decimal point, m decimal digits, $_{10}$, sign, n decimal digits (leading zeroes being replaced by spaces), one space;

RUNOUT: gives a length of blank tape;

PUNLCR: a procedure that punches the control symbol NewLineCarriageReturn;

PUTEXT 1(s): a procedure that punches the string s, without the (outermost) quotes.

SUM(i,a,b,c): a function designator the declaration of which could be:

real *procedure* SUM(i,a,b,c); *value* b; *integer* i,a, b; *real* c; *begin* *real* s; s:=0; *for* i:=a *step* 1 *until* b *do* s:=s+c; SUM:= s *end*.

```
begin comment JAZ162, R743;
  real x,y; integer a,j; Boolean fi,first; array e[1:2],d[1:4];
  procedure RK1(x,a,b,y,ya,fxy,e,d,fi); value b,fi; real x,a,b,y,ya,fxy;
  Boolean fi; array e,d;
  begin < body of RK1 > end RK1;
  e[1]:= e[2]:= read; RUNOUT; PUNLCR; PUTEXT1(‹JAZ162, R743›);
  PUNLCR; PUNLCR; PUTEXT1(‹eps = ›); FLOP(2,2,e[1]); PUNLCR;
  PUNLCR; PUTEXT1(‹  x          y›); fi:= true; a:= 1;
A:  PUNLCR; ABSFIXP(2,0,0); ABSFIXP(1,10,1); first:= true; for j:= a step
  a until 10 do begin RK1(x,0,j,y,1,-y,e,d,first); PUNLCR; ABSFIXP(2,0,x);
  ABSFIXP(1,10,y); first:= false end; if fi then begin PUNLCR; a:= 2; fi:=
  false; goto A end end
```

JAZ162, R743		JAZ162, R743		JAZ162, R743			
eps = +.10$_{10}$- 3		eps = +.10$_{10}$- 5		eps = +.10$_{10}$- 7			
x	y	x	y	x	y	x	e^{-x}
0	1.0000000000	0	1.0000000000	0	1.0000000000	0	1.0000000000
1	.3678768464	1	.3678794323	1	.3678794411	1	.3678794412
2	.1353321356	2	.1353352723	2	.1353352832	2	.1353352832
3	.0497841248	3	.0497870555	3	.0497870683	3	.0497870684
4	.0183137400	4	.0183156233	4	.0183156388	4	.0183156389
5	.0067363359	5	.0067379295	5	.0067379469	5	.0067379470
6	.0024766389	6	.0024787351	6	.0024787521	6	.0024787522
7	.0009098208	7	.0009118607	7	.0009118819	7	.0009118820
8	.0003342328	8	.0003354391	8	.0003354625	8	.0003354626
9	.0001227841	9	.0001233931	9	.0001234096	9	.0001234098
10	.0000451061	10	.0000453831	10	.0000453997	10	.0000453999
0	1.0000000000	0	1.0000000000	0	1.0000000000		
2	.1111111111	2	.1111111111	2	.1111111111		
4	.0123456790	4	.0123456790	4	.0123456790		
6	.0013717421	6	.0013717421	6	.0013717421		
8	.0001524158	8	.0001524158	8	.0001524158		
10	.0000169351	10	.0000169351	10	.0000169351		

JAZ162, using RK1, integrates, for different values of the tolerances, the equation

$dy/dx= -y;\ y(0)= 1,$

in two ways: from x=0 until x=1, then continuing until x=2 and so on until x=10; and from x=0 until x=2, then continuing until x=4 and so on until x=10.

We see that the first case behaves as was to be expected; the error becomes smaller with eps. In the second case, however, a disaster occurred. Instead of $y(x)=e^{-x}$ we find $y(x)=3^{-x}$.

The reason for this is the following. For th^5dy we find,

$th^5dy=abs(h^5(2-h)y/240)$,

where y is the value of y(x) at the beginning of a step of length h. When called, RK1 tries to do the integration in one step; thus, when called to integrate form x=0 until x=2, it tries h=2 and with great success as $th^5dy=0$ for h=2. Thus the step is accepted and so on. Furthermore we have

$$y^*(X+h)=y(X)(1-h+\frac{1}{2}h^2-\frac{1}{6}h^3+\frac{1}{24}h^4-\frac{1}{120}h^5+\frac{1}{1440}h^6),$$

whence, with h=2 we find

$$y^*(X+h)=y(X)/9.$$

Of course, in the first case, h, although depending upon y and eps, is much smaller than 1, so that the required accuracy is actually attained.

```
begin comment JAZ161, R743, v.d.Pol;
  integer j,k; real x0; Boolean first; array e[0:7],xa,x[0:2],d[0:5];
  real procedure ZERO(x,a,b,fx,e); value a,b; real x,a,b,fx; array e;
  begin real c,fa,fb,fc,m,i,tol,re,ae; re:= e[1]; ae:= e[2]; x:= a; fa:= fx; x:=
    b; fb:= fx; goto entry;
  goon: if abs(i-b) < tol then i:= b + sign(c-b) × tol; x:= if sign(i-m) = sign
    (b-i) then i else m; a:= b; fa:= fb; b:= x; fb:= fx; if sign(fc) = sign(fb)
    then
  entry: begin c:= a; fc:= fa end; if abs(fb) > abs(fc) then begin a:= b; fa:=
    fb; b:= c; fb:= fc; c:= a; fc:= fa end; m:= (b+c)/2; i:= if fb - fa ≠ 0
    then (a × fb - b × fa)/(fb - fa) else m; tol:= abs(b × re) + ae; if
    abs(m - b) > tol then goto goon; ZERO:= x:= b end ZERO;
  procedure RK4n(x,xa,b,fxj,j,e,d,fi,n,l,pos); value fi,n,l,pos; integer j,n,l;
  Boolean fi,pos; real b,fxj; array x,xa,e,d;
  begin < body of RK4n > end RK4n;
  procedure PUNCH(x); array x;
  begin PUNLCR; FIXP(2,8,x[0]); FIXP(2,8,x[1]); FIXP(2,8,x[2]); if j≠0 then
    FIXP(2,8,x[0]-x0) end PUNCH;
  e[0]:= e[1]:= e[2]:= e[3]:= e[4]:= e[5]:= read; e[6]:= e[7]:= ₁₀-10; RUNOUT;
  PUNLCR; PUTEXT1(⋞JAZ161, R743, v.d.Pol⋟); PUNLCR; PUNLCR;
  PUTEXT1(⋞eps = ⋟); FLOP(2,2,e[0]); PUNLCR; PUNLCR;
  PUTEXT1(⋞       x[0]              x[1]              x[2]              p⋟); xa[0]:=
  xa[2]:= 0; xa[1]:= 2; j:= 0; PUNCH(xa); x0:= xa[0]; first:= true;
A: RK4n(x,xa,x[2],if k=1 then x[2] else 10×(1-x[1]↑2)×x[2]-x[1],k,e,d,first,2,
  0,true); j:= j + 1; PUNCH(x); x0:= x[0]; first:= false; if j < 4 then goto
  A end
```

JAZ161, R743, v.d.Pol

eps = +.10₁₀- 3

x[0]	x[1]	x[2]	p
+ .00000000	+ 2.00000000	+ .00000000	
+ 9.32386578	- 2.01428557	+ .00000000	+ 9.32386578
+18.86305405	+ 2.01428553	+ .00000000	+ 9.53918828
+28.40224162	- 2.01428553	+ .00000000	+ 9.53918756
+37.94142918	+ 2.01428553	+ .00000000	+ 9.53918756

JAZ161, R743, v.d.Pol

eps = +.10₁₀- 5

x[0]	x[1]	x[2]	p
+ .00000000	+ 2.00000000	+ .00000000	
+ 9.32386574	- 2.01428536	+ .00000000	+ 9.32386574
+18.86305053	+ 2.01428536	+ .00000000	+ 9.53918479
+28.40223531	- 2.01428536	+ .00000000	+ 9.53918479
+37.94142010	+ 2.01428536	+ .00000000	+ 9.53918479

RK4n is used in JAZ161 to integrate the van der Pol equation for mu=10, written in the notation of the ALGOL 60 program as

$dx_1/dx_0=x_2;$

$dx_2/dx_0=10(1-x_1^2)x_2-x_1.$

The starting values are $x_0=0$, $x_1=2$, $x_2=0$ and the integration proceeds until the next zero of x_2; then it continues until the next zero and so on until the fourth zero is encountered. Because of the nature of the equation and of the end condition we chose RK4n, which interchanges integration variables. The output, given for the tolerances 10^{-4} and 10^{-6}, consists of x_0, x_1, x_2 and p, where p is the difference of consecutive values of x_0. As mu is large, the solution reaches its periodical limiting solution very soon. The values of x_1, and p are the amplitude and the half period respectively.

Due to the nearly singular behaviour of the solution, the automatic step adjustment shows itself to full advantage. Actually, the step length varies by a factor of several hundreds during a cycle, the small step only being used in a short interval. Without the adjustment, therefore, the integration would take several hundred times longer.

```
begin comment JAZ163, R743, v.d.Pol;
  real mu; integer j,k; Boolean first; array e[0:5],xa,x[0:1],d[1:4];
  real procedure ZERO(x,a,b,fx,e); value a,b; real x,a,b,fx; array e;
  begin < body of ZERO > end ZERO;
  procedure RK5n(x,xa,b,fxj,j,e,d,fi,n,l,pos); value fi,n,l,pos; integer j,n,l;
  Boolean fi,pos; real b,fxj; array x,xa,e,d;
  begin < body of RK5n > end RK5n;
  procedure PUNCH(x); array x;
  begin PUNLCR; FIXP(2,8,x[0]); FIXP(2,8,x[1]); if j≠0 then FIXP(3,6,abs
    (d[1])) end PUNCH;
  e[0]:= e[1]:= e[2]:= e[3]:= read; e[4]:= e[5]:= ⏨-10; mu:= read; RUNOUT;
  PUNLCR; PUTEXT1(⟨JAZ163, R743, v.d.Pol⟩); PUNLCR; PUNLCR;
  PUTEXT1(⟨eps = ⟩); FLOP(2,2,e[0]); PUTEXT1(⟨,  mu = ⟩); ABSFIXP(2,
  0,mu); PUNLCR; PUNLCR;
  PUTEXT1(⟨      x[0]          x[1]            s⟩); xa[0]:= 2; xa[1]:= 0;
  j:= 0; PUNCH(xa); first:= true;
A: RK5n(x,xa,x[1],if k=0 then x[1] else mu×(1-x[0]↑2)×x[1]-x[0],k,e,d,first,
  1,1,false); j:= j + 1; PUNCH(x); first:= false; if j < 4 then goto A end
```

JAZ163, R743, v.d.Pol

eps = +.10⏨- 3 , mu = 0

x[0]	x[1]	s
+ 2.00000000	+ .00000000	
− 2.00001099	− .00000000	+ 6.283195
+ 2.00002259	+ .00000000	+ 12.566427
− 2.00003417	+ .00000000	+ 18.849694
+ 2.00004576	− .00000000	+ 25.132998

JAZ163, R743, v.d.Pol

eps = +.10⏨- 3 , mu = 10

x[0]	x[1]	s
+ 2.00000000	+ .00000000	
— 2.01428606	+ .00000000	+ 29.387384
+ 2.01428627	+ .00000000	+ 58.788435
− 2.01428632	+ .00000000	+ 88.189486
+ 2.01428626	− .00000000	+117.590537

JAZ163, R743, v.d.Pol

eps = +.10⏨- 5 , mu = 0

x[0]	x[1]	s
+ 2.00000000	+ .00000000	
− 2.00000004	+ .00000000	+ 6.283185
+ 2.00000007	+ .00000000	+ 12.566371
− 2.00000011	+ .00000000	+ 18.849556
+ 2.00000014	− .00000000	+ 25.132742

JAZ163, R743, v.d.Pol

eps = +.10⏨- 5 , mu = 10

x[0]	x[1]	s
+ 2.00000000	+ .00000000	
− 2.01428537	− .00000000	+ 29.387383
+ 2.01428537	+ .00000000	+ 58.788433
− 2.01428537	− .00000000	+ 88.189483
+ 2.01428537	+ .00000000	+117.590533

In JAZ163 we used RK5n to integrate the van der Pol equation in the phase plane; in the notation of the program:

$dx_1/dx_0-(mu(1-x_0^2)x_1-x_0)/x_1=0$.

Starting values are $x_0=2$, $x_1=0$. The integration proceeds until the next zero of x_1; then it continues until the next zero and so on until the fourth zero is encountered. Because of the nature of the solution, in the limit a closed curve, we had to use either RK4 or RK5n. We chose RK5n using the arc length s as integration variable. The output given, for the tolerances 10^{-4} and 10^{-6}, with mu=0 and mu=10, consists of x_0, x_1, s.

For mu=0 we have

$dx_1/dx_0=-x_0/x_1$; $x_0=2$; $x_1=0$,

the solution of which is

$x_0^2+x_1^2=4$.

In this case, there is no limit cycle, any concentric circle with (0,0) as origin being a solution. The numerical solution, therefore, departs from the starting circle with an error that continually increases. In this case, s is a multiple of 2π .

For mu=10 we have a stable limit cycle and the numerical solution approaches it very quickly. Again, x_0 is the amplitude. The half period could have been found by computing the integral

$$p=\int\frac{dx_0}{x_1},$$

integrating e.g., from one zero of x_1 to the next one.

The remarks at the end of the former example, JAZ161, concerning the step length apply here also.

```
begin comment JAZ164, R743, Outer Planets;
  integer k,t; real a,k2,x; Boolean fi; array y,ya,z,za[1:15],m[0:5],e[1:60],
  d[1:33];
  real procedure f(k); integer k;
  begin integer i,j,i3,j3; real p; own real array d[1:5,1:5],r[1:5];
    if k≠1 then goto A; for i:= 1 step 1 until 4 do begin i3:= 3×i; for j:=
    i+1 step 1 until 5 do begin j3:= 3×j; p:= (y[i3-2] - y[j3-2])↑2 + (y[i3
    -1] - y[j3-1])↑2 + (y[i3] - y[j3])↑2; d[i,j]:= d[j,i]:= 1/p/sqrt(p) end end;
    for i:= 1 step 1 until 5 do begin i3:= 3×i; d[i,i]:= 0; p:= y[i3-2]↑2 +
    y[i3-1]↑2 + y[i3]↑2; r[i]:= 1/p/sqrt(p) end;
  A:  i:= (k - 1) ÷ 3 + 1; f:= k2 × (- m[0] × y[k] × r[i] + SUM(j,1,5,m[j]
    ×((y[3×(j-i)+k]-y[k])×d[i,j]-y[3×(j-i)+k]×r[j]))) end f;
  procedure RK3n(x,a,b,y,ya,z,za,fxyj,j,e,d,fi,n); value b,fi,n; integer j,n;
  real x,a,b,fxyj; Boolean fi; array y,ya,z,za,e,d;
  begin < body of RK3n > end RK3n;
  procedure PUNCH(x); array x;
  begin integer k; PUNLCR; PUTEXT1(‹T = ›); ABSFIXP(7,1,t+a);
    PUNLCR; PUNLCR; for k:= 1 step 1 until 5 do begin if k=1 then
    PUTEXT1(‹J    ›) else if k=2 then PUTEXT1(‹S    ›) else if k=3 then
    PUTEXT1(‹U    ›) else if k=4 then PUTEXT1(‹N    ›) else
    PUTEXT1(‹P    ›); FIXP(2,9,x[3×k-2]); FIXP(2,9,x[3×k-1]); FIXP(2,9,
    x[3×k]); PUNLCR end end PUNCH;
  a:= read; for k:= 1 step 1 until 15 do begin ya[k]:= read; za[k]:= read
  end; for k:= 0 step 1 until 5 do m[k]:= read; k2:= read; e[1]:= read;
  for k:= 2 step 1 until 60 do e[k]:= e[1]; RUNOUT; PUNLCR; PUTEXT1
  (‹JAZ164, R743, Outer Planets›); PUNLCR; PUNLCR; for k:= 1 step 1
  until 15 do begin FLOP(12,2,ya[k]); FLOP(12,2,za[k]); PUNLCR end;
  for k:= 0 step 1 until 5 do begin PUNLCR; FLOP(12,2,m[k]) end;
  PUNLCR; PUNLCR; FLOP(12,2,k2); PUNLCR; PUNLCR; PUTEXT1
  (‹eps = ›); FLOP(2,2,e[1]); PUNLCR; t:= 0; PUNCH(ya); fi:= true; for
  t:= 500,1000 do begin RK3n(x,0,t,y,ya,z,za,f(k),k,e,d,fi,15); fi:= false;
  PUNCH(y) end end
```

JAZ164, R743, Outer Planets

+.342947415189$_{10}$+ 1 −.557160570446$_{10}$− 2
+.335386959711$_{10}$+ 1 +.505696783289$_{10}$− 2
+.135494901715$_{10}$+ 1 +.230578543901$_{10}$− 2
+.664145542550$_{10}$+ 1 −.415570776342$_{10}$− 2
+.597156957878$_{10}$+ 1 +.365682722812$_{10}$− 2
+.218231499728$_{10}$+ 1 +.169143213293$_{10}$− 2
+.112630437207$_{10}$+ 2 −.325325669158$_{10}$− 2
+.146952576794$_{10}$+ 2 +.189706021964$_{10}$− 2
+.627960525067$_{10}$+ 1 +.877265322780$_{10}$− 3
−.301552268759$_{10}$+ 2 −.240476254170$_{10}$− 3
+.165699966404$_{10}$+ 1 −.287659532608$_{10}$− 2
+.143785752721$_{10}$+ 1 −.117219543175$_{10}$− 2
−.211238353380$_{10}$+ 2 −.176860753121$_{10}$− 2
+.284465098142$_{10}$+ 2 −.216393453025$_{10}$− 2
+.153882659679$_{10}$+ 2 −.148647893090$_{10}$− 3

+.100000597682$_{10}$+ 1
+.954786104043$_{10}$− 3
+.285583733151$_{10}$− 3
+.437273164546$_{10}$− 4
+.517759138449$_{10}$− 4
+.277777777778$_{10}$− 5

+.295912208286$_{10}$− 3

eps = +.10$_{10}$− 3

T = 2430000.5

J	+ 3.429474152	+ 3.353869597	+ 1.354949017
S	+ 6.641455425	+ 5.971569579	+ 2.182314997
U	+11.263043721	+14.695257679	+ 6.279605251
N	−30.155226876	+ 1.656999664	+ 1.437857527
P	−21.123835338	+28.446509814	+15.388265968

T = 2430500.5

J	− .049534455	+ 4.714982495	+ 2.023963513
S	+ 4.277614611	+ 7.483210480	+ 2.909418313
U	+ 9.582290073	+15.567813885	+ 6.685732380
N	−30.235783049	+ .215924799	+ .849602274
P	−21.994991444	+27.345130515	+15.303485551

T = 2431000.5

J	− 3.535429691	+ 3.610053139	+ 1.635176964
S	+ 1.496149963	+ 8.261862331	+ 3.351487277
U	+ 7.805112554	+16.281370896	+ 7.023579152
N	−30.235569469	− 1.228279723	+ .257987477
P	−22.837219187	+26.205087209	+15.197406000

eps = $+.10_{10}-5$

T = 2430000.5

J	+ 3.429474152	+ 3.353869597	+ 1.354949017
S	+ 6.641455425	+ 5.971569579	+ 2.182314997
U	+11.263043721	+14.695257679	+ 6.279605251
N	−30.155226876	+ 1.656999664	+ 1.437857527
P	−21.123835338	+28.446509814	+15.388265968

T = 2430500.5

J	− .049532751	+ 4.714984315	+ 2.023964252
S	+ 4.277614624	+ 7.483210494	+ 2.909418318
U	+ 9.582290074	+15.567813886	+ 6.685732381
N	−30.235783047	+ .215924801	+ .849602274
P	−21.994991442	+27.345130517	+15.303485552

T = 2431000.5

J	− 3.535427150	+ 3.610059334	+ 1.635179559
S	+ 1.496149997	+ 8.261862381	+ 3.351487296
U	+ 7.805112556	+16.281370902	+ 7.023579155
N	−30.235569466	− 1.228279717	+ .257987479
P	−22.837219185	+26.205087215	+15.197406002

eps = $+.10_{10}-7$

T = 2430000.5

J	+ 3.429474152	+ 3.353869597	+ 1.354949017
S	+ 6.641455425	+ 5.971569579	+ 2.182314997
U	+11.263043721	+14.695257679	+ 6.279605251
N	−30.155226876	+ 1.656999664	+ 1.437857527
P	−21.123835338	+28.446509814	+15.388265968

T = 2430500.5

J	− .049532744	+ 4.714984323	+ 2.023964255
S	+ 4.277614624	+ 7.483210494	+ 2.909418318
U	+ 9.582290074	+15.567813886	+ 6.685732381
N	−30.235783047	+ .215924801	+ .849602274
P	−21.994991442	+27.345130517	+15.303485552

T = 2431000.5

J	− 3.535427138	+ 3.610059361	+ 1.635179571
S	+ 1.496149998	+ 8.261862381	+ 3.351487296
U	+ 7.805112556	+16.281370902	+ 7.023579155
N	−30.235569466	− 1.228279717	+ .257987479
P	−22.837219185	+26.205087215	+15.197406002

We used in JAZ164 the procedure RK3n to integrate the 15 differential equations describing the motion of the five outer planets as given in Astronomical Papers, vol XII [17]. The inner planets are not considered, their mass being added to that of the sun. The equations are

$$dx_i^2/dt^2=-k2((1+m_0+m_i)x_i/r_j^3+\sum_{\substack{k=1\\k\neq i}}^{15} m_l((x_k-x_i)/d_{jl}^3-x_k/r_l^3);$$

$i=1(1)15$; $j=entier((i+2.5)/3)$; $l=entier((k+2.5)/3)$.

m_0 is the sum of the masses of the sun and the four inner planets; r_j is the distance of the j^{-th} planet from the sun and d_{jl} the mutual distance of the j^{-th} and l^{-th} planets.

For Julian day 2430000.5 the coordinates x_i were taken from [17]. The velocities dx_i/dt were obtained by numerical differentiation of a high accuracy table [17] (p322-326).

The output of JAZ164 consists first of the input: x_i, dx_i/dt, m_j and the gravitational constant k2.

Output of the integration is given for T=2430500.5 and T=2431000.5. It can be seen , since the values obtained with eps=10^{-8} are correct according to the table, that for larger eps the coordinates except those for Jupiter are correct already. This means that the step length used in the integration was controlled by the equations for Jupiter and that the other equations were integrated with a step that was far too small. This is an unfortunate circumstance which often arises in the integration of coupled differential equations. For comparison, we put the mass of Pluto equal to zero, thereby eliminating its influence on the other planets. The equation remaining for Pluto and the coordinates resulting from it are nonsense. The influence of Pluto, even after a short period of 1000 days is clearly seen from the results.

JAZ164, R743, Outer Planets

+.342947415189$_{10}$+ 1 -.557160570446$_{10}$- 2
+.335386959711$_{10}$+ 1 +.505696783289$_{10}$- 2
+.135494901715$_{10}$+ 1 +.230578543901$_{10}$- 2
+.664145542550$_{10}$+ 1 -.415570776342$_{10}$- 2
+.597156957878$_{10}$+ 1 +.365682722812$_{10}$- 2
+.218231499728$_{10}$+ 1 +.169143213293$_{10}$- 2
+.112630437207$_{10}$+ 2 -.325325669158$_{10}$- 2
+.146952576794$_{10}$+ 2 +.189706021964$_{10}$- 2
+.627960525067$_{10}$+ 1 +.877265322780$_{10}$- 3
-.301552268759$_{10}$+ 2 -.240476254170$_{10}$- 3
+.165699966404$_{10}$+ 1 -.287659532608$_{10}$- 2
+.143785752721$_{10}$+ 1 -.117219543175$_{10}$- 2
-.211238353380$_{10}$+ 2 -.176860753121$_{10}$- 2
+.284465098142$_{10}$+ 2 -.216393453025$_{10}$- 2
+.153882659679$_{10}$+ 2 -.148647893090$_{10}$- 3

+.100000597682$_{10}$+ 1
+.954786104043$_{10}$- 3
+.285583733151$_{10}$- 3
+.437273164546$_{10}$- 4
+.517759138449$_{10}$- 4
+ 0

+.295912208286$_{10}$- 3

eps = +.10$_{10}$- 7

T = 2430000.5

J	+ 3.429474152	+ 3.353869597	+ 1.354949017
S	+ 6.641455425	+ 5.971569579	+ 2.182314997
U	+11.263043721	+14.695257679	+ 6.279605251
N	-30.155226876	+ 1.656999664	+ 1.437857527
P	-21.123835338	+28.446509814	+15.388265968

T = 2430500.5

J	- .049532732	+ 4.714984323	+ 2.023964254
S	+ 4.277614641	+ 7.483210502	+ 2.909418320
U	+ 9.582290108	+15.567813908	+ 6.685732388
N	-30.235783114	+ .215924764	+ .849602256
P	-21.994991481	+27.345130567	+15.303485580

T = 2431000.5

J	- 3.535427088	+ 3.610059340	+ 1.635179555
S	+ 1.496150074	+ 8.261862405	+ 3.351487298
U	+ 7.805112697	+16.281370989	+ 7.023579184
N	-30.235569734	- 1.228279864	+ .257987407
P	-22.837219343	+26.205087416	+15.197406114

```
begin comment JAZ165, R743;
  real x,y; array e[1:2],d[1:4];
  procedure RK1(x,a,b,y,ya,fxy,e,d,fi); value b,fi; real x,a,b,y,ya,fxy;
  Boolean fi; array e,d;
  begin < body of RK1 > end RK1;
  RUNOUT; PUNLCR; PUTEXT1(‹JAZ165, R743›); PUNLCR; e[2]:= 0; for
  e[1]:= ⏨-4,⏨-6 do begin PUNLCR; PUTEXT1(‹eps = ›); FLOP(2,2,e[1]);
  RK1(x,0,1,y,0,1/sqrt(1-x),e,d,true); FIXP(2,8,y); ABSFIXP(1,3,(2-y)/(14×
  e[1]↑3)↑.125); ABSFIXP(2,0,d[1]) end end
```

```
JAZ165, R743

eps = +.10⏨- 3 + 1.95358909  1.055   6
eps = +.10⏨- 5 + 1.99187085  1.039  17
```

As last example we integrated the integral discussed in chapter 6, dy/dx=1/sqrt(1-x); y(0)=0,
with eps=10^{-4} and 10^{-6}, up to x=1.
The output gives the result of the integration together with the ratio of the error made and that given in chapter 6 (26). The agreement is good. In addition the number of steps skipped is given.

To check the behaviour of the integration procedures, we did some examples which counted the steps and gave additional output. The results we found are:

JAZ163: with mu=10, starting at x_0=2, x_1=0, integrating until the first zero of x_1 we have:

eps	steps accepted	function values used	steps rejected	function values not used	discr/tol
10^{-4}	145	1015	26	130	.68
10^{-6}	400	2800	22	110	.79

JAZ161: starting at $x_0=0$, $x_1=2$, $x_2=0$, integrating until the first zero of x_2 we have:

eps	steps accepted	function values used	steps rejected	function values not used	discr/tol
10^{-4}	139	973	20	100	.69
10^{-6}	418	2926	22	110	.79

Here discr/tol is the mean ratio of the last term taken into account to the tolerance. The choice of a safety margin of 5% in the step length is aimed at discr/tol=.80. This means that, for eps=10^{-6}, the ideal step length would have been 5% larger than the one actually used. Another 4% of computing time is lost due to rejections. The conclusion is that, if the permissible step length had been known in advance, we could have solved the problem in only 10% less time.