ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVII, 4 (1983)

J. S. CHOMICZ, A. OLEJNICZAK and M. SZYSZKOWICZ (Wrocław)

# A METHOD FOR FINDING THE STEP SIZE OF INTEGRATION OF A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS

In this paper we give a method for finding the step size of integration of an initial value problem. This method can be used with a slight modification in other numerical problems (see [1]). Some numerical results are presented which show the advantages of the method.

**1. Method of choice of the step size.** The initial value problem which we wish to solve numerically is

$$y' = f(x, y), \quad y(a) = y_0, \quad x \in [a, b], \quad y, f \in R^n. \tag{1.1}$$

Let (1.1) have a unique solution. For the numerical solution of this problem we apply a one-step method of rank $p$ ($p \geqslant 1$) and of the form

$$\begin{cases} \eta_k^0 = y_{0k} & \\ & (k = 1, 2, \ldots, n), \\ \eta_k^{i+1} = \eta_k^i + h_i \Phi(x, \eta_1^i, \eta_2^i, \ldots, \eta_n^i, h_i) & (i = 0, 1, \ldots). \end{cases} \tag{1.2}$$

In the case of one equation ($n = 1$), one of the possible methods for finding the step size $h_i$ was given by Krylov ([2], p. 103-107) and Stoer ([3], p. 114-118). In this paper we apply this method to the general case of a system of ordinary differential equations. Let $\eta_k(x_0+h, h)$ denote the approximation of the solution of problem (1.1) at the point $x_0+h$ obtained from (1.2) with step size $h$ and let $\eta_k(x_0+h, h/2)$ denote the approximation of this solution by twofold application of (1.2) with step size $h/2$.

For one-step methods of rank $p$ we have (see [2])

$$y_k(x_0+h) = \eta_k(x_0+h, h) + C_k^0(x_0)h^{p+1} + C_k^1(x_{0k})h^{p+2} \quad (k = 1, 2, \ldots, n),$$

where $x_{0k}$ are the points from $[x_0, x_0+h]$, $y_k(x)$ is the exact solution of (1.1), and $C_k^i(x)$ are any continuous functions independent of $h$. Since we are interested only in the rank of the error, we can write

$$y_k(x_0+h)-\eta_k(x_0+h, h) \doteq C_k(x_0)h^{p+1} \tag{1.3}$$

and

$$y_k(x_0+h)-\eta_k(x_0+h, h/2) \doteq 2C_k(x_0)(h/2)^{p+1} \tag{1.4}$$

for $k = 1, 2, \ldots, n$, where $C_k(x_0) = C_k^0(x_0)$.

The derivation of formula (1.4) is as follows:

From (1.3) with step size $h/2$ at the point $x_0$ we have

$$y_k(x_0+h/2)-\eta_k(x_0+h/2, h/2) \doteq C_k(x_0)(h/2)^{p+1}. \tag{1.5}$$

Applying this formula twice at the points $x_0, x_0+h/2$ with $h/2$ we have

$$y_k(x_0+h) \doteq y_k(x_0)+\frac{h}{2}\Phi\left(x_0, y_1(x_0), y_2(x_0), \ldots, y_n(x_0), \frac{h}{2}\right)+$$

$$+C_k(x_0)\left(\frac{h}{2}\right)^{p+1}+\frac{h}{2}\Phi\left(x_0+\frac{h}{2}, y_1\left(x_0+\frac{h}{2}\right), y_2\left(x_0+\frac{h}{2}\right), \ldots, y_n\left(x_0+\frac{h}{2}\right), \frac{h}{2}\right)+$$

$$+C_k\left(x_0+\frac{h}{2}\right)\left(\frac{h}{2}\right)^{p+1}.$$

Since we omit elements with the exponent $h$ greater than $p+1$, we can write

$$C_k(x_0+h/2)(h/2)^{p+1} \doteq C_k(x_0)(h/2)^{p+1}.$$

Using this formula and replacing $y_1(x_0+h/2), \ldots, y_n(x_0+h/2)$ by suitable values from (1.5) and omitting elements with the exponent $h$ greater than $p+1$ in the Taylor series $\Phi$, we have

$$y_k(x_0+h) \doteq \eta_k(x_0+h, h/2)+2C_k(x_0)(h/2)^{p+1} \qquad (k = 1, 2, \ldots, n).$$

From (1.3) and (1.4) we obtain

$$\Delta_k = \eta_k\left(x_0+h, \frac{h}{2}\right)-\eta_k(x_0+h, h) \doteq C_k(x_0)h^{p+1}\frac{2^p-1}{2^p} \tag{1.6}$$

$$(k = 1, 2, \ldots, n).$$

Let $H$ denote the integration step which gives an approximation of the solution with given error and let $\varepsilon$ denote the error vector. We want to obtain

$$\varepsilon_k \geqslant |y_k(x_0+H)-\eta_k(x_0+H, H)| \doteq |C_k(x_0)H^{p+1}| \qquad (k = 1, 2, \ldots, n). \tag{1.7}$$

Let $H = h/\omega$. By (1.7) we can write

$$\varepsilon_k \geqslant |C_k(x_0)H^{p+1}| = |C_k(x_0)(h/\omega)^{p+1}|,$$

and using (1.6) we have

$$\varepsilon_k \geqslant \frac{|\Delta_k|2^p}{\omega^{p+1}(2^p-1)} \qquad (k = 1, 2, \ldots, n).$$

Finally, for finding $\omega$ which designates $H$ we have

$$\omega = a \sqrt[p+1]{\frac{2^p}{2^p-1} \max_{1\leqslant k\leqslant n} \frac{|\Delta_k|}{|\varepsilon_k|}}, \tag{1.8}$$

where the constant $a > 1$ is introduced for the algorithm to work.

It is clear from (1.3) that the error of the obtained solution is $O(H^{p+1})$. Applying Richardson's extrapolation we obtain

$$\eta_k^*(x_0+H) = \eta_k\left(x_0+H, \frac{H}{2}\right) + \frac{\eta_k(x_0+H, H/2) - \eta_k(x_0+H, H)}{2^p-1} \tag{1.9}$$

$$(k = 1, 2, \ldots, n),$$

where $\eta_k^*(x_0+H)$ is a solution with error $O(H^{p+2})$.

We may accept this value as the exact value to the next step of method (1.2).

The numerical realization of this method is shown in Fig. 1.

**2. The procedure** *diffsystrunkut4*. The procedure *diffsystrunkut4* calculates the approximate values $y_1(x), y_2(x), \ldots, y_n(x)$ of the solutions of the system of differential equations

$$y_k' = f_k(x, y_1, y_2, \ldots, y_n), \tag{2.1}$$

$$y_k(x_0) = y_{0k} \tag{2.2}$$

for $k = 1, 2, \ldots, n$ $(n \geqslant 1)$ using the standard Runge-Kutta method of fourth order.

Data:

*x0* — parameter in (2.2);
*x1* — value of $x$ for which we want to obtain a solution of (2.1);
*eps, eta* — positive parameters $\varepsilon, \eta$ characterizing the precision of the solutions;
*hmin* — the minimum admissible absolute value of the step size $h$;
*n* — number of equations;
*y0[1 : n]* — values of $y_{0k}$ in (2.2).

Results:

*x0* — given value of *x1*;
*y0[1 : n]* — approximate values of $y_k(x_1)$ $(k = 1, 2, \ldots, n)$.

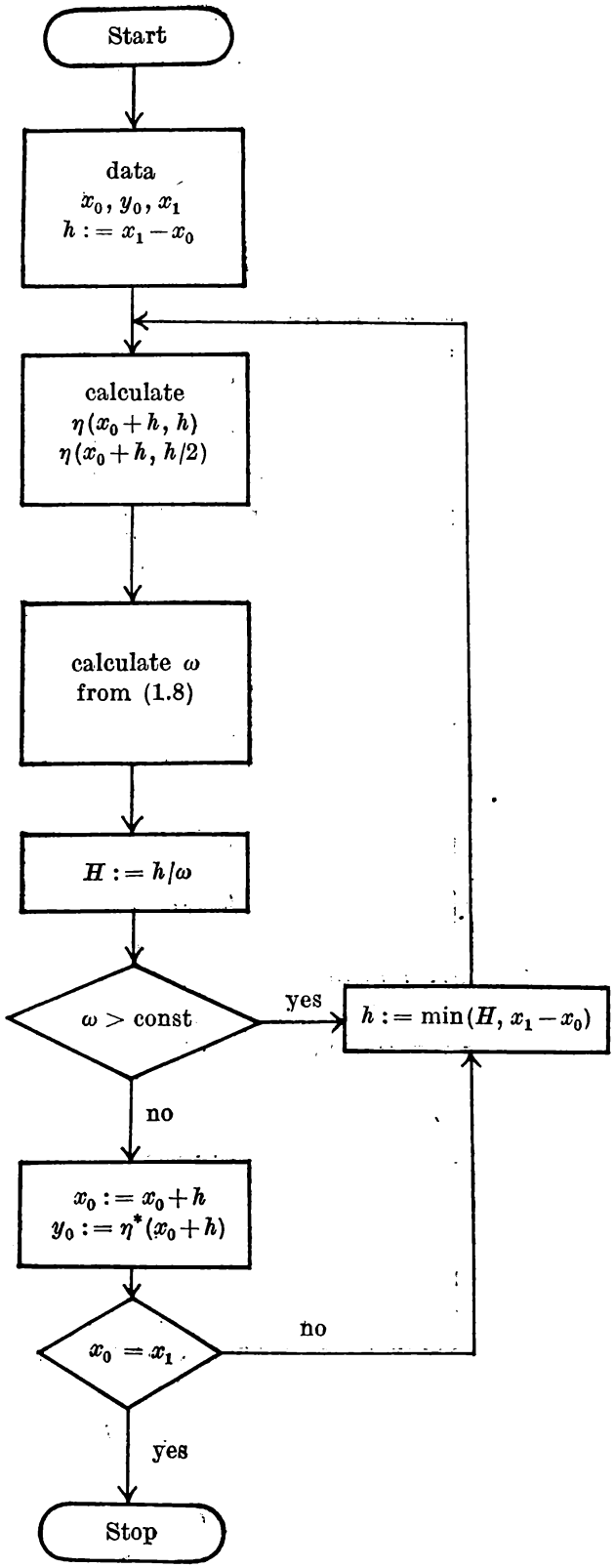


Fig. 1. Flow diagram of the method

```
procedure diffsystrunkut4(x0,x1,eps,eta,hmin,n,y0,notacc,f);
 value x1,eps,eta,hmin,n;
 real x0,x1,eps,eta,hmin;
 integer n;
 array y0;
 label notacc;
 procedure f;
 begin
  real h,hh,ww,w1,w2,w3,w4;
  integer i;
  Boolean last;
  array k1,yh,y1,y2,y3[1:n];
  procedure steprunkut(h,x,y,df);
   value h,x;
   real h,x;
   array y,df;
   begin
    w1:=w4:=.5×h;
    w2:=x;
    for i:=1 step 1 until n do
     begin
      yh[i]:=y[i];
      df[i]:=.0
     end i;
    for w3:=w4,h,h,w4 do
     begin
      f(w2,n,yh,k1);
      for i:=1 step 1 until n do
       begin
        w2:=k1[i];
```

```
            df[i]:=df[i]+w3×w2;
            yh[i]:=y[i]+w1×w2
           end i;
          w2:=x+w1;
          w1:=w3
         end w3;
        for i:=1 step 1 until n do
         df[i]:=y[i]+.333333333333×df[i]
       end steprunkut;
      eps:=.033333333333/eps;
      h:=x1-x0;
      last:=true;
    conth:
      hh:=.5×h;
      steprunkut(h,x0,y0,y1);
      steprunkut(hh,x0,y0,y2);
      steprunkut(hh,x0+hh,y2,y3);
      ww:=.0;
      for i:=1 step 1 until n do
       begin
        w3:=y3[i];
        w1:=w3-y1[i];
        w3:=y3[i]:=w3+.066666666666×w1;
        w1:=abs(w1);
        w3:=abs(w3);
        if w3<eta
         then w3:=eta;
        w1:=w1/w3;
        if w1>ww
         then ww:=w1
```

```
 end i;
ww:=if ww=0 then eta else (eps<ww)↑.2×1.25;
hh:=h/ww;
if ww>1.25
 then
 begin
  if abs(hh)<hmin
   then go to notacc;
  last:=false
 end ww>1.25
 else
 begin
  x0:=x0+h;
  for i:=1 step 1 until n do
   y0[i]:=y3[i];
  if last
   then go to end;
  w1:=x1-x0;
  if (w1-hh)×h<0
   then
   begin
    hh:=w1;
    last:=true
   end (w1-hh)×h<0
 end ww<1.25;
 h:=hh;
 go to conth;
end:
end diffsystrunkut4
```

Other parameters:

*notacc* — label (outside of the body of the procedure *diffsystrunkut4*) to which an exit is made if the computed step size $h$ is smaller than *hmin*.

Remark. After a jump to *notacc* the variable *x0* has the value of $x$ ($x \in [x0, x1]$) for which a correct solution was obtained and the array *y0* contains this solution.

*f* — identifier of the procedure with the head

**procedure** $f(x, n, y, d)$;
**value** $x, n, y$;
**real** $x$;
**integer** $n$;
**array** $y, d$;

which for given $n$, $x$, and $y[1:n]$ computes the values of $d[1:n]$ of the right-hand side of (2.1).

**3. Examples of the use of the procedure.** We give three examples of the use of the procedure *diffsystrunkut4*.

As the first example we take a problem which was given by Stoer ([3], p. 117-118):

$$y' = -200xy^2, \quad y(-3) = 1/901$$

with the exact solution $y(x) = 1/(1+100x^2)$. We obtained the following results:

| $(\eta - y(0))/y(0)$ | Number of evaluations of the right-hand side | *eps* = *eta* |
|---|---|---|
| $-7.246325 \cdot 10^{-3}$ | 276 | $10^{-5}$ |
| $-5.561725 \cdot 10^{-4}$ | 456 | $10^{-6}$ |
| $-5.636424 \cdot 10^{-5}$ | 732 | $10^{-7}$ |
| $-4.719455 \cdot 10^{-6}$ | 1152 | $10^{-8}$ |
| $-5.210094 \cdot 10^{-7}$ | 1848 | $10^{-9}$ |

The remaining two examples are the following:

Example A.

$$\begin{cases} y_1' = 1/y_2, \ y_1(0) = 1, \\ y_2' = -1/y_1, \ y_2(0) = 1, \end{cases} \text{ with the solution } \begin{cases} y_1(x) = e^x, \\ y_2(x) = e^{-x}. \end{cases}$$

Example B.

$$\begin{cases} y_1' = y_2, \ y_1(0) = 0, \\ y_2' = -y_1, \ y_2(0) = 1, \end{cases} \text{ with the solution } \begin{cases} y_1(x) = \sin x, \\ y_2(x) = \cos x. \end{cases}$$

For these examples we obtained results which are given in Tables A and B, respectively. For comparison Table A presents also results obtained with the library procedure *RungeKutta4* [4].

TABLE A

| $x$ | *diffsystrunkut4* | | *RungeKutta4* | |
|---|---|---|---|---|
| | Errors for $eps = eta = 10^{-9}$ | Number of evaluations of the right-hand side | Errors for $eps = eta = 10^{-9}$ | Number of evaluations of the right-hand side |
| 0.5 | $3.53 \cdot 10^{-11}$ | 132 | $-7.06 \cdot 10^{-11}$ | 224 |
| | $0.00 \cdot 10^{0}$ | | $2.39 \cdot 10^{-11}$ | |
| 1.0 | $-4.28 \cdot 10^{-11}$ | 132 | $-1.07 \cdot 10^{-10}$ | 224 |
| | $1.58 \cdot 10^{-10}$ | | $7.91 \cdot 10^{-11}$ | |
| 1.5 | $-1.29 \cdot 10^{-10}$ | 132 | $-7.79 \cdot 10^{-11}$ | 224 |
| | $2.44 \cdot 10^{-10}$ | | $8.15 \cdot 10^{-11}$ | |
| 2.0 | $-2.52 \cdot 10^{-10}$ | 132 | $-1.57 \cdot 10^{-10}$ | 224 |
| | $3.49 \cdot 10^{-10}$ | | $1.07 \cdot 10^{-10}$ | |
| 4.0 | $-5.79 \cdot 10^{-10}$ | 492 | $-1.87 \cdot 10^{-10}$ | 816 |
| | $9.18 \cdot 10^{-10}$ | | $2.48 \cdot 10^{-11}$ | |
| 10.0 | $-4.61 \cdot 10^{-9}$ | 1416 | $1.51 \cdot 10^{-9}$ | 3136 |
| | $5.86 \cdot 10^{-9}$ | | $-1.58 \cdot 10^{-9}$ | |

TABLE B

| $x$ | Errors for $eps = eta = 10^{-3}$ | Number of evaluations of the right-hand side | Errors for $eps = eta = 10^{-6}$ | Number of evaluations of the right-hand side |
|---|---|---|---|---|
| 0.5 | $1.33 \cdot 10^{-6}$ | 12 | $3.27 \cdot 10^{-8}$ | 48 |
| | $4.04 \cdot 10^{-6}$ | | $4.75 \cdot 10^{-8}$ | |
| 1.0 | $5.37 \cdot 10^{-6}$ | 12 | $1.43 \cdot 10^{-7}$ | 36 |
| | $1.03 \cdot 10^{-5}$ | | $1.99 \cdot 10^{-7}$ | |
| 1.5 | $1.00 \cdot 10^{-5}$ | 12 | $1.92 \cdot 10^{-7}$ | 48 |
| | $5.85 \cdot 10^{-5}$ | | $6.21 \cdot 10^{-7}$ | |
| 2.0 | $1.57 \cdot 10^{-5}$ | 12 | $2.45 \cdot 10^{-7}$ | 48 |
| | $3.72 \cdot 10^{-6}$ | | $1.53 \cdot 10^{-7}$ | |
| 2.5 | $2.47 \cdot 10^{-5}$ | 12 | $4.14 \cdot 10^{-7}$ | 36 |
| | $1.28 \cdot 10^{-5}$ | | $3.02 \cdot 10^{-7}$ | |
| 3.0 | $6.84 \cdot 10^{-5}$ | 12 | $8.09 \cdot 10^{-7}$ | 48 |
| | $1.95 \cdot 10^{-5}$ | | $3.79 \cdot 10^{-7}$ | |
| 3.5 | $2.66 \cdot 10^{-6}$ | 12 | $2.32 \cdot 10^{-7}$ | 84 |
| | $2.69 \cdot 10^{-5}$ | | $4.17 \cdot 10^{-7}$ | |

In all presented examples *hmin* was equal to $10^{-6}$.

The procedure *diffsystrunkut4* has been verified on the Odra 1204 computer (with 38-bit mantissa) in the Institute of Computer Science of the University of Wrocław.

## References

[1] J. S. Chomicz, *Rozwiązanie liniowego zagadnienia brzegowego metodą kollokacji*, Report no. N-65, Institute of Computer Science, University of Wrocław, 1979.

[2] V. I. Krylov, V. V. Bobkov and P. I. Monastyrnyĭ (В. И. Крылов, В. В. Бобков и П. И. Монастырный), *Вычислительные методы высшей математики*, том 2, Минск 1975.

[3] J. Stoer and R. Bulirsch, *Einführung in die numerische Mathematik, II*, Berlin 1973.

[4] *Systemy programowania maszyny cyfrowej Odra 1204*, Zeszyt 1204-VIII-11, Mera-Elwro, Wrocław 1974.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

*Received on 7. 3. 1978;*
*revised version on 4. 4. 1979*

---

J. S. CHOMICZ, A. OLEJNICZAK i M. SZYSZKOWICZ (Wrocław)

## METODA ZNAJDOWANIA KROKU CAŁKOWANIA UKŁADU RÓWNAŃ RÓŻNICZKOWYCH ZWYCZAJNYCH

STRESZCZENIE

W pracy podana jest metoda znajdowania kroku całkowania zagadnienia początkowego (1.1). Jest to uogólnienie na przypadek układu równań różniczkowych pierwszego rzędu metod podanych przez Kryłowa ([2], str. 103-107) oraz Stoera ([3], str. 114-118). Po prostych modyfikacjach metodę można stosować także do niektórych innych zagadnień [1].

W pracy zamieszczono procedurę realizującą przedstawioną metodę oraz wyniki testów wykonanych na m.c. Odra 1204 w Instytucie Informatyki Uniwersytetu Wrocławskiego.

---