ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XIX, 1 (1987), pp. 149-156

M. SZYSZKOWICZ (Wrocław)

TWO ONE-STEP METHODS TO SOLVE THE INITIAL VALUE PROBLEM

1. Description of the problem. The system of ordinary differential equations

(1) $$y' = f(x, y), \quad y(a) \text{ given},$$

is solved by a one-step method.

The one-step method is realized with automatic step size control. The situation when the step-length depends on the tolerance or on the interval of absolute stability may be recognized by the algorithm. In this paper the procedure *sodel s* (in ALGOL 60) which realizes two one-step methods of the first order and two-stages recognizes this situation automatically.

2. Description of the method. Two one-step methods Φ_1 and Φ_2 of the first order and two-stages are used. The method Φ_1 is given by the formulae

$$\Phi_1: \quad \begin{aligned} y_{n+1/3} &= y_n + (h/3) f_n, \\ y_{n+1} &= y_n + h f_{n+1/3}. \end{aligned}$$

The method Φ_2 is given by the formulae

$$\Phi_2: \quad \begin{aligned} y_{n+1/3} &= y_n + (h/3) f_n, \\ y_{n+1} &= y_n + (h/2)(f_n + f_{n+1/3}). \end{aligned}$$

If Φ_1 and Φ_2 are applied to the test equation

$$y' = \lambda y, \quad y(0) \text{ given}, \quad \lambda \in C$$

with constant step h, then these methods yield a numerical solution $\{y_n\}$ which satisfies a recurrence relation of the following form

$$y_{n+1} = w(z) y_n,$$

where $z = h\lambda$ and $w(z) - e^z = O(z^2)$.

For the method Φ_1 we have

$$w(z) = 1+z+z^2/3$$

and the interval of absolute stability is $(-3.0, 0)$.

The method Φ_2 has the interval of absolute stability $(-6.0, 0)$ and the polynomial is

$$w(z) = 1+z+z^2/6.$$

In the procedure *sodels* the method Φ_1 is mainly used. The method Φ_1 is realized with the step h (solution y_{n+1}) and with the step $h/2$ (solution $\bar{y}_{n+1}$). For $\bar{y}_{n+1}$ we get

$$\bar{y}_{n+1} = w^2(z/2)\, y_n.$$

Richardson's extrapolation is applied and the new numerical solution y^*_{n+1} is obtained

$$y^*_{n+1} = w^*(z)\, y_n,$$

where

$$w^*(z) = 2w^2(z/2) - w(z)$$

and (see [2])

$$w^*(z) - e^z = O(z^4).$$

For the method Φ_2 we have

$$w^*(z) - e^z = O(z^3).$$

The solutions obtained by the method Φ_2 are "constructed" without computation of the function $f(x, y)$. The realization of the methods Φ_1 and Φ_2 together requires 5 evaluations of the function $f(x, y)$ in the interval $[x_n, x_n+h]$. The step size control mechanism (see [1]) is applied, two step sizes h and two solutions y^*_{n+1} (one obtained from Φ_1 and another from Φ_2) are proposed. The method which gives the greater h is realized in the current step of integration.

3. Numerical example. The initial value problem has the form

$$y_1' = 10\,\mathrm{sgn}\sin(20x)\, y_2, \qquad y_1(0) = 0,$$
$$y_2' = -10\,\mathrm{sgn}\sin(20x)\, y_1, \qquad y_2(0) = 1.$$

The exact solution of this problem are the following functions:

$$y_1(x) = |\sin 10x|, \qquad y_2(x) = |\cos 10x|.$$

Below, in Table 1, the results obtained by the methods Φ_1 and Φ_2 are given. In Table 2 the results obtained by the method $\Phi_1+\Phi_2$ (the procedure *sodels*) are presented. As results the relative error $(y_n-y(x_n))/y(x_n)$ and the number of function evaluations of $f(x, y)$ ($[f]$) are given.

TABLE 1

x	Φ_1 $\varepsilon={}_{10}-4$	$[f]$	Φ_2 $\varepsilon={}_{10}-4$	$[f]$
1.0	$-7.64_{10}-4$	3978	$-6.66_{10}-4$	3346
	$-4.13_{10}-4$		$-1.46_{10}-4$	

TABLE 2

x	$\Phi_1+\Phi_2$ $\varepsilon={}_{10}-4$	$[f]$
1.0	$1.79_{10}-3$	1598
	$-2.70_{10}-4$	

4. Algol procedure. The procedure *sodels* has the following parameters.

Data:

x – value *a* in (1),
x1 – value of the argument for which the problem (1) is solved,
eps – relative error (the given tolerance),
eta – numer which is used instead of zero obtained as solution,
h – initial step-size,
hmin – minimum allowed step-size,
n – number of differential equations,
y[1:n] – vectòr with the initial data $y(a)$ in (1).

Results:

x – value of *x1*,
h – step-size to the next integration,
y[1:n] – vector with the solution at point *x1*.

Additional parameters:

steph – label outside of the body of the procedure *sodels* to which a jump is made if $|h|<hmin$ or $sgn(h)\times sgn(x1-x)<0$; increasing *eps* or decreasing *hmin* it is possible to continue the computations,
f – procedure with the heading: **procedure** $f(x, n, y, d)$; **value** x, n; **real** x; **integer** n; **array** y, d; which computes the right-hand side of (1) and assigns them to $d[1:n]$.

```
procedure sodels(x,x1,eps,eta,h,hmin,n,y,steph,f);
 value x1,eps,eta,hmin,n;
 real x,x1,eps,eta,h,hmin;
 integer n;
 array y;
 label steph;
 procedure f;
 begin
  real hh,hhh,wh,ws,ws1,ww,w1,w2,w3;
  integer i;
  Boolean last,stab;
  array d1,ys1,ys2,y1,y2,y3,y4[1:n];
  if abs(h)<hmin∨h×sign(x1-x)≤0
   then go to steph;
  last:=false;
  if abs(h)≥abs(x1-x)
   then
   begin
    last:=true;
    h:=x1-x
   end h>(x1-x);
  eps:=.5/eps;
  f(x,n,y,d1);
conth:
  hh:=.5×h;
  hhh:=.5×hh;
  w3:=h/3.0;
  ww:=.5×w3;
  for i:=1 step 1 until n do
   begin
```

```
  w1:=y[i];
  w2:=d1[i];
  y1[i]:=w1+w3×w2;
  y2[i]:=w1+ww×w2
 end i;
f(x+w3,n,y1,y3);
for i:=1 step 1 until n do
 begin
  w1:=y[i];
  w2:=y3[i];
  y1[i]:=w1+h×w2;
  ys1[i]:=w1+hh×(d1[i]+w2)
 end i;
f(x+ww,n,y2,y3);
for i:=1 step 1 until n do
 y2[i]:=y[i]+hh×y3[i];
f(x+hh,n,y2,y3);
for i:=1 step 1 until n do
 begin
  w1:=y2[i];
  w2:=y3[i];
  y4[i]:=w1+ww×w2;
  ys2[i]:=w1+hhh×w2
 end i;
f(x+hh+ww,n,y4,y3);
ws:=ww:=.0;
for i:=1 step 1 until n do
 begin
  ws1:=y3[i];
  w2:=y2[i]+hh×ws1;
```

```
    w3:=w2-y1[i];
    w1:=y3[i]:=w2+w3;
    w3:=abs(w3);
    w1:=abs(w1);
    if w1<eta
     then w1:=eta;
    w1:=w3/w1;
    if w1>ww
     then ww:=w1;
    w2:=ys2[i]+hhh×ws1;
    w3:=w2-ys1[i];
    w1:=ys1[i]:=w2+w3;
    w1:=abs(w1);
    w3:=abs(w3);
    if w1<eta
     then  w1:=eta;
    w1:=w3/w1;
    if w1>ws
     then  ws:=w1
  end i;
stab:=ws<ww;
if stab
 then ww:=ws;
ww:=if ww=0 then eta else sqrt(eps×ww)×1.25;
hh:=h/ww;
if ww>1.25
 then
 begin
  if abs(hh)<hmin
   then go to steph;
```

```
     last:=false
    end ww>1.25
    else
    begin
     x:=x+h;
     if stab
      then
      begin
       for i:=1 step 1 until n do
        y[i]:=ys1[i]
      end
      else
      for i:=1 step 1 until n do
       y[i]:=y3[i];
     if last
      then go to endp;
     f(x,n,y,d1);
     w1:=x1-x;
     wh:=h;
     if (w1-hh)×sign(h)<0
      then
      begin
       hh:=w1;
       last:=true
      end (w1-hh)×h<0
    end ww<1.25;
  h:=hh;
  go to conth;
endp:
  h:=wh
 end sodels;
```

References

[1] J. Chomicz, A. Olejniczak, M. Szyszkowicz, *A method for finding the step size integration of a system of ordinary differential equations*, Zastos. Mat. 17 (1983), p. 645-653.

[2] M. Szyszkowicz, *Metody jednokrokowe o podwyższonym rzędzie dokładności*, Report N-118, Inst. Comp. Sci., Wrocław University, January 1983.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 1982.05.21;
revised version on 1983.12.09