ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVII, 4 (1983)

ALGORITHMS 85-86

M. SZYSZKOWICZ (Wrocław)

TWO REALIZATIONS OF THE TRAPEZOIDAL METHOD FOR SOLVING THE INITIAL VALUE PROBLEM

1. Procedure declarations. Procedures *diffsystheun* and *diffsystheun2* solve numerically the following initial value problem:

$$y' = f(x, y), \quad x \in [a, b], \tag{1}$$

$$y(a) = y_0, \tag{2}$$

where $y = [y_1(x), y_2(x), \ldots, y_n(x)]^T$ and y_0 is given. Both procedures have the same formal parameters.

Data:

x0 — the value of a in (2);
x1 — the value of the argument for which we solve the problem;
eps — the relative error of the solutions;
eta — the number which is used instead of 0 obtained in the solution;
hmin — the least absolute value of the step length h;
n — the number of equations in (1)-(2);
y0[1 : n] — the values of the right-hand sides of (2).

Results:

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

Additional parameters:

notacc — label outside of the body of the procedure to which a jump is made if the absolute value of the step length is smaller than *hmin*. In this case, *x0* is equal to the value of x ($x0 < x < x1$) for which the approximate solution has a relative error equal to the given *eps* and *y0[1 : n]* contains the values of this approximate solution. By increasing *eps* or decreasing *hmin* one may continue the computations.

f — the name of procedure **procedure** $f(x, n, y, d)$; **value** x, n; **real** x; **integer** n; **array** y, d; which computes the values of $d[1 : n]$ of the right-hand sides of (1).

```
procedure diffsystheun(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 d,d0,d1,y1,y2,y3[1:n];
  eps:=.166666666666/eps;
  h:=x1-x0;
  last:=true;
  f(x0,n,y0,d1);
conth:
  hh:=.5×h;
  for i:=1 step 1 until n do
   y1[i]:=y0[i]+h×d1[i];
  f(x0+h,n,y1,d);
  for i:=1 step 1 until n do
   begin
    w1:=d1[i];
    w2:=y0[i];
    y1[i]:=w2+hh×(w1+d[i]);
    y2[i]:=w2+hh×w1
   end i;
  f(x0+hh,n,y2,d);
  ww:=.5×hh;
```

```
for i:=1 step 1 until n do
 y2[i]:=y0[i]+ww×(d1[i]+d[i]);
f(x0+hh,n,y2,d);
for i:=1 step 1 until n do
 y3[i]:=y2[i]+hh×d[i];
f(x0+h,n,y3,d0);
hh:=ww;
ww:=.0;
for i:=1 step 1 until n do
 begin
  w3:=y2[i]+hh×(d0[i]+d[i]);
  w2:=w3-y1[i];
  w4:=y3[i]:=w3+.333333333333×w2;
  w2:=abs(w2);
  w4:=abs(w4);
  if w4<eta
   then w4:=eta;
  w1:=w2/w4;
  if w1>ww
   then ww:=w1
 end i;
ww:=(if ww=0 then eta else (eps×ww)↑.333333333333)×1.25;
hh:=h/ww;
if ww1.25
 then
 begin
  last:=false;
  if abs(hh)<hmin
   then go to notacc
 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;
 f(x0,n,y0,d1);
 w1:=x1-x0;
 if (w1-hh)×h<0
  then
  begin
   hh:=w1;
   last:=true
  end(x1-x0-hh)×h<0
 end ww<1.25;
h:=hh;
go to conth;
end:
 end diffsystheun
```

2. Method used. We use a method which yields a sequence of approximations $\eta_i \approx y(x_i)$ on the set of points $x_{i+1} = x_i + h_i$, $i = 0, 1, \ldots, N$, $x_0 = a$, $x_N = b$, h_i is the step length. The problem (1)-(2) is solved by using the trapezoidal method

$$\eta_{i+1} = \eta_i + h_i f(x_i, \eta_i), \tag{3}$$

$$\eta_{i+1} = \eta_i + \frac{h_i}{2}\big(f(x_i, \eta_i) + f(x_{i+1}, \eta_{i+1})\big). \tag{4}$$

Formula (3) gives the approximate solution with a local error $O(h^2)$ and formula (4) gives the approximate solution with a local error $O(h^3)$.

The realization of formulae (3) and (4) requires two evaluations of the function f for each step length h. Let $\eta(x_i + h_i, h_i)$ and $\eta(x_i + h_i, h_i/2)$

```
procedure diffsystheun2(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,h5,ww,w1,w2,w3,w4;
  integer i;
  Boolean last;
  array d,d0,d1,d2,y1,y2,y3[1:n];
  eps:=.166666666666/eps;
  h:=x1-x0;
  last:=true;
  f(x0,n,y0,d);
conth:
  h5:=.5×h;
  hh:=.5×h5;
  for i:=1 step 1 until n do
   y1[i]:=y0[i]+h×d[i];
  f(x0+h,n,y1,d1);
  for i:=1 step 1 until n do
   begin
    w1:=y0[i];
    w2:=d[i];
    w3:=d2[i]:=d1[i];
    y1[i]:=w1+h5×(w3+w2);
    y2[i]:=w1+h5×w2
   end i;
```

```
f(x0+h5,n,y2,d1);
for i:=1 step 1 until n do
 y2[i]:=y0[i]+hh×(d1[i]+d[i]);
f(x0+h5,n,y2,d0);
for i:=1 step 1 until n do
 y3[i]:=y2[i]+h5×d0[i];
f(x0+h,n,y3,d1);
ww:=.0;
for i:=1 step 1 until n do
 begin
  w3:=y2[i]+hh×(d0[i]+d1[i]);
  w2:=w3-y1[i];
  w4:=y3[i]:=w3+.333333333333×w2;
  w2:=abs(w2);
  w4:=abs(w4);
  w4:=abs(w3+.333333333333×(w3-w1));
  if w4<eta
   then w4:=eta;
  w1:=w2/w4;
  if w1>ww
   then ww:=w1
 end i;
ww:=(if ww=0 then eta else (eps×ww)↑.333333333333)×1.25;
hh:=h/ww;
if ww>1.25
 then
 begin
  last:=false;
  if abs(hh)<hmin
   then go to notacc
```

```
end ww>1.25
else
begin
 x0:=x0+h;
 for i:=1 step 1 until n do
  begin
   w3:=d1[i];
   d[i]:=w3+.333333333333×(w3-d2[i]);
   y0[i]:=y3[i];
  end i;
 if last
  then go to end;
 w1:=x1-x0;
 if (w1-hh)×h<0
  then
  begin
   hh:=x1-x0;
   last:=true
  end (x1-x0-hh)×h<0
end ww<1.25;
h:=hh;
go to conth;
end:
end diffsystheun2
```

denote the approximate solution at the point x_i+h_i calculated for step lengths h_i and $h_i/2$, respectively, by using (3) and (4). We apply Richardson's extrapolation and obtain

$$(5)\qquad \eta(x_i+h_i) = \eta(x_i+h_i,\ h_i/2)+\tfrac{1}{3}\bigl(\eta(x_i+h_i,\ h_i/2)-\eta(x_i+h_i,\ h_i)\bigr)+O(h^4).$$

The method described by formulae (3)-(5) with step-length control was presented in [1]; here it is realized in the form of procedure *diffsystheun*. Formulae (3) and (4) can be written also in the forms

$$\tilde{\eta}_{i+1} = \eta_i + h_i f(x_i, \tilde{\eta}_i), \tag{6}$$

$$\eta_{i+1} = \eta_i + \frac{h_i}{2}\big(f(x_i, \tilde{\eta}_i) + f(x_{i+1}, \tilde{\eta}_{i+1})\big). \tag{7}$$

The order of the local error is the same as in (3) and (4). Formulae (6) and (7) require only one evaluation of the function f per step of integration. Now, we may also use Richardson's extrapolation for the values of the function f:

$$f(x_i+h_i) = f(x_i+h_i, \tilde{\eta}_{i+1}, h_i/2) + \\ + \tfrac{1}{3}\big(f(x_i+h_i, \tilde{\eta}_{i+1}, h_i/2) - f(x_i+h_i, \tilde{\eta}_{i+1}, h_i)\big) + O(h^3), \tag{8}$$

where $f(x_i+h_i, \tilde{\eta}_{i+1}, h_i)$ and $f(x_i+h_i, \tilde{\eta}_{i+1}, h_i/2)$ denote the values of the function f at the point x_i+h_i calculated at $\tilde{\eta}(x_i+h_i, h_i)$ and $\tilde{\eta}(x_i+h_i, h_i/2)$, respectively.

Procedure *diffsystheun2* is an implementation of formulae (6) and (7) and formulae (5) and (8) in which for the first step of integration we use (3) and (4) with step length $h_i/2$.

3. Certification. The algorithms were tested on the Odra 1204 computer with 37-bit mantissa. In each example we set *hmin* $= {}_{10}{-15}$ and *eta* $=$ *eps*.

Results for example (A)

x	*diffsystheun*		*diffsystheun2*	
	eps $= {}_{10}{-9}$	number of evaluations of f	*eps* $= {}_{10}{-9}$	number of evaluations of f
0.5	$-2.11_{10}{-10}$ $-4.79_{10}{-11}$	1,089	$-2.29_{10}{-9}$ $2.39_{10}{-11}$	873
1.0	$-8.56_{10}{-11}$ $-3.95_{10}{-10}$	1,089	$-1.07_{10}{-10}$ $-2.76_{10}{-10}$	873
1.5	$4.15_{10}{-10}$ $-1.22_{10}{-9}$	1,089	$-2.59_{10}{-10}$ $-6.84_{10}{-10}$	873
2.0	$1.18_{10}{-9}$ $-2.69_{10}{-9}$	1,089	$-1.89_{10}{-10}$ $-1.61_{10}{-9}$	877
4.0	$4.77_{10}{-9}$ $-6.72_{10}{-9}$	4,344	$3.46_{10}{-9}$ $-6.03_{10}{-9}$	3,477
10.0	$1.84_{10}{-8}$ $-2.42_{10}{-8}$	13,018	$2.29_{10}{-8}$ $-2.78_{10}{-8}$	10,417

Examples.

$$\text{(A)}\quad \begin{cases} y_1' = 1/y_2, & y_1(0) = 1, \quad y_1 = e^x, \\ y_2' = -1/y_1, & y_2(0) = 1, \quad y_2 = e^{-x}. \end{cases}$$

$$\text{(B)}\quad \begin{cases} y_1' = -y_1, & y_1(0) = 1, \quad y_1 = e^{-x}, \\ y_2' = -y_2^2, & y_2(0) = 1, \quad y_2 = 1/(1+x). \end{cases}$$

$$\text{(C)}\quad \begin{cases} y_1' = 10\,\mathrm{sgn}(\sin 20x)\,y_2, & y_1(0) = 0, \quad y_1 = |\sin 10x|, \\ y_2' = -10\,\mathrm{sgn}(\sin 20x)\,y_1, & y_2(0) = 1, \quad y_2 = |\cos 10x|. \end{cases}$$

Results for example (B)

x	*diffsystheun*		*diffsystheun2*	
	eps = $_{10}-9$	number of evaluations of f	*eps* = $_{10}-9$	number of evaluations of f
0.5	$-3.11_{10}-10$	1,014	$-4.55_{10}-10$	813
	$-3.49_{10}-10$		$-4.36_{10}-10$	
1.0	$-4.94_{10}-10$	869	$-9.69_{10}-10$	697
	$-5.16_{10}-10$		$-8.07_{10}-10$	
1.5	$-8.80_{10}-10$	869	$-1.92_{10}-9$	697
	$-4.18_{10}-10$		$-4.91_{10}-10$	
2.0	$-1.04_{10}-9$	869	$-2.31_{10}-9$	697
	$-6.33_{10}-10$		$-6.54_{10}-10$	
4.0	$-1.26_{10}-9$	3,513	$-2.97_{10}-9$	2,797
	$-5.09_{10}-10$		$-4.72_{10}-10$	
10.0	$-9.99_{10}-9$	10,338	$-9.19_{10}-9$	8,273
	$-2.92_{10}-9$		$3.28_{10}-9$	

Results for example (C)

x	*diffsystheun*		*diffsystheun 2*	
	eps = $_{10}-3$	number of evaluations of f	*eps* = $_{10}-3$	number of evaluations of f
0.5	$-8.05_{10}-4$	890	$-1.30_{10}-3$	1089
	$-8.48_{10}-4$		$-1.59_{10}-3$	
1.0	$-1.77_{10}-3$	868	$-2.80_{10}-3$	989
	$-1.72_{10}-3$		$-2.78_{10}-3$	
1.5	$-2.64_{10}-3$	988	$-4.19_{10}-3$	881
	$-2.64_{10}-3$		$-4.23_{10}-3$	

Reference

[1] M. Szyszkowicz, *Two algorithms for solving the initial value problem by using Runge-Kutta-Heun's method*, Report N-53, Institute of Computer Science, University of Wrocław, 1978.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 22. 2. 1979

ALGORYTMY 85-86

M. SZYSZKOWICZ (Wrocław)

DWIE REALIZACJE METODY TRAPEZÓW DLA ROZWIĄZYWANIA ZAGADNIENIA POCZĄTKOWEGO

STRESZCZENIE

Procedury *diffsystheun* i *diffsystheun2* o tych samych nagłówkach rozwiązują w sposób numeryczny zagadnienie początkowe (1)-(2) dla danych $y = [y_1(x), y_2(x), \ldots, y_n(x)]^T$ oraz y_0.

Dane:

x0 — wartość *a* w (2);

x1 — wartość argumentu, dla którego zagadnienie ma być rozwiązane;

eps — błąd względny rozwiązań;

eta — liczba zastępująca zero w rozwiązaniu;

hmin — najmniejsza dopuszczalna wartość kroku *h*;

n — liczba równań w (1)-(2);

y0[*1* : *n*] — wartości prawych stron równań w (2).

Wyniki:

x0 — wartość *x1*;

y0[*1* : *n*] — wartości rozwiązania przybliżonego $y_k(x1)$ $(k = 1, 2, \ldots, n)$.

Inne parametry:

notacc — etykieta poza treścią procedury, do której się skacze, gdy wartość bezwzględna długości kroku jest mniejsza od *hmin*. W tym przypadku, *x0* równa się wartości *x* $(x0 < x < x1)$, dla której rozwiązanie przybliżone ma błąd względny równy danemu *eps*, a *y0*[*1* : *n*] zawiera wartości tego rozwiązania przybliżonego. Zwiększając *eps* lub zmniejszając *hmin* można kontynuować obliczenia.

f — nazwa procedury o nagłówku **procedure** $f(x, n, y, d)$; **value** x, n; **real** x; **integer** n; **array** y, d; która oblicza wartości $d[1:n]$ prawych stron równań w (1).

W algorytmach posłużono się metodą trapezów, a w trakcie obliczeń dobierano krok całkowania. Procedury sprawdzono na m. c. Odra 1204, a wyniki obliczeń przedstawiono w tablicach.
