ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XIX, 1 (1987), pp. 139–147

M. SZYSZKOWICZ (Wrocław)

THE APPLICATION OF A CLASS OF ONE-STEP METHODS TO SOLVE THE INITIAL VALUE PROBLEM

A method for the numerical solution of initial value problems with different step sizes is described. A procedure which realizes this method is also given.

1. Introduction. For the numerical solution of the initial value problem for the system of ordinary differential equations

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0, \quad y, y_0 \in R^s, \quad s \geqslant 1, \tag{1.1}$$

we use a one-step method.

The realization of any one-step method may have the following form

$$\begin{aligned} \eta_0 &:= y_0, \\ \eta_{i+1} &:= \eta_i + h_i \Phi(x_i, \eta_i, h_i), \\ x_{i+1} &:= x_i + h_i, \end{aligned} \qquad i = 0, 1, 2, \ldots,$$

where $\Phi(x, y, h)$ is the increment function of the one-step method and η_i, $\Phi(x_i, \eta_i, h_i) \in R^s$.

Here, for the numerical solution of the problem (1.1) we use the one-step methods which were given by Bobkov (see Krylov et al. [2]). The Bobkov method is described by the parameters A_i, α_i $(i = 0, 1, \ldots, q)$ and some formulae for the approximation of the solutions at knots α_i. The parameters A_i, α_i $(i = 0, 1, \ldots, q)$ are obtained from the condition

$$y(x+h) - y(x) \approx h \sum_{i=0}^{q} A_i y'(x+\alpha_i h) = h \sum_{i=0}^{q} A_i f_{n+\alpha_i}, \tag{1.2}$$

where equality holds up to the terms with h^p, and p is the order of the one-step method. In what follows $\tilde{f}_{n+1} = f(x+h, \tilde{\eta}_{n+1})$ and $\tilde{\eta}_{n+1}$ is not the final numerical solution of (1.1) at point $x+h$, i.e. for $\tilde{\eta}_{n+1}$ we have $y(x+h) = \tilde{\eta}_{n+1} + O(h^p)$.

For example, the second order Bobkov method may have the form

$$\tilde{\eta}_{n+1} = \eta_n + hf_n,$$

$$\eta_{n+1} = \eta_n + (h/2)(f_n + \tilde{f}_{n+1}),$$

where $A_0 = A_1 = \frac{1}{2}$ and $\alpha_0 = 0$, $\alpha_1 = 1$.

This method is equivalent to the Heun method.

2. A method of step size control. The presented method may be applied to the problem (1.1) with $s > 1$, but the formulae below are given for one differential equation only.

From (1.2) we see that the local truncation error of the Bobkov method has the following form

$$r = h^{p+1} y^{(p+1)}(x)\left[\frac{1}{(p+1)!} - \frac{1}{p!}\sum_{i=0}^{q} A_i \alpha_i^p\right] + O(h^{p+2}),$$

where p is the order of the method.

Let

$$\gamma = \frac{1}{(p+1)!} - \frac{1}{p!}\sum_{i=0}^{q} A_i \alpha_i^p.$$

We see that γ depends on the parameters A_i, α_i ($i = 0, 1, \ldots, q$). If we use the one-step method for the numerical solution of (1.1) with the step size h, we have

$$y(x+h) - \eta(x+h, h) \doteq C(x) h^{p+1},$$

where $\eta(x+h, h)$ is the numerical solution obtained with the step size h and p is the order of this method.

After application of the same one-step method with the step size $h/2$ we have [1]

$$y(x+h) - \eta(x+h, h/2) \doteq 2C(x)(h/2)^{p+1}. \tag{2.1}$$

In a typical situation the step size control mechanism is made on the basis of the solutions $\eta(x+h, h)$ and $\eta(x+h, h/2)$. In the paper [1] the following algorithm of the step size control is given

$$h_{\text{new}} := h_{\text{old}}/w, \tag{2.2}$$

where

$$w := 1.25 \sqrt[p+1]{\frac{1}{2(2^p-1)} \max_{1 \leqslant k \leqslant s} \frac{|\eta_k(x+h, h) - \eta_k(x+h, h/2)|}{|\eta_k^*(x+h)| \cdot eps}} \tag{2.3}$$

and

$$\eta^*(x+h) := \eta(x+h, h/2) + \frac{\eta(x+h, h/2) - \eta(x+h, h)}{2^p - 1}. \tag{2.4}$$

The constant 1.25 in (2.3) has been chosen experimentally and gives a safe algorithm.

Now we use the same mechanism for the step size control as was described in [1] and is given above in short form. However, we obtain $\eta(x+h, h/2)$ in a different way. By a special choice of the coefficient γ we may simulate calculation with the step size $h/2$.

If we have the Bobkov method which satisfies

$$y(x+h)-\eta_{n+1} \doteq \gamma y^{(p+1)}(x)\, h^{p+1},$$

we may find another method of the same order which gives

$$y(x+h)-\bar{\eta}_{n+1} \doteq 2\gamma y^{(p+1)}(x)\left(\frac{h}{2}\right)^{p+1} = \frac{\gamma}{2^p}\, y^{(p+1)}(x)\, h^{p+1}.$$

The obtained solution $\bar{\eta}_{n+1}$ may be treated as $\eta(x+h, h/2)$ in (2.1) and the formulae (2.2)–(2.4) may be used with η_{n+1} and $\bar{\eta}_{n+1}$.

The solution at knots α_i ($i = 0, 1, \ldots, q$) must be obtained with local error of order not less than p. For example, consider the second order method. We obtain the parameters of the method from the system

$$\sum_{i=0}^{q} A_i = 1, \qquad \sum_{i=0}^{q} A_i \alpha_i = \tfrac{1}{2},$$

which is formed from (1.2) and also we have the relationship

$$\sum_{i=0}^{q} A_i \alpha_i^2 = \tfrac{1}{3}-2\gamma.$$

Let $q = 1$ and $\alpha_0 = 0$, then we obtain

$$\alpha_0 = 0, \quad A_0 = \frac{1-24\gamma}{4(1-6\gamma)}, \qquad \alpha_1 = \tfrac{2}{3}(1-6\gamma), \quad A_1 = \frac{3}{4(1-6\gamma)}.$$

To have a one-step character of the method we must take $0 < \alpha_1 \leqslant 1$ and from this restriction we obtain

$$-1/12 \leqslant \gamma < 1/6.$$

Instead of $\eta_{n+\alpha}$ we use $y_{n+\alpha}$ in the following formulae. Taking $\gamma = -1/12$ we have the formula

(2.5)
$$y_{n+1} = y_n+(h/2)(f_n+\bar{f}_{n+1}).$$

Now we have to obtain a formula with $\gamma = -\frac{1}{48}$, i.e.

(2.6)
$$y_{n+1} = y_n+(h/3)(f_n+2f_{n+3/4}).$$

This formula is the same as (2.5) used with the step size $h/2$ (with respect to the first term of the error). For the calculations of $\bar{f}_{n+1}, f_{n+3/4}$ with order not less than 2 we may apply for (2.5) the formulae

$$y_{n+1/2} = y_n + (h/2) f_n,$$

$$\tilde{y}_{n+1} = y_n + h f_{n+1/2},$$

and for (2.6) the formulae

$$y_{n+3/8} = y_n + \tfrac{3}{8} h f_n,$$

$$y_{n+3/4} = y_n + \tfrac{3}{4} h f_{n+3/8}.$$

Of course, we may apply here also other formulae.

In a general situation we may use the formulae

$$y_{n+\alpha/4} = y_n + (\alpha/4) h f_n,$$

$$y_{n+\alpha/2} = y_n + (\alpha/2) h f_{n+\alpha/4},$$

$$y_{n+\alpha} = y_n + \alpha h f_{n+\alpha/2},$$

(2.7) $$y_{n+1} = y_n + A_0 h f_n + A_1 h f_{n+\alpha/2}$$

and

(2.8) $$y_{n+1} = y_n + B_0 h f_n + B_1 h f_{n+\alpha}.$$

In this paper the procedure *diffsysthek* (in ALGOL 60) which realizes the formulae (2.7) and (2.8) with $\alpha = 1$ is presented. The new step size h is computed from the formulae (2.2)–(2.4). It is not necessary to obtain a solution with γ and $\gamma/2^p$, one may also use different values of γ and γ_0.

When we know γ and γ_0 we may also apply Richardson's extrapolation to obtain a better solution. Heun's method uses formula (2.4).

3. Numerical experiments. We have tested our procedure for the problems

(A) $y_1' = 1/y_2, \quad y_1(0) = 1, \quad y_1 = e^x,$
$\quad\; y_2' = -1/y_1, \quad y_2(0) = 1, \quad y_2 = e^{-x};$

(B) $y' = 10 \cos 10x, \quad y(0) = 0, \quad y = \sin 10x;$

(C) $y_1' = 10 \operatorname{sgn} \sin(20x) y_2, \quad y_1(0) = 0, \quad y_1 = |\sin 10x|,$
$\quad\; y_2' = -10 \operatorname{sgn} \sin(20x) y_1, \quad y_2(0) = 1, \quad y_2 = |\cos 10x|.$

The calculations were made for $eps = eta = {}_{10}-3,\ {}_{10}-6,\ {}_{10}-9$ (for problem (C) only for $eps = {}_{10}-3$) at points $x = 0.5, 1.0, 1.5, 10.0$. Tables 1–4 present the relative error $(y_n - y(x))/y(x)$ and the number of evaluations of the function f ([f]) at points $x = 1.5$ and $x = 10.0$.

TABLE 1. Problem (A) (*diffsysthek* procedure)

x	${}_{10}-3$	$[f]$	${}_{10}-6$	$[f]$	${}_{10}-9$	$[f]$
1.5	$-2.7_{10}-4$	4	$-1.4_{10}-7$	31	$5.1_{10}-11$	255
	$2.5_{10}-4$		$1.3_{10}-7$		$9.7_{10}-11$	
10.0	$-2.3_{10}-3$	54	$-2.5_{10}-6$	442	$-6.2_{10}-10$	4266
	$2.0_{10}-3$		$-2.4_{10}-6$		$6.6_{10}-10$	

TABLE 2. Problem (A) (Heun method)

x	${}_{10}-3$	$[f]$	${}_{10}-9$	$[f]$
1.5	$-6.92_{10}-5$	19	$4.15_{10}-10$	1089
	$-4.85_{10}-4$		$-1.22_{10}-9$	
10.0	$1.91_{10}-2$	148	$1.94_{10}-8$	13018
	$-2.95_{10}-2$		$-2.42_{10}-8$	

TABLE 3. Problem (B) (*diffsysthek* procedure)

x	${}_{10}-3$	$[f]$	${}_{10}-6$	$[f]$	${}_{10}-9$	$[f]$
1.5	$-1.3_{10}-4$	27	$1.6_{10}-8$	255	$1.0_{10}-9$	2527
10.0	$1.0_{10}-2$	447	$6.2_{10}-7$	4912	$-2.4_{10}-8$	49059

TABLE 4. Problem (C)

	(*diffsysthek* procedure)		(Heun method)	
x	${}_{10}-3$	$[f]$	${}_{10}-3$	$[f]$
1.5	$-1.3_{10}-1$	129	$-2.64_{10}-3$	988
	$8.0_{10}-2$		$2.64_{10}-3$	
10.0	$7.9_{10}0$	1113	$-1.65_{10}-2$	11648
	$4.4_{10}-1$		$-1.65_{10}-2$	

4. Conclusions. In the same way as is described in Section 2 we may obtain methods with different values of parameter γ. We may "simulate" the calculations with different step size sequences, for example with $h/2$, $h/3, \ldots,$ but with respect to the first term of the error only.

5. Description of procedure *diffsysthek*

Procedure declaration. The procedure *diffsysthek* solves the initial value problem of the form

(1) $$y_k' = f_k(x, y_1(x), y_2(x), \ldots, y_n(x)),$$

(2) $$y_k(x_0) = y_{0k} \quad (k = 1, 2, \ldots, n)$$

at the points $x_1, x_2, \ldots$

Data:

x0	– value of x_0 at (2),
x1	– value of the argument for which we solve the problem (1), (2),
eps	– relative error (given tolerance),
eta	– number which is used instead of zero if the obtained solution is zero or near to zero; this number is used to compute the relative error,
hmin	– least absolute value of the step size,
n	– number of differential equations in (1),
y0 [*1* : *n*]	– values of the right-hand sides of (2).

Results:

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

Additional parameters:

notacc	– label outside of the body of procedure *diffsysthek* to which a jump is made if the absolute value of the step size is smaller than *hmin*; the array *y0* [*1* : *n*] contains the values at a point *x*, where $x0 < x < x1$,
f	– identifier of the procedure which computes the values of the right-hand sides of (1) and puts them in *d* [*1* : *n*] and which has the following heading: **procedure** *f* (*x*, *n*, *y*, *d*); **value** *x*, *n*; **real** *x*; **integer** *n*; **array** *y*, *d*.

References

[1] J. Chomicz, A. Olejniczak, M. Szyszkowicz, *Dobór kroku obliczeń dla metod jednokrokowych*, Raport nr N-35, Instytut Informatyki Uniwersytetu Wrocławskiego, Wrocław 1978.

[2] V. I. Krylov, V. V. Bobkov and P. I. Monastyrnyĭ, *Numerical methods in higher mathematics* (in Russian), Vol. 2, Moscow 1977.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 1981.01.09;
revised version on 1982.03.11

```
procedure diffsysthek(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,w3,w4;
  integer i;
  Boolean last;
  array d,y,yf[1:n];
  eps:=.008/eps;
  h:=x1-x0;
  last:=true;
  f(x0,n,y0,yf);
conth;
  hh:=.25×h;
  for i:=1 step 1 until n do
   y[i]:=y0[i]+hh×yf[i];
  f(x0+hh,n,y,d);
  hh:=hh+hh;
  for i:=1 step 1 until n do
   y[i]:=y0[i]+hh×d[i];
  f(x0+hh,n,y,d);
  for i:=1 step 1 until n do
   y[i]:=y0[i]+h×d[i];
  f(x0+h,n,y,d);
  ww:=.0;
```

```
for i:=1 step 1 until n do
 begin
  w3:=y[i];
  w4:=y0[i]+hh×(d[i]+yf[i]);
  w4:=w4-w3;
  w3:=y[i]:=w3+.333333333333×w4;
  w4:=abs(w4);
  w3:=abs(w3);
  if w3<eta
   then w3:=eta;
  w3:=w4/w3;
  if w3>ww
   then ww:=w3
 end i;
ww:=if ww=0 then eta else 1.25×(eps×ww)↑.333333333333;
hh:=h/ww;
if ww>2.5
 then
 begin
  if abs(hh)<hmin
   then go to notacc;
  last:=false;
 end ww>2.5
 else
 begin
  x0:=x0+h;
  for i:=1 step 1 until n do
   y0[i]:=y[i];
  if last
```

```
      then go to endp;
     f(x0,n,y0,yf);
     w3:=x1-x0;
     if (w3-hh)×h<0
      then
      begin
       hh:=w3;
       last:=true
      end (w3-hh)×h<0
    end ww<2.5;
   h:=hh;
   go to conth;
 endp:
  end diffsysthek;
```