ALGORITHM 9 RUNGE-KUTTA INTEGRATION
3 (May 1960), 318 F. NAUR

procedure RK(x,y,n,FKTEpsEta,E,E,F,Fi) ; value x,y
integer n ; Boolean Fi ; real x,EpsEta,E,E
array y,E,E ; procedure FKTEpsEta
comment : RK integrates the system y'_k = f_k(x,y_1,y_2,...,y_n)
(k=1,2,...,n) of differential equations with the method of Runge-
Kutta with automatic search for appropriate length of integration
step. Parameters are: The initial values x and y[k] for x and the
unknown functions y_k(x). The order n of the system. The procedure
FKT(x,y,n,E) which represents the system to be integrated, i.e.
the set of functions f_k. The tolerance values eps and eta which
govern the accuracy of the numerical integration. The end of the
integration interval xE. The output parameter yE which repre-
sents the solution at x=xE. The Boolean variable Fi, which must
always be given the value true for an isolated or first entry into
RK. If however the functions y must be available at several mesh-
points x_0, x_1, ..., x_n, then the procedure must be called repeated-
ely (with x=x_i, xE=x_{i+1}, for k=0,1,...,n-1) and then the
later calls may occur with Fi=false which saves computing time.
The input parameters of FKT must be x=x_i, the output parameter
y[k] represents the set of derivatives y_k[1]=f_1(x,y_1), y_k[2],...,y_k[n]
for x and the actual y_1. A procedure comp enters as a non-local
identifier :

begin
array y[1:n] ; real x[1:2] ; Boolean out
integer j ; own real s,hs
procedure RKIST(x,y,h,hs,yE) ; real x,y,hs
array y,yE ;
comment : RKIST integrates one single RUNGE-KUTTA
with initial values x_0,k which yields the output
parameters x=x+h and y[k], the latter being the
solution at x. Important : the parameters n, FKTEpsEta
enter RKIST as non-local entities

begin
array w[1:n] ; integer j ;
x := x
for k := 1 step 1 until n do
begin
y[k] := y[k]+a[j]*x[k]
yE[k] := yE[k] + a[j+1]*x[k]/3
end
end
end RKIST

Begin of program:
if Fi then begin H := xE-x ; s := 0 end else H := xE ;
out := false ;
AA: if x+2.01xH-xE>0 or (H>0) then
begin Hs := H ; out := true ; H := (xE-x)/2
end if ;
RKIST(x,y,2xH,x_1,y_1)
BB: RKIST(x,y,H,x_2,y_2) ; RKIST(x_2,y_2,H,x_3,y_3)
for k := 1 step 1 until n do
if comp(y_1,k),y_3[k],eta)>eps then go to CC
; comment : comp(a,b,c) is a function designator, the value
of which is the absolute value of the difference of the
mantissa of a and b, after the exponents of these
quantities have been made equal to the largest of the
exponents of the originally given parameters a,b,c
x := x_3 ; if out then go to DD ;
for k := 1 step 1 until n do
y[k] := y_3[k] ;
if s=5 then begin s := 0 ; H := 2xH end if ;
s := s+1 ; go to AA
CC: H := 0.5xH ; out := false ; x_1 := x_2
for k := 1 step 1 until n do
y[k] := y[k] ;
go to BB
DD: for k := 1 step 1 until n do
yE[k] := y[k]
end RKIST

* This RK-program contains some new ideas which are related
to ideas of S. GILL. A process for the step-by-step integration
differential equations in an automatic computing machine,
On the solution of ordinary differential equations with digital com-
puting machines, Fysiograf. Sällsk. Lund, Förh. 20 Nr. 11 (1959)
p. 130-152. It must be clear, however, that with respect to com-
puting time and round-off errors it may not be optimal, nor has it
actually been tested on a computer.

CERTIFICATION OF ALGORITHM 9 [D2]
RUNGE-KUTTA INTEGRATION [P. Naur et al.,
Comm. ACM 3 (May 1960), 318]
HENRY C. TRACHER, Jr. (Recd. 28 July 1964 and 22 Nov.
1965)
Argonne National Laboratory, Argonne, Ill.

Algorithm 9 was transcribed into the hardware representation
for CDC 3600 ALGOL and run successfully. The following procedure
was used for the global procedure comp:
real procedure comp(a,b,c) ; value a,b,c ; real a,b,c
begin integer AE, BE, CE;
integer procedure expos(x) ; real x;
comment : This function produces the base 10 exponent of x;
expos := if x = 0 then -999 else
enter (434294819 X ln(abs(x))+1);
comment : The number -999 may be replaced by any number
less than the exponent of the smallest positive number handled
by the particular machine used, for this algorithm assumes
that true zero has an exponent smaller than any nonzero
floating-point number. Users implementing real procedure
comp by machine code should make sure that this condition
is satisfied by their program;
AE := expos(a) ; BE := expos(b) ; CE := expos(c);
if AE < BE then AE := BE ; if AE < CE then AE := CE ;
comp := abs(a-b)/10 + AE
end
This has the advantage of machine independence, but is highly inefficient compared to machine code.

The procedure was tested using the two following procedures for $FKT$:

procedure $FKT(X, Y, N, Z)$; real $X$; integer $N$; array $Y, Z$;

comment $(dy/dx) = z_1 = y_2$, $(dy_1/dz) = z_2 = -y_1$. With $y(0) = 0, y(0) = 1$, the solution is $y_1 = \sin x$, $y_2 = \cos x$;


procedure $FKT(X, Y, N, Z)$; real $X$; integer $N$; array $Y, Z$;

comment $(dy/dz) = 1 + y^2$. For $y(0) = 0, y(z) = \tan z$;


The $RK$ procedure was used to integrate the differential equations represented by the first $FKT$ procedure from $x = 0(0.5)\pi$, with $\varepsilon = 10^{-4}$, and with $y(0) = 0, y(0) = 1$. The actual step size $h$ was $0.025$ for most of the range, but was reduced to $0.03125$ in the neighborhood of $x = \pi/2$, where one or the other of the solutions is small.

The computed solutions at $x = 7.0$ were: $y_1 = 6.5690902746 \times 10^{-3}, y_2 = 7.3390270246 \times 10^{-1}$, with errors $-5.71 \times 10^{-3}$ and $4.48 \times 10^{-3}$, respectively.

Results for the second differential equation are summarized in Table 1 below.

The efficiency of the procedure would be increased slightly on most computers by changing the type of the own variable $s$ from real to integer.

The error is estimated by comparing the results of successive pairs of steps with that of a single double step. This is somewhat more time-consuming than the Kutta-Merson process presented in Algorithm 218 [Comm. ACM 6 (Dec. 1963) 737-8]. However, the criterion for step-size variation in Algorithm 9 which effectively applies an approximate relative error criterion, $\varepsilon$, for $|y| > \eta s$, and an absolute error criterion $\eta s \times \varepsilon$, for $|y| < \eta s$, appears superior when the solution fluctuates in magnitude.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$h_{min}$</th>
<th>Absolute error</th>
<th>Relative error</th>
<th>$h_{min}$</th>
<th>Absolute error</th>
<th>Relative error</th>
<th>$h_{min}$</th>
<th>Absolute error</th>
<th>Relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>$10^{-2}$</td>
<td>0.03125</td>
<td>$-$1 $\times$ 10$^{-4}$</td>
<td>$-$2 $\times$ 10$^{-9}$</td>
<td>0.03125</td>
<td>9 $\times$ 10$^{-4}$</td>
<td>6 $\times$ 10$^{-4}$</td>
<td>0.03090625</td>
<td>$-$1 $\times$ 10$^{-4}$</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>$10^{-3}$</td>
<td>0.125</td>
<td>$-$5 $\times$ 10$^{-3}$</td>
<td>$-$9 $\times$ 10$^{-3}$</td>
<td>0.0625</td>
<td>8 $\times$ 10$^{-4}$</td>
<td>5 $\times$ 10$^{-4}$</td>
<td>0.0078125</td>
<td>$-$2 $\times$ 10$^{-4}$</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>$10^{-4}$</td>
<td>0.25</td>
<td>$-$1 $\times$ 10$^{-4}$</td>
<td>$-$2 $\times$ 10$^{-4}$</td>
<td>0.25</td>
<td>$-$2 $\times$ 10$^{-4}$</td>
<td>$-$1 $\times$ 10$^{-4}$</td>
<td>0.03125</td>
<td>$-$3 $\times$ 10$^{-4}$</td>
</tr>
</tbody>
</table>