

STICHTING
MATHEMATISCH CENTRUM

2e BOERHAAVESTRAAT 49
AMSTERDAM (O.)

Telefoon ~~51660 en 56643~~

Gem. Giro M. 2138
Postgiro 462 890

Amsterdam, 4 th February 1960

Dr. P. Naur
Regnecentralen
Gl. Carlsbergvej 2
Valby, Denmark

Uw. ref.:

Onze ref.: AvW/CK

Onderwerp:

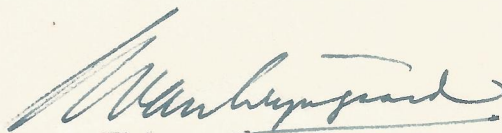
Dear Peter,

Herewith I send you some procedures as examples of what I understood to be Algol 60, but what does not conform completely with Woodger's list.

I might draw your attention to some points which came up on Sunday 17th, and I think a proper rewording of 160 should be obtained by inserting under line 13 (i.e. between "new significance" and "identifiers which"):

, except the formal variables of a procedure declaration, if the block is the procedure compound of that declaration (cf.....). These formal variables are declared to be local for the block by their appearance in the procedure heading.

Sincerely yours,


A. van Wijngaarden

Copies to:

John W. Backus,	New York
McCarthy,	Cambridge
Julien Green,	New York
C. Katz,	Philadelphia
Alan J. Perlis,	Pittsburgh
H. Rutishauser,	Zürich
B. Vauquois,	Grenoble
J. Wegstein,	Washington
M. Woodger,	Teddington
F.L. Bauer,	Mainz
K. Samelson,	Mainz

value eps, tim; integer tim

procedure euler (fct, sum, eps, tim); ~~name (fct, sum);~~

real procedure fct; real sum, eps;

comment euler computes the sum of fct(i) for i from zero upto infinity by means of a suitably refined euler transformation. The summation is stopped as soon as tim times in succession the absolute value of the terms of the transformed series are found to be less than eps. Hence, one should provide a function fct with one integer argument, an upper bound eps and an integer tim. The output is the sum sum. euler is particularly efficient in the case of a slowly convergent or divergent alternating series;

begin

integer ~~i, k, n, t, tim~~;

array ~~m [0:15]~~;

real (mn, mp, ds);

i:=n:=t:=0; m[0]:=fct(0); sum:=m[0]/2;

nextterm: i:=i+1; mn:=fct(i);

for k:=0 step 1 until n do begin

mp:=(mn+m[k])/2; m[k]:=mn;

mn:=mp end means;

if (abs(mn) < abs(m[n])) ^ (n < 15) then begin

ds:=mn/2; n:=n+1; m[n]:=mn end accept

else ds:=mn;

sum:=sum+ds;

if abs(ds) < eps then t:=t+1

else t:=0;

if t < tim then goto nextterm

end

euler;

value n, eps, xE ;

RK 1

```
procedure RK(x,y,n,Fct,eps,xE); name (x,y,Fct);
comment RK integrates differential equation  $y' = Fct(x,y,n)$ ,
where  $y[k]$  ( $k=1, \dots, n$ ) are the unknown functions,  $n$  the
order of the system from  $x$  to  $xE$ . It searches the length
of integration step according to given tolerance  $\text{eps}$ .
After leaving the procedure,  $x,y$  are the corresponding
values at the end point  $xE$  of the interval.

begin
integer {n,j,k,s};
real {x0,x1,x2,eta};
array (y,y1,y2,y3,z[1:n]); value  $h$ ;
procedure RK1ST(x,y,h,xE,ye) name (x,y,xE,ye);
comment RK1ST is subprocedure for one single Runge-Kutta
integration step with given length  $h$ ;

begin
array (y,ye,w[1:n],a[1:5]);
initialize a:=0.5,0.5,1,1,0.5;

xe:=x; TRANSPORT (ye,y); TRANSPORT (w,y);
for j:=1 step 1 until 4 do begin
Fct(xe,w,n,z); xe:=x+a[j]*h;
for k:=1 step 1 until 4 do begin
w[k]:=y[k]+a[j]*h*z[k];
ye[k]:=ye[k]+a[j+1]*h*z[k]/3 end k
end j
end RK1ST;
procedure TRANSPORT (y,Y); name (y,Y);
begin
array (y,Y[1:n]);
for j:=1 step 1 until n do y[j]:=Y[j]
end TRANSPORT

here start the statements of RK;
```



```

      x0:=x; s:=0; eta:=(xE-x)/2;
ENDTEST: if x=xE then goto RETURN;
      if (x+2*eta > xE)≡(xE > x0) then eta:=(xE-x)/2;
      RK1ST (x,y,2*eta,x1,y1);
TWOSTEP: RK1ST (x,y,eta,x2,y2);
      RK1ST (x2,y2,eta,x1,y3);
      for k:=1 step 1 until n do
          if comp(y1[k],y3[k]) > eps then goto HALVE;
comment comp(a,b) is a function which computes from two
given numbers a,b the absolute value of the difference
of their mantissae (of course after the exponents
have been made equal in the usual way). This program
must be available in machine notation;
      x:=x1; TRANSPORT (y,y3); s:=s+1;
      if s=5 then begin s:=0; eta:=2*eta end;
      goto ENDTEST;
HALVE:   eta:=0.5*eta; x1:=x2; TRANSPORT (y1,y2);
      goto TWOSTEP;
RETURN:  end RK;

```