ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVI, 4 (1980)

ALGORITHM 67

H. MALINOWSKI and R. SMARZEWSKI (Lublin)

DETERMINATION OF THE SOLUTION OF ABEL INTEGRAL EQUATIONS, II

1. Procedure declaration. The procedure *inteqAbel2* determines the approximation $f_\Delta(s)$ of the solution $f(s)$ of the integral equations

$$g(t) = \int_0^t \frac{f(s)}{\sqrt{t^2-s^2}}\,ds, \qquad t \in [0, R], \tag{1}$$

and

$$g(t) = \int_t^R \frac{f(s)}{\sqrt{s^2-t^2}}\,ds, \qquad t \in [0, R]. \tag{2}$$

These approximations $f_\Delta(s)$ are defined by

$$f_\Delta(s) = \frac{2}{\pi}\left(g(0)+s\int_0^s \frac{g'_\Delta(t)}{\sqrt{s^2-t^2}}\,dt\right), \qquad s \in (0, R], \tag{3}$$

and

$$f_\Delta(s) = \frac{2s}{\pi}\left(\frac{g(R)}{\sqrt{R^2-s^2}} - \int_s^R \frac{g'_\Delta(t)}{\sqrt{t^2-s^2}}\,dt\right), \qquad s \in (0, R], \tag{4}$$

respectively, where g_Δ is a spline function of degree $m = 2k-1$ $(1 < k \leqslant n)$ interpolating the function g at the nodes t_i, $0 = t_1 < t_2 < \ldots < t_n = R$, and given in the form

$$g_\Delta(t) = \sum_{i=0}^{m} A_i t^i + \sum_{j=1}^{n} B_j \theta(t, t_j)(t-t_j)^m \tag{5}$$

with

$$\theta(t, t_j) = \begin{cases} 0 & \text{if } t < t_j, \\ 1 & \text{if } t \geqslant t_j. \end{cases}$$

```
procedure inteqAbel2(m,n,s,F,R,g,A,B,t,f);
 value m,n,s,R;
 integer m,n;
 real s,R,g,f;
 array A,B,t;
 Boolean F;
 if ¬F∧s=R
  then f:=.0
  else
  begin
   integer i,k,m1,j,v;
   real s2,R2,Rsq,Ri,si,Rp,sum2,tj,tj2,bj,sk,sp1,ci,ci1,tp1,
   h1;
   array h[1:m];
   s2:=s×s;
   if F
    then R2:=Rsq:=R:=.0
    else
    begin
     R2:=R×R;
     Rsq:=sqrt(R2-s2)
    end ¬F;
   h1:=h[1]:=if F then 1.5707963268 else -ln((R+Rsq)/s);
   f:=A[1]×h1;
   Ri:=Rsq×R;
   si:=s;
   for i:=3 step 2 until m do
    begin
     h[i]:=((i-2)×s2×h[i-2]-Ri)/(i-1);
     f:=f+i×h[i]×A[i];
```

```
        Ri:=Ri×R2;
        if F
         then
         begin
          h[i-1]:=si;
          f:=f+si×A[i-1]×(i-1);
          si:=si×s2×(i-1)/i
         end F
      end i;
     if ¬F
      then
      begin
        k:=0;
        Ri:=s2/(R2-s2);
        Rp:=Rsq/Ri;
        for i:=2 step 2 until m do
         begin
          sp1:=.0;
          k:=k+1;
          si:=Ri;
          m1:=i-1;
          for v:=1 step 1 until k do
           begin
            sp1:=sp1+si/m1;
            m1:=m1-2;
            si:=si×Ri×(k-v)/v
           end v;
          h[i]:=-Rp×sp1;
          Rp:=Rp×(R2-s2);
          f:=f+i×A[i]×h[i]
```

```
    end i
   end ¬F;
  sum2:=B[1]×h[m];
  n:=n-1;
  for j:=2 step 1 until n do
   begin
    tj:=t[j];
    if s≤tj∧F
     then go to E1;
    tj2:=tj×tj;
    si:=tj↑(m-1);
    ci1:=-tj/s;
    ci1:=if F then arctan(ci1/sqrt(1.0-ci1×ci1)) else if s<tj
         then -ln(s/(tj+sqrt(tj2-s2))) else .0;
    bj:=si×(h1+ci1);
    si:=-si×(m-1)/tj;
    m1:=1;
    k:=0;
    if F
     then
     begin
      Ri:=sqrt(s2-tj2);
      Rp:=Ri×tj;
      sk:=1.0;
      sp1:=tj2/s2
     end F
     else
      if s<tj
       then
       begin
```

```
      sk:=tj2-s2;
      Ri:=sqrt(sk);
      Rp:=Ri×tj;
      sp1:=s2/sk
     end ¬F∧s<tj;
for i:=2 step 2 until m do
 begin
  if s≥tj∧¬F
   then
   begin
    ci:=h[i];
    ci1:=.0
   end s≥tj∧¬F
   else
   begin
    ci:=.0;
    k:=k+1;
    if F
     then tp1:=1.0/s
     else
     begin
      tp1:=1.0/sk;
      m1:=i-1
     end ¬F;
    for v:=1 step 1 until k do
     begin
      ci:=ci+tp1/m1;
      tp1:=0.5×tp1×sp1×(if F then v+v-1 else i-(v+v))/v;
      if ¬F
       then m1:=m1-2
```

```
                    end v;
            Ri:=Ri×sk;
            ci:=if F then h[i]×ci×Ri else h[i]+ci×Ri;
            ci1:=((i-1)×s2×ci1+Rp)/i;
            Rp:=Rp×tj2
          end ¬(s≥tj∧¬F);
         bj:=bj+si×ci;
         si:=-si×(m-i)/(i×tj);
         bj:=bj+si×(h[i+1]+ci1);
         si:=-si×(m-i-1)/((i+1)×tj)
        end i;
       sum2:=sum2+bj×B[j]
      end j;
  E1:f:=0.63661977236×s×(g/(if F then s else Rsq)+f+m×sum2)
   end F∨s≠R,inteqAbel2
```

Data:

m — degree of the spline function (5);
n — number of nodes of the spline function g_Δ;
s — an arbitrary fixed point from the interval $(0, R]$;
F — Boolean variable; if $F \equiv$ **true**, then equation (1) is solved, otherwise equation (2) is solved;
R — if $F \equiv$ **false**, then R denotes the upper limit of integral (2), otherwise it is inessential;
g — if $F \equiv$ **true**, then g is equal to $g(0)$, else to $g(R)$;
$A[0:m]$, $B[1:n]$ — arrays of coefficients of g_Δ;
$t[1:n]$ — array of nodes of g_Δ.

Result:

f — approximate value of $f(s)$ of equation (1) if $F \equiv$ **true** or of (2) if $F \equiv$ **false**.

2. Method used. The method from [2] has been used. Additionally, the numerically stable method from [1] is proposed to the determination of A_i and B_j in (5).

3. Certification. The procedure *inteqAbel2* has been tested on the Odra 1204 computer. The obtained results of calculations have been given in [2].

References

[1] H. Malinowski and R. Smarzewski, *A numerical method for solving the Abel integral equation*, Zastosow. Matem. 16 (1978), p. 275-281.

[2] R. Smarzewski and H. Malinowski, *Numerical solutions of a class of Abel integral equations*, J. Inst. Maths. Applics. 22 (1978), p. 159-170.

DEPARTMENT OF NUMERICAL METHODS
M. CURIE-SKŁODOWSKA UNIVERSITY
20-031 LUBLIN

Received on 7. 7. 1978

ALGORYTM 67

H. MALINOWSKI i R. SMARZEWSKI (Lublin)

WYZNACZANIE ROZWIĄZAŃ RÓWNAŃ CAŁKOWYCH ABELA, II

STRESZCZENIE

Procedura *inteqAbel2* wyznacza aproksymację $f_\Delta(s)$ rozwiązania $f(s)$ równania całkowego (1) lub (2). Aproksymacja $f_\Delta(s)$ określona jest odpowiednio przez (3) lub (4), gdzie g_Δ jest funkcją sklejaną stopnia $m = 2k-1$ $(1 < k \leqslant n)$, interpolującą funkcję g w węzłach t_i, $0 = t_1 < t_2 < \ldots < t_n = R$, i daną wzorem (5).

Dane:

m — stopień funkcji sklejanej (5);
n — liczba węzłów funkcji sklejanej g_Δ;
s — dowolnie ustalony punkt z przedziału $(0, R]$;
F — zmienna boolowska; gdy $F \equiv$ **true**, wtedy rozwiązywane jest równanie (1), w przeciwnym razie — równanie (2);
R — jeśli $F \equiv$ **false**, to R jest górną granicą całki (2), w przeciwnym razie — jest nieistotne;
g — jeśli $F \equiv$ **true**, to g jest równe $g(0)$, w przeciwnym razie zaś $g(R)$;
$A[0:m]$, $B[1:n]$ — tablice współczynników g_Δ;
$t[1:n]$ — tablica węzłów g_Δ.

Wynik:

f — dokładna wartość $f_\Delta(s)$, aproksymująca rozwiązanie $f(s)$ równania (1), gdy $F \equiv$ **true**, lub równania (2), gdy $F \equiv$ **false**.

W procedurze *inteqAbel2* zastosowano metodę szczegółowo opisaną w [2]. Do wyznaczania współczynników A_i oraz B_j w (5) autorzy proponują numerycznie stabilną metodę z pracy [1].

Obliczenia, wykonane na maszynie Odra 1204, wykazały poprawność algorytmu. Wyniki obliczeń przedstawiono w pracy [2].
