Pavla Holasová

Algorithms. 27. AULEY. Finding zeros of a polynomial by the extended Mc Auley's method

Aplikace matematiky, Vol. 17 (1972), No. 5, 403–407

Persistent URL: `http://dml.cz/dmlcz/103431`

ALGORITMY

27. AULEY

FINDING ZEROS OF A POLYNOMIAL BY THE EXTENDED MC AULEY'S METHOD

PAVLA HOLASOVÁ, Matematicko-fysikální fakulta KU, Praha

Procedure *AULEY*$(a, n, eps1, eps2, mO, nO, \mathrm{L})$ zeros: (x, y);
comment a is a field of the real coefficients of a polynomial, n being its degree, $eps1$, $eps2$ the admissible inaccuracies, $m0$, $n0$ the initial approximations for the coefficients m, n of the quadratic factor $Q(x) = x^2 + mx + n$, x and y are respectively the fields of real, and imaginary parts of the zeros;
value n, $eps1$, $eps2$, $m0$, $n0$;
integer n;
real $eps1$, $eps2$, $m0$, $n0$;
array a, x, y; **label** L;
begin integer i, $niter$, $npol$, k;
  real mk, nk, $r1$, $r2$, $r3$, $r4$, $r5$, $r6$, $dr1m$, $dr2m$, $dr1n$, $dr2n$, dm, dn, $d2r1n$, $d2r2n$, $d21mn$, $d22mn$, $d2r1m$, $d2r2m$, s, $s1$, $s2$, $mk1$, $nk1$, tm, tn;
  array $ma\ [-2{:}30]$, mb, $mc\ [-2{:}30]$;
  if $abs\ (a\ [n])\ -1.0 \leqq eps1$ **then go to** AL;
  for $i := 0$ **step** 1 **until** $n-1$ **do**
  $a[i] := a[i]/a[n]$;
  $a[n] := 1.0$;
AL: $npol := 0$;
AF: $niter := 0$;
  $mk := m0$;
  $nk := n0$;
$AF1$: $ma[-2] := 0$; $ma[-1] := 0$;
  for $i = 0$ **step** 1 **until** $n-3$ **do**
  $ma[i] := (a[i] - mk \times ma[i-1] - ma[i-2])/nk$;
  $ma[n-2] := 1$;
  $r1 := a[n-1] - (mk + ma[n-3])$;
  $r2 := a[n-2) - (nk + mk \times ma[n-3] + ma[n-4])$;

```
      mb[−2] := 0;
      mb[−1] := 0;
      for i := 0 step 1 until n − 3 do
      mb[i] := (ma[i] − mk × mb[i − 1] − mb[i − 2)]/nk;
      dr1m := mb[n − 4] − 1.0;
      dr2m := mk × mb[n − 4] + mb[n − 5] − ma [n − 3];
      dr1n := mb[n − 3];
      dr2n := mk × mb[n − 3] + mb[n − 4] − 1.0;
      r3 := −dr1n; r4 := −dr2n;
      dn := (dr2m × r1 − dr1m × r2)/(dr2n × dr1m − dr1n × dr2m);
      dm := (dr2n × r1 − dr1n × r2)/(dr1n × dr2m − dr1m × dr2n);
      comment the end of the first step of the method, the second step follows which
      makes dm and dn more exact, this step is repeated twice;
      mc[−2] := 0;
      mc[−1] := 0;
      for i := 0 step 1 until n − 3 do
      mc[i] := (2 × mb[i] − mk × mc[i − 1] − mc[i − 2])/nk;
      r5 := −mc[n − 3]; d2r1n := r5;
      r6 := −(mc[n − 4] + mk × mc[n − 3]);
      d2r2n := r6;
      d21mn := r6 − r5 × mk;
      d22mn := r3 − r5 × nk;
      d2r1m := −r5 × nk − r6 × mk + 2 × r3 + r5 × mk ↑ 2;
      d2r2m := nk × (−r6 + r5 × mk);
      k := 1;
AK:   s := r1 + 0.5 × d2r1m × dm ↑ 2 + d21mn × dm × dn + 0.5 × d2r1n
        × dn ↑ 2;
      s1 := r2 + 0.5 × d2r2m × dm ↑ 2 + d22mn × dm × dn + 0.5 × d2r2n
        × dn ↑ 2;
      dn := (dr2m × s − dr1m × s1)/(dr2n × dr1m − dr1n × dr2m);
      dm := (dr2n × s − dr1n × s1)/(dr1n × dr2m − dr1m × dr2n);
      comment the end of the second step;
      k := k + 1;
      if k = 2 then go to AK;
      mk1 := mk + dm;
      nk1 := nk + dn;
      tm := abs((mk1 − mk)/mk1);
      tn := abs((nk1 − nk)/nk1);
      if tm ≧ eps1 then go to JE;
      if tn ≧ eps1 then go to JE;
RYC:  niter := niter + 1;
      comment computation of the polynomial zeros follows;
```

```
BF:   s2 := −mk1/2;
      s := s2 ↑ 2 − nk1;
      if s ≧ 0 then go to REA;
      s1 := sqrt(−s);
      x[2 × npol + 1] := s2;
      y[2 × npol + 1] := s1;
      x[2 × npol + 2] := s2;
      y[2 × npol + 2] := −s1;
      go to DA;
REA:  s1 := sqrt(s);
      s := s2 + s1;
      x[2 × npol + 1] := s;
      y[2 × npol + 1] := 0.0;
      s := s2 − s1;
      x[2 × npol + 2] := s;
      y[2 × npol + 2] := 0.0;
DA:   npol := npol + 1;
      n := n − 2;
      if n ≦ 0 then go to KO;
      if n < 3 then go to NM2;
      comment division of the polynomial by the quadratic factor follows;
      for i := 0 step 1 until n do
      a[i] := ma[i];
      comment the installation of the computed values mk, nk for the initial values of
      the coefficients m, n of the next quadratic factor follows;
      m0 := mk;
      n0 := nk;
      go to AF;
NM2:  if n = 2 then
      begin niter := 0;
            mk1 := ma[1];
            nk1 := ma[0];
            go to BF
      end;
      s := −ma[0];
      x[2 × npol + 1] := s;
      y[2 × npol + 1] := 0.0;
      go to KO;
JE:   if abs(r1) > eps2 then go to PRY;
      if abs(r2) < eps2 then go to RYC;
PRY:  niter := niter + 1;
      mk := mk1;
```

```
    nk := nk1;
    if niter < 500 then go to AF1;
    comment if the number of the iterations is 500, the computation will continue
    at the label L;
    go to L;
KO: end;
```

The above procedure isolates successively the quadratic factors

$$Q(x) = x^2 + mx + n$$

of the polynomial

$$f(x) = a_0 x^N + a_1 x^{N-1} + \ldots + a_N \quad (a_i \ldots \text{real numbers})$$

by means of the extended Mc Auley's Method described in [1]. The convergency criteria used:

$$\left|\frac{m_{k+1} - m_k}{m_k}\right| \leqq eps1 \quad \text{and} \quad \left|\frac{n_{k+1} - n_k}{n_k}\right| \leqq eps1$$

or

$$|r_1| \leqq eps2 \quad \text{and} \quad |r_2| \leqq eps2 \,,$$

where

$$\lim_{k \to \infty} m_k = m \,, \quad \lim_{k \to \infty} n_k = n$$

and r_1, r_2 are the coefficients of the remainder,

$$r(x) = r_1 x + r_2$$

and

$$\frac{f(x)}{x^2 + m_k x + n_k} = r_1 x + r_2 \quad \text{holds.}$$

The values $eps1$, $eps2$ depend on the accuracy of the computer. On the computer MINSK 22, it was taken

$$eps1 = 10^{-6}$$

$$eps2 = 10^{-6} \,.$$

For $m0$, $n0$ it usually suffices to choose

$$m0 = 1$$

$$n0 = 1 \,.$$

If the method with the above mentioned initial values does not converge, the choice

$$n0 = \max_{i=1\ldots N} |a_i|$$
$$m0 = n0/2$$

may help in some cases.

The program has been made up according to [1] for the computer MINSK 22 in Fel Algol. It has been verified in fifteen examples (for polynomials of degrees 6−9). The results show that this method is a rapid and exact one, and at the same time its algorithm appears to be relatively short. Solving a non-trivial polynomial of degree 9 takes 2.2 minutes, with the accuracy of five decade figures. The program occupies 1127 memory cells. One can see the advantages of this method also from the tables in the paper [1]. However, it is necessary to remark that the results of these tables were obtained on a highly accurate computer by means of double precision (48 bit numbers in single-precision, 96 bit number in double-precision).

Example

$$f(x) = x^6 - 1\,.$$

$$\begin{array}{lll} \text{Zeros:} & -0{\cdot}5 \pm 0{\cdot}866025 & (1) \\ & +0{\cdot}5 \pm 0{\cdot}866025 & (8) \\ & +1 & (0) \\ & -1 & (0) \end{array}$$

(The number of iterations is stated in brackets).

Literature

[1] *John L. Smallwood:* A comparison of five numerical methods for solving polynomial equations with real coefficients, TNN-88, Jan. 1969.