ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XII, 2 (1971)

ALGORITHM 12

Z. CYLKOWSKI (Wrocław)

# MINIMUM OF A FUNCTION OF ONE VARIABLE

## 1. Procedure declaration.

```
real procedure minif(f, eps, t);
  value eps;
  real f, eps, t;
  comment The function minif gives the minimum value of
    the function f(t) of one variable. It is assumed that
    1° function f(t) has a minimum near the zero-point,
    2° the interpolation polynomial of degree at most two which agrees
    with the function at certain nodes t−1, t, t+1 does not differ much
    from the function.
    Data:
      f — arithmetic expression, dependent upon the parameter t and
          having the value f(t),
    eps — positive number, approximately the absolute error of the
          point in which f(t) reaches its minimum.
    Additional result:
    t — point in which f(t) reaches its minimum;
  begin
    real p, q, sg, t1, t2, t3, t4, y1, y2, y3, y4;
    t: = t1: = 0;
    y1: = f;
    t: = t2: = t3: = sg: = 1;
    y2: = f;
    if y1 < y2
      then begin
             t1: = sg: = −1;
             t2: = 0;
             y3: = y1;
             y1: = y2;
             y2: = y3
```

```
            end y1 < y2
        else t3: = 2;
    t: = t3 × sg;
    y3: = f;
aa: if y1 = y2 ∧ y2 = y3 ∨ y1 < y2 ∧ y1 < y3
        then begin
                t: = t1 × sg;
                minif: = y1;
                go to dd
            end y1 = y2 = y3 ∨ y1 = min(y1, y2, y3);
    q: = t3 − t1;
    p: = q + q + t3;
    y4: = (y2 − y1) × q;
    t4: = (y3 − y1) × (t2 − t1);
    q: = t4 − y4;
    if q ⩽ 0
        then t4: = p
        else begin
                t4: = ((t2 + t1) × t4 − (t3 + t1) × y4)/(q + q);
                if t4 > p
                    then t4: = p
            end q > 0;
    if abs(t4 − t2) < eps ∨ t4 < t1 + eps
        then begin
                t: = t2 × sg;
                minif: = y2;
                go to dd
            end abs(t4 − t2) < eps ∨ t4 < t1 + eps;
    if abs(t4 − t3) < eps
        then begin
                t: = t3 × sg;
                minif: = y3;
                go to dd
            end abs(t4 − t3) < eps;
    t: = t4 × sg;
    y4: = f;
    if t3 ⩽ t4
        then go to bb
        else begin
                p: = t4;
                t4: = t3;
```

```
            t3: = p;
            p: = y4;
            y4: = y3;
            y3: = p;
            if t3 < t2
               then begin
                       p: = t3;
                       t3: = t2;
                       t2: = p;
                       p: = y3;
                       y3: = y2;
                       y2: = p
                     end t3 < t2
          end t3 > t4;
   if y2 ⩽ (if y3 ⩽ y4
              then y3
              else y4)
     then begin
            if 3 × (t4 − t2) < t2 − t1
              then begin
                       sg: = −sg;
                       t1: = −t4;
                       t4: = −t2;
                       t2: = −t3;
                       y1: =  y4;
                       y4: =  y2;
                       go to cc
                     end 3 × (t4 − t2) < t2 − t1
          end y2 = min(y2, y3, y4)
     else if y4 ⩽ y3 ∨ t4 − t3 ⩽ 3 × (t3 − t1)
            then begin
                 bb: t1: = t2;
                     y1: = y2;
                     t2: = t3;
                 cc: y2: = y3;
                     t3: = t4;
                     y3: = y4
                  end y4 = min(y2, y3, y4) ∨
                       y3 = min(y2, y3, y4) ∧ t4 − t3 ⩽ 3 × (t3 − t1);
     go to aa;
dd: end minif
```

**2. Method used.** The function *minif* uses the modified method of Powell [1] of calculating the minimum of a function of one variable. The algorithm calculates $m = \min_t f(st)$, where $s = \pm 1$.

The number $s$ and the three interpolation nodes $t_1, t_2, t_3$ are defined at the beginning as follows:

$$(s, t_1, t_2, t_3) = \begin{cases} (1, 0, 1, 2), & \text{if } f(0) \geqslant f(1), \\ (-1, -1, 0, 1), & \text{if } f(0) < f(1). \end{cases}$$

The notation $f_k = f(st_k)$ $(k = 1, 2, 3, 4)$ is used in the algorithm which is composed as follows:

1. If $f_1 = f_2 = f_3$ or $f_1 < f_2$ and $f_1 < f_3$ then the calculations are finished and $m = f_1$, $t = st_1$.

2. Let $w(t) = at^2+bt+c$ be a polynomial satisfying $w(t_1) = 0$, $w(t_2) = f_2-f_1$, $w(t_3) = f_3-f_1$. Evaluated is

$$t_4 = \begin{cases} -\dfrac{b}{2a} & \text{for } a > 0 \text{ and } -\dfrac{b}{2a} \leqslant t_3+2(t_3-t_1), \\ t_3+2(t_3-t_1) & \text{otherwise}. \end{cases}$$

3. If $|t_4-t_2| < eps$ or $t_4 < t_1+eps$ then the calculations are finished and $m = f_2$, $t = st_2$.

4. If $|t_4-t_3| < eps$ then the calculations are finished and $m = f_3$, $t = st_3$.

5. The numbers $t_1, t_2, t_3, t_4$ are ordered as follows $t_1 < t_2 < t_3 < t_4$.

6. Go to step 1 if

$$f_2 = \min\{f_2, f_3, f_4\} \quad \text{and} \quad t_2-t_1 \leqslant 3(t_4-t_2)$$

or

$$f_3 = \min\{f_2, f_3, f_4\} \quad \text{and} \quad t_4-t_3 > 3(t_3-t_1)$$

else set

$$(s, t_1, t_2, t_3) = \begin{cases} (-s, -t_4, -t_3, -t_2), & \text{if } f_2 = \min\{f_2, f_3, f_4\}, \\ (s, t_2, t_3, t_4) & \text{otherwise} \end{cases}$$

and also go to step 1.

**3. Certification.** The function *minif* has been used to calculate the minima of the functions

$$f_1(t) = \frac{\sin 50t}{t}, \quad f_2(t) = te^{\frac{1}{80}t} \text{ and } \quad f_3(t) = [3|t-20|],$$

where [ ] denotes the integral part. Table 1 gives the calculation results obtained on the Odra 1204 computer.

TABLE 1

| Function | *eps* | *t* | *minif* | Number of function evaluations |
|---|---|---|---|---|
| $f_1(t)$ | $5_{10}-2$ | 1.1125218082 | −.8292540240 | 6 |
| | $5_{10}-5$ | 1.1099127737 | −.9008871327 | 11 |
| | $5_{10}-8$ | 1.1099634867 | −.9009157185 | 14 |
| $f_2(t)$ | $5_{10}-2$ | −79.9464305015 | −29.4303486917 | 12 |
| | $5_{10}-5$ | −79.9999034134 | −29.4303552929 | 15 |
| | $5_{10}-8$ | −79.9999676709 | −29.4303552929 | 16 |
| $f_3(t)$ | $5_{10}-2$ | 19.9807032754 | .0000000000 | 11 |
| | $5_{10}-5$ | 19.9807032754 | .0000000000 | 13 |
| | $5_{10}-8$ | 19.9807032754 | .0000000000 | 13 |

**Reference**

[1] M. J. D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Computer Journ. 7 (1964), p. 155-162.

DEPT. OF NUMERICAL METHODS
MATHEMATICAL INSTITUTE
UNIVERSITY OF WROCŁAW

*Received on 29. 7. 1970*

---

ALGORYTM 12

Z. CYLKOWSKI (Wrocław)

## WARTOŚĆ MINIMALNA FUNKCJI JEDNEJ ZMIENNEJ

STRESZCZENIE

Wartością funkcji *minif* jest wartość minimalna funkcji $f(t)$ jednej zmiennej. O funkcji $f(t)$ zakłada się, że

1° osiąga minimum w pobliżu zera,

2° wielomian interpolacyjny co najwyżej drugiego stopnia, identyczny z funkcją w węzłach $t-1$, $t$, $t+1$, nie odbiega zbytnio od niej.

Dane:

*f* — wyrażenie arytmetyczne o wartości *f*(*t*), zależne od parametru *t*,

*eps* — liczba dodatnia; w przybliżeniu błąd bezwzględny punktu, w którym funkcja *f*(*t*) osiąga minimum.

Wynik dodatkowy:

*t* — punkt, w którym funkcja *f*(*t*) osiąga minimum.

W funkcji *minif* użyto zmodyfikowanej metody Powella [1] obliczania minimum funkcji jednej zmiennej. Szczegóły zawiera § 2. Wyniki otrzymane na maszynie cyfrowej Odra 1204, zamieszczone w § 3, wykazały poprawność algorytmu.