Josef Čermák

Algoritmy. 24. FOURIER. Fourierova analýza

Aplikace matematiky, Vol. 16 (1971), No. 6, 461–465

Persistent URL: http://dml.cz/dmlcz/103383

ALGORITMY

24. FOURIER

FOURIEROVA ANALÝZA

JOSEF ČERMÁK, CSc., VŠChT Pardubice

procedure *FOURIER* (n, t, y, *period*, p, l, s, vl, vs) výsledek: (c, g);

value n, l, *period*, vl, vs;

integer n, l, s, p;

real *period*, vl, vs, g;

real array t, y, c;

begin comment Tato procedura hledá c_i v rozvoji

$$f(t) = c_1 + \sum_{i=1}^{p} (c_{2i} \cos i\omega t + c_{2i+1} \sin i\omega t) + \Theta(t)$$

kde $f(t)$ je dáno tabulkou t_j, $Y_{j,0}$ pro $j = 1, 2, \ldots, n(f(t_j) = Y_{j,0})$. Hodnoty t_j nemusí být ekvidistantní ani nemusí tvořit monotonní posloupnost. Musí však platit $n \geqq 2p + 1$. Identifikátor *period* odpovídá periodě funkce $f(t)$. Po provedení procedury je:

$$\begin{aligned}
Y_{j,1} &= f(t_j) - \Theta(t_j) \\
Y_{j,2} &= \Theta(t) \\
Y_{j,3} &= c_2 \cos \omega t_j + c_4 \sin \omega t_j \\
Y_{j,4} &= c_4 \cos 2\omega t_j + c_5 \sin 2\omega t_j \\
Y_{j,5} &= c_6 \cos 3\omega t_j + c_7 \sin 3\omega t_j \\
&\text{atd.}
\end{aligned}$$

a za g se dosadí součet $\sum_{i=1}^{n} [\Theta(t_i)]^2$.

Blok *GRAF* této procedury vynáší grafy těchto funkcí:

(1)	$f(t)$	jednotlivé body označuje písmenem	O
	$f(t) - \Theta(t)$	jednotlivé body označuje písmenem	X
	$\Theta(t)$	jednotlivé body označuje písmenem	Z
	$c_2 \cos 2\omega t + c_3 \sin \omega t$	jednotlivé body označuje písmenem	A
	$c_4 \cos 2\omega t + c_5 \sin 2\omega t$	jednotlivé body označuje písmenem	B
	$c_6 \cos 3\omega t + c_7 \sin 3\omega t$	jednotlivé body označuje písmenem	C

(kdyby se na témže místě grafu měla tisknout dvě nebo více písmen, pak se z nich tiskne to, jež je v (1) uvedeno nejníže.) V tomto grafu je osa *t* umístěna svisle. *l* udává počet řádků, *s* počet sloupců, v nichž bude graf vynesen. Podle užitého typu tiskárny je nutno udat vzdálenost řádků *vl* a středovou vzdálenost sousedních znaků *vs* (obojí např. v *mm*);

```
integer n1, i, j, k, m;
real s1, r, omega, tt, yy;
real array a[1 : 2 × p + 1, 1 : 2 × p + 1], b[1 : 2 × p + 1];
n1 := 2 × p + 1; omega := 6.28318531/period;
for i := 1 step 1 until n1 do
  begin b[i] := 0;
    for j := 1 step 1 until n1 do a[i, j] := 0
  end i;
a[1, 1] := n;
for k := 1 step 1 until n do
  begin s1 := 0; tt := t[k]; yy := y[k, 0]; b[1] := b[1] + yy;
    for j := 2 step 2 until n1 do
      begin s1 := s1 + omega; c[j] := cos (tt × s1);
        a[1, j] := a[1, j] + c[j]; c[j + 1] := sin (tt × s1);
        a[1, j + 1] := a[1, j + 1] + c[j + 1]
      end j;
    for j := 2 step 1 until n1 do
      begin s1 := c[j]; r := yy × s1; b[j] := b[j] + r;
        a[j, 1] := a[1, j];
        for i := 2 step 1 until n1 do a[j, i] := a[j, i] + c[i] × s1
      end j
  end k;
for k := 1 step 1 until n1 do
  begin r := 1/a[k, k];
    for j := k + 1 step 1 until n1 do a[k, j] := a[k, j] × r;
    b[k] := b[k] × r;
    for j := k + 1 step 1 until n1 do
      for i := k + 1 step 1 until n1 do a[i, j] := a[i, j] − a[i, k] × a[k, j];
      for i := k + 1 step 1 until n1 do b[i] := b[i] − a[i, k] × b[k]
  end k;
```

```
for i := n1 step -1 until 1 do
  begin s1 := 0;
    for j := i + 1 step 1 until n1 do s1 := s1 + a[i, j] × c[j];
    c[i] := b[i] - s1
  end zde je možno provést výstup koeficientů C_i pro i = 1, 2, ..., 2p;
g := 0;
for k := 1 step 1 until n do
  begin real cc, om; tt := t[k]; cc := c[1]; om := 0;
    for f := 1 step 1 until p do
      begin
        om := om + omega; r := c[2 × j] × cos (om × tt) + c[2 × j + 1] ×
        × sin (om × tt); cc := cc + r; y[k, j + 2] := r
      end j;
    y[k, 1] := cc; r := y[k, 0] - cc; y[k, 2] := r; g := g + r × r
  end k;
comment zde je možno provést výstup součtu g druhých mocnin odchylek aproximo-
vané funkční hodnoty od dané;
GRAF:
begin integer array inlin [1 : l], sp[1 : s];
  real xmin, xmax, ymin, ymax, xmer, ymer;
  comment dále následuje provedení grafu. Předpokládá se existence výstupní proce-
  dury line(k) pro k nových řádků a procedury text (A), (kde A je řetěz) tisknoucí
  výstupním prostředkem řetěz A;
m := p + 2;
for i := 1 step 1 until l do inlin [i] := 0;
xmin := xmax := t[1]; ymin := ymax := y[1, 1];
for i := 1 step 1 until n do
  begin xmer := t[i];
    if xmin > xmer then xmin := xmer;
      if xmax < xmer then xmax := xmer;
      for j := 0 step 1 until m do
      begin ymer := y[i, j];
        if ymin > ymer then ymin := ymer;
        if ymax < ymer then ymax := ymer;
      end j
  end i;
```

```
xmer := (l − 1)/(xmax − xmin); ymer := (s − 1)/(ymax − ymin);
for i := n step −1 until 1 do inlin[1 + (t[i] − xmin) × xmer] := i;
for i := 1 step 1 until l do
  begin
    switch sw := sw1, sw2, sw3, sw4, sw5, sw6, sw7;
    comment je-li p > 3, třeba připojit ještě dalších p − 3 cílových výrazů, jimiž
    jsou návěští sw8, sw9, ... za příkaz sw7 a za ním následující go to s1 připojit
    ještě:
    sw8: text ('e') go to s1
    sw9: text ('f') go to s1
    ...
    ...
    (jednotlivé instrukce ovšem nutno oddělit středníkem)
    Procedury text v těchto instrukcích obstarávají tisk bodů funkcí
    c8 cos 4ωt + c9 sin 4ωt
    c10 cos 5ωt + c11 sin 5ωt,
    ...
    jež budou v grafu zobrazeny znaky E, F, ...;
    if inlin [i] = 0 then go to newline;
    for j := 1 step 1 until s do sp[j] := 1;
    for j := 0 step 1 until m do
    sp[1 + (y[inlin[i], j] − ymin) × ymer] := 2 + j; j := 1;
    cykl : go to sw[sp[j]];
    sw1 : text ('␣'); go to s1;
    sw2 : text ('0');  go to s1;
    sw3 : text ('x');  go to s1;
    sw4 : text ('z');  go to s1;
    sw5 : text ('a');  go to s1;
    sw6 : text ('b');  go to s1;
    sw7 : text ('c');  go to s1;
    s1 : j := j + 1; if j ≦ s then go to cykl;
    newline: line (1)
  end
```

zde je možné vytisknout:

ymin, jež udává polohu osy *T* grafu vzhledem k levému okraji papíru

xmin, jež udává polohu osy *Y* vzhledem k hornímu okraji grafu

$xmer \times vl$ měřítko na ose T

$ymer \times vs$ měřítko na ose Y

(měřítka jsou v těchže délkových jednotkách, v nichž byly udány vs a vl)

end

end proc FOURIER

Nechť je dána funkce $y = f(t)$ jednotlivými body (v počtu n). Pak hodnoty t uložíme do vektoru $t[1 : n]$, hodnoty y do nultého sloupce matice $y[1 : n, 0 : p + 2]$. Hodnota p udává maximální řád harmonické složky, který od programu žádáme. Uživatel procedury musí též udat velikost základní periody *period* signálu. Program určí koeficienty c_i harmonické řady

$$c[1] + \sum_{j=1}^{p} (c[2 \times i] \times \cos i\omega t + c[2 \times i + 1] \times \sin i\omega t) .$$

Proto je třeba v hlavním programu deklarovat pro c pole $[1 : 2 \times p + 1]$.

Hodnota g po projití procedury obsahuje součet druhých mocnin zbytku.

Programu lze použít též k aproximaci periodické funkce $y = f(t)$ trigonometrickými polynomy řádu p. Určíme n hodnot y_i příslušných zvoleným hodnotám t_i a programem pak určíme $2p + 1$ koeficientů trigonometrického polynomu.

Program provádí dálnopisem, případně tiskárnou i grafické znázornění výsledků. Hodnotám t_i je úměrný počet řádků, souřadnicím y pak počet mezer (případně jiných symbolů).

Kontrolní příklad:

Tabelováním funkce

$$y = \sin \frac{100\pi}{18}(t - 0{,}03) + \sin \frac{100\pi}{18}(2t - 0{,}06) + \sin \frac{100\pi}{18}(3t - 0{,}09)$$

pro hodnoty $t = 0, 0{,}03, 0{,}06 \ldots 0{,}36$ dostáváme následující tabulku:

$t_1 = 0{,}00$	$y_{1,0} = -2{,}36603$	$t_7 = 0{,}18$	$y_{7,0} = 0{,}63397$
$t_2 = 0{,}03$	$y_{2,0} = 0$	$t_8 = 0{,}21$	$y_{8,0} = 0$
$t_3 = 0{,}06$	$y_{3,0} = 2{,}36603$	$t_9 = 0{,}24$	$y_{9,0} = -0{,}63397$
$t_4 = 0{,}09$	$y_{4,0} = 1{,}73206$	$t_{10} = 0{,}27$	$y_{10,0} = 0$
$t_5 = 0{,}12$	$y_{5,0} = 0$	$t_{11} = 0{,}30$	$y_{11,0} = 0$
$t_6 = 0{,}15$	$y_{6,0} = 0$	$t_{12} = 0{,}33$	$y_{12,0} = -1{,}73206$
		$t_{13} = 0{,}36$	$y_{13,0} = -2{,}36603$

Volíme-li tyto hodnoty jako vstupní a položíme-li hodnotu *period* $= 0{,}36$ pak aplikací algoritmu dostáváme následující hodnoty

$$c_1 = 0\,, \quad c_2 = -0{,}500\,, \quad c_3 = 0{,}866\,, \quad c_4 = -0{,}866\,,$$
$$c_5 = 0{,}500\,, \quad c_6 = -1\,, \quad c_7 = 0\,.$$