Liet. matem. rink., T. 43, spec. nr., 592–596

Splaininių kreivių ir paviršių konstravimas MATLAB aplinkoje

Danutė PLUKIENĖ, Kostas PLUKAS (KTU)
el. paštas: kostas.plukas@ktu.lt

1. Įvadas

Normalizuotieji B splainai (toliau B splainai) yra patogi splainų erdvės bazė [1]. Bet kokia splaininė kreivė arba paviršius yra tiesinis B splainų darinys [2, 3]. Konstruojant splainines kreives ir paviršius, jie užrašomi parametrine forma, sutapatinant tiesinio darinio koeficientus su kontrolinių (de Būro) taškų koordinatėmis. Todėl, keičiant kurio nors kontrolinio taško padėtį, lokaliai keičiasi splaininės kreivės (paviršiaus) forma. Dėl šios savybės, o taip pat dėl kreivės (paviršiaus) glodumo splaininės kreivės ir paviršiai plačiai naudojami geometriniame dizaine, pvz., projektuojant automobilių kėbulus, lėktuvų fuzeliažus ir kt.

n-osios eilės splaininių kreivių ir paviršių konstravimas, naudojant paskalines procedūras, pateiktas [3] literatūroje. Šiame darbe nagrinėjami teoriniai ir praktiniai kreivių ir paviršių konstravimo aspektai naudojant MATLAB aplinką. Ši aplinka įgalina dviem eilėm sutrumpinti splaininių kreivių ir paviršių apskaičiavimo laiką ir lengvai pavaizduoti juos grafiškai.

2. B splainų parametrinė forma

Splaininėms kreivėms ir paviršiams konstruoti naudojami B splainai, kuriuos apibrėžia vienetinio žingsnio sveikaskaitinis tinklelis. Tokių B splainų parametrinė forma yra invariantiška tinklelio mazgo atžvilgiu. Todėl nagrinėsime n-osios eilės B splaino, apibrėžto mazgo $x = 0$ atžvilgiu, parametrinę formą $B_n^0(u)$, $0 \leqslant u \leqslant 1$. Splaino $B_n^0(u)$ analitinę išraišką intervale $[k, k+1]$, $k = \overline{0, n}$, pažymėkime $B_n^{0[k,k+1]}(u)$ ir aptarkime, kaip ją apskaičiuoti.

B splainai tenkina rekurenčiąją formulę [1]:

$$\begin{aligned} B_n^i(x) &= \frac{x - x_i}{x_{i+n} - x_i} \cdot B_{n-1}^i(x) + \frac{x_{i+n+1} - x}{x_{i+n+1} - x_{i+1}} \cdot B_{n-1}^{i+1}(x), \quad n \geqslant 1, \\ B_0^i(x) &= \begin{cases} 1, & x \in [x_i, x_{i+1}), \\ 0, & x \notin [x_i, x_{i+1}). \end{cases} \end{aligned} \tag{1}$$

Remdamiesi (1) išraiška, išvesime rekurenčiąją formulę apskaičiuoti $B_n^{0[k,k+1]}(u)$.

Kadangi tinklelis sveikaskaitinis ir jo žingsnis lygus 1, tai, įvedus pakeitimą, $u = x - k$ ir įvertinus, kad 1) $B_n^{1[k,k+1]}(u) = B_n^{0[k-1,k]}(u)$, 2) $B_0^{[0,1]}(u) = 1$, iš (1) formulės gausime

$$B_n^{[k,k+1]}(u) = \frac{1}{n}\left((u+k)B_{n-1}^{[k,k+1]}(u) + (n+1-k-u)B_{n-1}^{[k-1,k]}(u)\right), \quad k=\overline{0,n},$$
$$B_0(u) = \begin{cases} 1, & x \in [0,1], \\ 0, & x \notin [0,1], \end{cases} \tag{2}$$

čia viršutinis indeksas 0 praleistas.

(2) formulė sudaro n-osios eilės splaininių kreivių ir paviršių konstravimo pagrindą. Ji įgalina sudaryti universalią procedūrą, kuri leidžia duotoms n ir u reikšmėms apskaičiuoti $B_n^{[k,k+1]}(u)$, $k = \overline{0,n}$, reikšmes, iš kurių apskaičiuojami splaininių kreivių ir paviršių taškai.

3. Kreivių konstravimas

Tarkime, $p_i = (x_i, y_i), i = \overline{1,N}$, – kontrolinių taškų seka. Tada konstruojamos n-osios eilės splaininės kreivės dalies, kurią apibrėžia taškai $p_i, p_{i+1}, \ldots, p_{i+n}$, parametrinė forma [2, 3] yra:

$$p(u) = \sum_{l=i}^{n+i} p_l \cdot B_n^{[n+i-l,n+i+1-l]}(u), \quad u \in [0,1]. \tag{3}$$

Norint, kad kreivės pradžios taškas būtų p_1, o galo – p_N, kontrolinių taškų seką iš kairės papildykime $(n-1)$ taškais p_1, o iš dešinės – $(n-1)$ taškais p_N. Kadangi $\sum_{k=0}^{n} B_n^{[k,k+1]}(u) = 1$, tai nesunku įsitikinti, kad taip papildyta kontrolinių taškų seka apibrėš kreivę nuo p_1 iki p_N.

Greičiausiai kontrolinių taškų seką papildysime atlikę veiksmą:

$$p\_naujas = [repmat(p(1), 1, n-1)\ p\ repmat(p(N), 1, n-1)]\,.$$

Šioje formulėje MATLAB funkcija $repmat$ $(n-1)$ kartą pakartoja pirmojo parametro reikšmę. Todėl p_naujas apibrėžia anksčiau nurodytu būdu papildytą kontrolinių taškų seką. Tolesniame dėstyme šią seką žymėsime ankstesniu simboliu $p_i, i = \overline{1, N+2n-2}$.

Taigi, splaininės kreivės taškams apskaičiuoti reikia:

1) vieną kartą pagal (2) formulę apskaičiuoti $B_n^{[k,k+1]}(u)$, $k = \overline{0,n}$, visoms $u = 0 : h : 1$ reikšmėms; (ši formulė reiškia, kad u įgauna reikšmes nuo 0 iki 1, kurios nutolę viena nuo kitos žingsniu h; čia ir toliau naudosime MATLAB sintaksę).

2) šias reikšmes naudoti kiekvienos kreivės dalies taškams apskaičiuoti pagal (3) formulę.

MATLAB įgalina efektyviai apskaičiuoti B splainų reikšmes, kai $u = 0 : h : 1$.

***B* splaino reikšmių apskaičiavimas**

$h = 0.001;\ u = 0 : h : 1;\ m = numel(u);$
$t = u';\ b = zeros(1, n + 1);\ b(1) = 1; a = repmat(b, m, 1);$
$for\ l = 1 : n$
  $for\ k = l + 1 : -1 : 1$
    $if\ k \sim= 1$
      $a(:, k) = ((t + k - 1). * a(:, k) + (l + 2 - k - t). * a(:, k - 1))/l;$
    else
    $a(:, k) = (t + k - 1). * a(:, k)/l;$
    end
  end
end

Šiame algoritme standartinės MATLAB funkcijos apskaičiuoja:

- $numel(u)$ – masyvo u elementų skaičių,
- $zeros(1, n + 1)$ – nulinį vektorių, turintį $(n + 1)$ elementą,
- $repmat(b, m, 1)$ – $m \times (n + 1)$ formato matricą, kurios eilutės yra vektorius b.

Be to, simboliai $'$ ir $. *$ atitinkamai žymi transponavimo ir paelementę daugybos operacijas, o $a(:, k)$ žymi matricos a k-ąjį stulpelį.

Nesunku pastebėti, kad $a(i, k) = B_n^{[k,k+1]}(i \cdot h)$, $k = \overline{0, n}$, $i = \overline{0, (1/h)}$.

Dabar pagal (3) formulę brėšime splaininę kreivę.

***n*-osios eilės splaininės kreivės brėžimas**

$for\ i = 1 : N + n - 2$
  $c = x(i : i{+}n);\ d = c(end : -1 : 1);\ p = repmat(d, m, 1);\ xx = sum((a.{*}p)');$
  $c = y(i : i{+}n);\ d = c(end : -1 : 1);\ p = repmat(d, m, 1);\ yy = sum((a.{*}p)');$
  plot(xx,yy); hold on;
end

Čia simbolis $x(i : i{+}n)$ žymi masyvo x elementus, pradedant indeksu i ir baigiant $(i{+}n)$; simbolis $c(end : -1 : 1)$ žymi, kad masyvo c elementai surašomi atvirkščia tvarka, pradedant paskutiniuoju ir baigiant pirmuoju; funkcija $sum(g)$ apskaičiuoja matricos g stulpelių elementų sumas; funkcija $plot(xx, yy)$ brėžia funkcijos, nusakytos reikšmių lentele (xx, yy), grafiką, o funkcija $hold\ on$ reiškia, kad grafiko dalys bus brėžiamos tame pačiame lange.

Kaip matyti iš teksto, kreivė brėžiama dalimis, kurias nusako $(n + 1)$ gretimų kontrolinių taškų aibės: taškai $p_i, p_{i+1}, \ldots, p_{i+n}$ sudaro i-ąją, $i = \overline{1, N + n - 2}$, aibę. Be to, MATLAB funkcijos įgalina visus i-osios dalies taškus apskaičiuoti vienu metu. Ši galimybė dviem eilėm sutrumpina kreivės konstravimo laiką.

4. Paviršiaus konstravimas

Tarkime, kad n-osios eilės splaininio paviršiaus kontroliniai taškai nusakyti matrica $P = [p_{ij}]$, $i = \overline{1, M}$, $j = \overline{1, N}$, čia $p_{ij} = (x_{ij}, y_{ij}, z_{ij})$. Tada šio paviršiaus dalį, kurią nusako kontroliniai taškai, patenkantys į $(n+1)\times(n+1)$ formato submatricą $L=[p_{s,t}]$, $s=\overline{i,i+n}$, $t=\overline{j,j+n}$, sudarys taškai

$$t(u,v)=(x(u,v),y(u,v),z(u,v)), \quad 0\leqslant u\leqslant 1,\ 0\leqslant v\leqslant 1,$$

apskaičiuoti pagal formulę

$$t(u,v)=\sum_{k=1}^{n+1}\left(B_n^{[n+1-k,n+2-k]}(v)\cdot\left(\sum_{l=1}^{n+1}\left(p_{i+l,j+k}\cdot B_n^{[n+1-l,n+2-l]}(u)\right)\right)\right). \quad (4)$$

Aišku, kad konstruojant paviršių, submatrica L turi nuosekliai (po vieną stulpelį iš kairės į dešinę ir po vieną eilutę iš viršaus į apačią) slinkti per matricą P.

Norint, kad kontroliniai taškai $p_{11},p_{1N},p_{M1},p_{MN}$ priklausytų paviršiui, matricą P reikia papildyti, įvedant viršuje ir apačioje po $(n-1)$ eilutę, o kairėje ir dešinėje – po $(n-1)$ stulpelį. Papildomos eilutės viršuje pakartoja matricos P pirmąją, o apačioje – paskutiniąją eilutę. Papildomi stulpeliai kairėje pakartoja matricos P pirmąjį, o dešinėje – paskutinįjį stulpelį.

Matricą P, t.y., kontrolinių taškų *xp*, *yp* ir *zp* matricas, papildysime taip:

$$xt=[repmat(xp(1,:),n-1,1);\ xp;\ repmat(xp(M,:),n-1,1)];$$
$$xx=[repmat(xt(:,1),1,n-1)\ xt\ repmat(xt(:,N),1,n-1)];$$

Matricos xp ir zp išplečiamos analogiškai; jas atitinkamai žymėsime yy ir zz.

Tolesniame nagrinėjime laikysime, kad parametrų u ir v diskrečiųjų reikšmių skaičius lygus m.

Paviršiaus apskaičiavimas ir brėžimas

$for\ i=1:M+n-2$

$for\ j=1:N+n-2$

Submatricos L išskyrimas

$x=xx(i:i+n,j:j+n);\ y=yy(i:i+n,j:j+n);\ z=zz(i:i+n,j:j+n);$

Skaičiavimas pagal (4) *formulę*

$for\ k=1:n+1$

$c=x(:,k)';\ d=c(end:-1:1);\ p=repmat(d,m,1);\ xu(k,:)=sum((a.*p)');$

Analogiškai apskaičiuojamos matricos yu *ir* zu.

end

$for\ k=1:m$

$c=xu(:,k)';\ d=c(end:-1:1);\ p=repmat(d,m,1);\ xuv(k,:)=sum((a.*p)');$

Analogiškai apskaičiuojamos matricos yuv *ir* zuv.

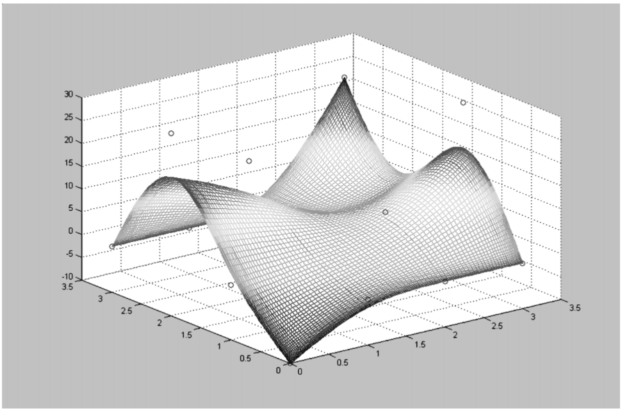
end

Apskaičiuotos paviršiaus dalies brėžimas

$mesh(xuv,yuv,zuv);$ *hold on*

end

end



1 pav. Kubinis splaininis paviršius.

Nesunku pastebėti, kad paviršiaus konstravimo algoritmas turi visus kreivės skaičiavime išvardintus privalumus.

1 pav. pavaizduotas pagal išnagrinėtą algorimą apskaičiuotas kubinis splaininis paviršius, kurį apibrėžia tokios kontrolinių taškų matricos:

$$xp = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{pmatrix}, \quad yp = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{pmatrix}, \quad zp = \begin{pmatrix} -10 & 0 & 0 & 0 \\ 2 & 3 & 10 & 30 \\ 30 & 20 & 0 & 0 \\ 0 & 0 & 0 & 25 \end{pmatrix},$$

o taip pat kontroliniai taškai.

Išvados

1. Išvesta rekurenčioji parametrinių B splainų apskaičiavimo formulė įgalina sudaryti universalias kreivių ir paviršių konstravimo procedūras.
2. MATLAB aplinka dėka lanksčių matricinių veiksmų įgalina sukurti efektyvius splaininių kreivių ir paviršių konstravimo algoritmus.

Literatūra

[1] J.S. Zavjalov, B.I. Kvasov, V.A. Miroshnichenko, *The Methods of Spline-Functions*, Nauka, Moscow (1980) (in Russian).
[2] K. Plukas, *Skaitiniai metodai ir algoritmai*, Naujasis Lankas, Kaunas (2001).
[3] K. Plukas, D. Plukienė, Splaininių kreivių ir paviršių konstravimas, in: *Informacinės technologijos*, Technologija, Kaunas (2001), pp. 132–136.

Construction of splines curves and surfaces using MATLAB

D. Plukienė, K. Plukas

Theoretical and practical aspects of construction splines curves and surfaces with MATLAB are discussed in this paper.