Fakulteta za matematiko in fiziko

Računsko podprto geometrijsko oblikovanje

VS postopek za izračun vrednosti polinomov več spremenljivk

Janez Radeščček, Miha Avsec

3. januar 2019

1 Decasteljou

Naj bo T trikotnik v ravnini, ter naj bodo (r,s,t) baricentrične koordinate točke P glede na trikotnik T. V tem primeru lahko vsak polinom stopnje d definiran nad trikotnikom T zapišemo v Bernsteinovi bazi kot

$$p(r,s,t)=\sum_{i=0}^{d}\sum_{j=0}^{i} b_{d-i,i-j,j}B^d_{d-i,i-j,j},$$

kjer velja

$$B^d_{i,j,k}(r,s,t)=\frac{d!}{i!j!k!}rs^jt^k.$$

Tako podan polinom lahko evaluiramo s pomočjo deCasteljouvovega algoritma

Algoritem 1.1. *Naj bo p polinom stopnje d podan v Bezierjevi obliki, ter naj bodo r,s,t baricentirčne koordinate točke za katere velja $r \geq s, r \geq t$, tedaj lahko s sledečim algoritmom izračunamo vrenost polinoma p v točki (r,s,t)*

```
for k=1:d
  for i=0:d-k
      for j=0:i
```
$$b^k_{d-i-k,i-j,j}=r*b^{k-1}_{d-i-k+1,i-j,j}+s*b^{k-1}_{d-i-k,i-j+1,j}+r*b^{k-1}_{d-i-k,i-j,j+1}$$
$$p(r,s,t)=b^d_{0,0,0}$$

Ta algoritem potrebuje $d(d+1)(d+2)/2$ operacij.

2 Modificirana Bernstein-Bezierjeva reprezentacija

Če imamo podan polinom v Bernsteinovi obliki

$$p(r,s,t)=\sum_{i=0}^{d}\sum_{j=0}^{i} b_{d-i,i-j,j}B^d_{d-i,i-j,j}(r,s,t),$$

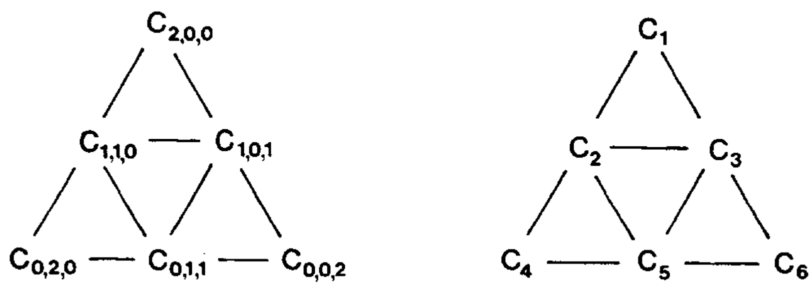
potem lahko tak polinom enostavno prepišemo v obliko

$$p(r,s,t)=\sum_{i=0}^{d}\sum_{j=0}^{i} c_{d-i,i-j,j}r^{d-i}s^{i-j}t^j,$$

tako da za koeficiente $c_{d-i,i-j,j}$ vzamemo

$$c_{d-i,i-j,j}=\frac{d!}{(d-i)!(i-j)!j!}b_{d-i,i-j,j}, \quad j=0,\ldots,i; i=0,\ldots,d.$$

Tej obliki polinoma rečemo modificirana Bernstein-Bezierjeva oblika ali krajše MBB. Pokazati želimo, da se polinom v MBB obliki, da evaluirati hitreje kakor v klasični Bezierjevi obliki. Ideja, ki se skriva v ozadju je ta, da lahko p zapišemo v gnezdeni obliki. Poglejmo si na primeru polinomov stopnje 2. Označimo kontrolne točke na sledeči način:



Če razpišemo sedaj polinom glede na spremenljivko r dobimo:

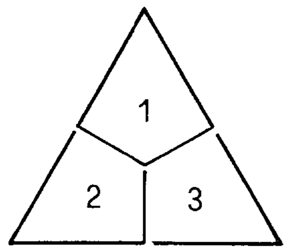
$$
\begin{aligned}
p(r,s,t) &= \sum_{i=0}^{2}\sum_{j=0}^{i} c_{2-i,i-j,j} r^{2-i} s^{i-j} t^{j} \\
&= r^2 \sum_{j=0}^{0} c_{2,-j,j} s^{-j} t^{j} + r \sum_{j=0}^{1} c_{1,1-j,j} s^{1-j} t^{j} + \sum_{j=0}^{2} c_{0,2-j,j} s^{2-j} t^{j} \\
&= r^2(c_1) + r(c_2 s + c_3 t) + (c_4 s^2 + c_5 st + c_6 t^2) \\
&= r^2(c_1 + \frac{s}{r}c_2 + \frac{t}{r}c_3 + \frac{s^2}{r^2}c_4 + \frac{st}{r^2}c_5 + \frac{t^2}{r^2}c_6) \\
&= r^2(\frac{s}{r}(c_2 + \frac{s}{r}c_4 + \frac{t}{r}c_5) + \frac{t}{r}(\frac{t}{r}c_6 + c_3) + c_1)
\end{aligned}
$$

Tu je potrebno biti pozoren na to, da ne delimo z 0. Torej je treba ločiti primere glede na to v katerem odseku trikotnika se naše baricentrične koordinate nahajajo in podobno razvijemo po spremenljivki s ali t. V Primeru ko razvijamo po r dobimo algoritem

Algoritem 2.1. *Naj bo p polinom stopnje n podan v MBB obliki, ter naj bodo r, s, t baricentirčne koordinate točke za katere velja $r \geq s, r \geq t$, tedaj lahko s ledečim algoritmom izračunamo vrenost polinoma p v točki (r, s, t)*

```
sr = s/r,          tr= s/r
A = c_{0,n,0};
for i = 1:n
    B = c_{0,n-i,i}
    for j = i:-1:1
        B = B*tr + c_{i-j+1,n-i,j-1};
    end
    A = A *sr +B;
end
p(r,s,t) = Ar^n
```

Zgornji algoritem potrebuje $(n^2 + 5n)/2$ množenj, 2 deljenj, ter $n - 1$ množenj za izračun r^n, kar lahko v splošnem naredimo tudi z manj operacijami (npr. za algoritmom kvadriraj in zmnoži). Potrebno je povedati še, kako se odločimo katero verzijo algoritma bomo vzeli.



Zgornji algoritem deluje za regijo ena, v kateri se nahajamo če $r \geq s, r \geq t$. S spremembami po kateri spremenljivki razvijamo pa pokrijemo tudi regiji 2 in 3. Pri čemer je regija 2 določena z $s > r, s \geq t$ in razvijamo po s, regija 3 pa z $t > r, t > s$ pri čemer razvijamo po t.

3 Polinom v Taylorjevi vrsti

Naj bo T trikotnik. Naj bo p polinom stopnje n definiran na T. Polinom p lahko zapišemo v Taylorjevi obliki kot

$$p(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{n-i} a_{i,j} u^i v^j.$$

To obliko bomo krajše klicali TAY. Vrednost polinoma p v točki (u, v) lahko izračunamo s sledečim algoritmom.

Algoritem 3.1.

```
p = a_{0,n}
for i = 1:d
    A = a_{i,n-i}
    for j = 1:i
        A = A * u + a_{i-j,n-i}
    end
    p = p * v + A
end
```

Ta postopek potrebuje $(n^2 + 3n)/2$ množenj.

4 Polinomi na tetraedrih

Naj bo T tetraeder. Za vsako točko U iz T naj bodo (r, s, t, u) pripadajoče Baricentrične koordinate glede na T. Naj bo p polinom stopnje n definiran na T. Polinom p lahko zapišemo v modificirani Bernstein-Bezierjevi obliki kot

$$p(r, s, t, u) = \sum_{i=0}^{n} \sum_{j=0}^{i} \sum_{k=0}^{j} c_{d-i,i-j,j-k,k} r^{d-i} s^{i-j} t^{j-k} u^{k}.$$

Vrednost polinoma p v točki (r, s, t, u) lahko izračunamo s sledečim algoritmom.

Algoritem 4.1.
TODO

Ta postopek potrebuje $(n^3 + 6d^2 + 17d)/6$ množenj in 3 deljenja.