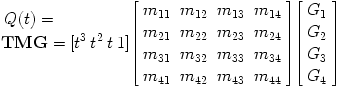
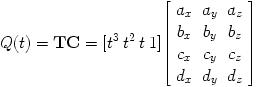
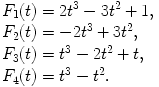
**Param. křivka**

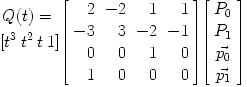
Q(t) = [x(t),y(t),z(t)] ~ **q**(t) = (...); teč. vekt. **q**‘(t) = dq(t)/dt; tečna P(m) = Q(t0) + m**q**‘(t0)

param. spoj. Cn 0 – dráha, 1 – směr a rychlost, 2 – zrychlení; geom. Gn podmínka - **q1**‘(1) = k \* **q2**‘(0), 1 – jen směr

**kubika.pngParam. kubika**

M – matice konstant, bázová m.; G – vektor geom. podmínek

sledujeme - invariance k lineárním transformacím (nez. na pořadí) konvexnost obálky, lokalita změn, křivka (ne)prochází počátečním a koncovým bodem řídícího polygonu.

**ferg2.pngFergusonova (Hermitova) kubika**

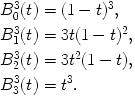
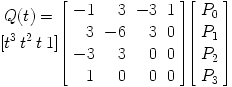
bez1.png

bez3.png

**bez2.pngBezierovy křivky**

bk n-stupně urč. n+1 body tvoř. řídící polygon

Bi - Bernsteinovy polynomy

**Beziérova kubika**

**bkub4.pngbkub1.pngCasteljau**

Beziér – půlení stran

**Coonsova kubika**

TODO

**Coonsův kubický B–spline**

TODO

**Cohen-Sutherland**

přímky vůči obdélníku, lze použít pro 3D. 4 bity UBRL, případy, koncové body: 0 – uvnitř, stejný jeden bit – mimo, jinak prochází hranou.

Nutnost float, mult, div.

Xmin, Xmax, Ymin, Ymax – hranice oblasti; zjištění kódu oblasti koncových bodů, jsou-li 0 oba, přeskočit ořezávání

cyklus; jestliže (k1 & k2) != 0, pak úsečka je mimo

výběr bodu, který není uvnitř, X, Y, kód bodu; test na jedničky v kódu:

(k & 1) != 0 -> y = ((y2–y1) / (x2–x1)) \* (Xmin–x) + y; x = Xmin; (k & 2) != 0 -> y = ((y2–y1) / (x2–x1)) \* (Xmax–x) + y; x = Xmax;

(k & 4) != 0 -> x = ((x2–x1) / (y2–y1)) \* (Ymin–y) + x; y = Ymin; (k & 8) != 0 -> x = ((x2–x1) / (y2–y1)) \* (Ymax–y) + x; y = Ymax;

opravení kódu pro nový bod; nerovnají-li se oba kódy 0, na začátek cyklu; vykreslit úsečku; další úsečka

**Cyrus-Beck**

Úsečky vůči konvex. n-úhelníku – nutno znát normály hran

A,B –koncové body úsečky; Ni – normála hrany; Yi – bod na hraně n-úhelníka; tmin = 0; tmax = 1

cyklus – projede všechny hrany

if (Ni \* (B-A)) = 0 -> hrana rovnoběžná s úsečkou

if (Ni \*(B-Pi)) = 0 -> úsečka na hraně, konec algoritmu, jinak pokračovat další hranou

if (Ni \* (B-A)) != 0 -> spočítat průsečík

t = - (Ni\*(A-Pi) / Ni \* (B-A)) -> výpočet průsečíku

if (Ni \*(B-A)) < 0 -> úsečka jde směrem do oblasti, tak tmin = max(t,tmin), jinak tmax = min(t,tmax)

dalsčí hrana

if (tmax >= tmin) -> vykreslit úsečku, jinak je úsečka mimo

**Weiler-Atherton**

Si – vrcholy n-úhelníku, Ci – vrcholy ořezávací oblasti, Ii – průsečíky; ve směru hod. 1. řádek vrcholy n-úh. a průsečíky, 2. řádek vrcholy oblasti + průsečíky; pokud má vzniknout víc n-úhelníků, tak ve třetí řádce \_vstupní\_ průsečíky. Vstup na průsečíku. Přechod do druhého řádku při průsečíku na stejný průsečík. Jsou li díry, pak proti směru hod.

Alg: Označit vrcholy oblasi po směru, vrcholy díry proti směru, vrcholy n-úh. po směru, díry n-úh. proti směru, označit všechny vzniklé průsečíky; Vypsat vrcholy n-ú. a průsečíky po směru (+1.), za to vrcholy díry a průsečíky proti směru; Vypsat vrcholy oblasti a průsečíky po směru (+1.), za to vrcholy díry oblasti a průsečíky proti směru. Seznam vstupních průsečíků; Začátek vstupním průsečíkem a odebrat ze seznamu, pak procházet seznamem vrcholů n-úhelníka. If (průsečík): if (!počáteční) ->pokračovat z 2. seznamu a odebrat ho ze seznamu. Vrchol jen přidat a pokračovat. Na konci pokračovat dál ze seznamu vst. vrcholů.

**Phongův osvětl. model**

N – normála, S – směr k bodovému zdroji, R – směr odraženého světla, V – směr ke kameře.

ambientní: Ia – intenzita světla, ka – koeficient amb. odrazu; difuzní a spekulární: kd – koef. dif. odrazu, ks – koef. spek. odr., IS – síla bodového světla, n – útlum; zrcadlový – index r, průhledný – index t

L = Ia ka + kd IS (N.S)/d2 + IS ks (**R.V**)n/d2 + kr Lr + kt Lt

**Phongův výpočet odrazu lesklého povrchu** = spekulární viz výše

**Konstantní stínování** – výpočet barvy plochy podle normály

**Gouraudovo stínování** – výpočet barvy vrcholů z jejich normál (ty se spočtou aritm. prům. ze všech normál ve vrcholu), hrany interpolací mezi vrcholy: IA = [I1.(YS-Y2)+I2(Y1-YS)]/(Y1-Y2); IB = [I1.(YS-Y3)+I3(Y1-YS)]/(Y1-Y3), pak podle jedné osy lin. interp. celý trojúh.:

IQ = [IA.(XB-XQ)+IB.(XQ-XA)]/(XB-XA)

**Phongovo stínování** – z normál ve vrcholech se interpolací vypočtou normály vnitřních bodů a z nich samotná barva bodu.

norm. vektory: nQ = nA+(nB-nA).t

**Optická densita** – log10( Imax / Imin)

**Půltónování – metoda vzorů** – jasný **Půlt. – M. konst. prahu** – jasný

**Půlt. – Floyd-Steinberg** – chyba distrib. doprava – dolů 3/8, ¼, 3/8

**Půlt. – dithering (rozmývání)** – vkládání chyb dle vzoru [0 2 | 3 1], větší vzory podle (2n)D = [4(n)U | 4(n)D + 2(n)U || 4(n)D + 3(n)U | 4(n)D + U(n)]

**Bressenham:**

line (x1, y1, x2, y2) {

x = x1; y = y1

dx = x2 – x1; dy = y2 – y1;

d = 2 \*dy – dx; a = 2 dy; b = 2 (dy – dx)

for (i = 0 to dx)

if (d <= 0) -> d = d + a; x = x + 1; step H (rovně)

else d = d + b; x = x + 1; y = y + 1; step D (šikmo)