Laboratorijske vaje pri predmetu Postopki konstruiranja

Leon Kos, Laboratorij za CAD - LECAD

Naloga

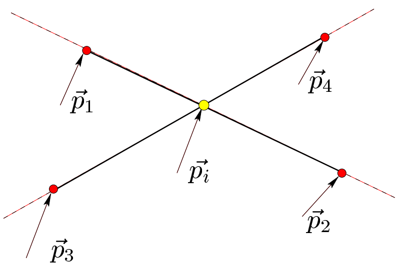
Izdelajte program za izračun poti orodja numerično krmiljenega stroja za poljubne zaprte dvodimenzionalne oblike sestavljene iz lokov in daljic. Generirana koda za numerično krmiljen stroj naj ustreza standardu DIN 66026/1/2 in ISO 5983/DIS (*G-code*). Program naj omogoča izračun odmika orodja za avtogeno rezanje, kot tudi frezanje v notranjosti likov. Velikost odmika orodja naj bo programsko nastavljiva. Potovanje orodja prikažite tudi gravfično, z uporabo jezika GKS in kompozicijo dvodimenzionalnih homogenih transformacij.

1 Dvodimenzionalna geometrija

Za avtogeno rezanje je potrebno narisane like občrtati z enakomerno oddaljenostjo odmika orodja w (*offset*). Pri frezanju notranjosti se krivulja potovanja orodja včrta in še serpentinasto zapolni.

1.1 Presek dveh daljic

Imejmo dve premici za kateri želimo izračunati preseka. Premica je določena, če poznamo dve točki na premici.



Slika 1: Presek dveh daljic

Če vsako točko predstavimo z vektorjem, npr. $\vec{p_1} = (x_1, y_1)$, lahko premico zapišemo s parametrično enačbo

$$\vec{p} = \vec{p_1} + \alpha(\vec{p_2} - \vec{p_1}) \tag{1}$$

kjer je $\vec{p}$ poljubna točka na premici določena s skalarjem α. Za daljico med točkama $\vec{p_1}$ in $\vec{p_2}$ je skalar α v mejah med nič in ena. Presek dveh premic $\vec{p_i}$, prikazanih sliki 1, zapišemo z enačbima

$$\vec{p_i} = \vec{p_1} + \alpha(\vec{p_2} - \vec{p_1}) \tag{2a}$$
$$\vec{p_i} = \vec{p_3} + \beta(\vec{p_4} - \vec{p_3}) \tag{2b}$$

Če enačbi (2) odštejemo dobimo enačbo

$$0 = (\vec{p_1} - \vec{p_3}) + \alpha(\vec{p_2} - \vec{p_1}) + \beta(\vec{p_4} - \vec{p_3}) \tag{3}$$

ki je v ekcplicitni obliki zapisana kot

$$\alpha(\vec{p_{2x}} - \vec{p_{1x}}) + \beta(\vec{p_{4x}} - \vec{p_{3x}}) = \vec{p_{3x}} - \vec{p_{1x}} \tag{4a}$$
$$\alpha(\vec{p_{2y}} - \vec{p_{1y}}) + \beta(\vec{p_{4y}} - \vec{p_{3y}}) = \vec{p_{3y}} - \vec{p_{1y}} \tag{4b}$$

Enačbi (4a) in (4b) tvorita sistem dveh linearnih enačb z dvema neznankama, ki ob rešitvi da:

$$\alpha = \frac{(\vec{p_{3y}} - \vec{p_{4y}})(\vec{p_{1x}} - \vec{p_{3x}}) - (\vec{p_{3x}} - \vec{p_{4x}})(\vec{p_{1y}} - \vec{p_{3y}})}{(\vec{p_{2y}} - \vec{p_{1y}})(\vec{p_{3x}} - \vec{p_{4x}}) - (\vec{p_{2x}} - \vec{p_{1x}})(\vec{p_{3y}} - \vec{p_{4y}})} \tag{5a}$$
$$\beta = \frac{(\vec{p_{2x}} - \vec{p_{1x}})(\vec{p_{1y}} - \vec{p_{3y}}) - (\vec{p_{2y}} - \vec{p_{1y}})(\vec{p_{1x}} - \vec{p_{3x}})}{(\vec{p_{2y}} - \vec{p_{1y}})(\vec{p_{3x}} - \vec{p_{4x}}) - (\vec{p_{2x}} - \vec{p_{1x}})(\vec{p_{3y}} - \vec{p_{4y}})} \tag{5b}$$

Presek premic obstaja, če je imenovalec enačb (5) različen od nič. Presek daljic d_{12} in d_{34} obstaja, če je α in β v mejah med nič in ena. Pogoji za presek dveh daljic so:

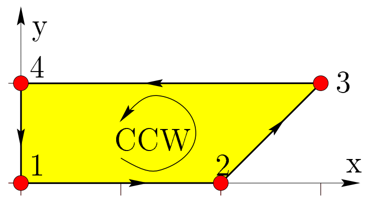
$$\begin{gathered}(\vec{p_{2y}} - \vec{p_{1y}})(\vec{p_{3x}} - \vec{p_{4x}}) - (\vec{p_{2x}} - \vec{p_{1x}})(\vec{p_{3y}} - \vec{p_{4y}}) \neq 0 \\ 0 \leq \alpha \leq 1 \\ 0 \leq \beta \leq 1\end{gathered} \tag{6}$$

1.2 Površina mnogokotnika

Površino mnogokotnika z n vozliščnimi točkami lahko izračunamo z enostavno formulo[4]

$$A = \frac{1}{2}\left|\begin{array}{ccccc} x_1 & x_2 & \dots & x_n & x_1 \\ y_1 & y_2 & \dots & y_n & y_1 \end{array}\right| \tag{7}$$

Točke mnogokotnika $P_i = (x_i, y_i)$ morajo biti v zaporedju nasprotnem urinem kazalcu (CCW). Shemo v enačbi (7) rešimo kot vsoto zaporednih poddeterminant. Primer izračuna površine za mnogokotnik na sliki 2.



Slika 2: Izračun površine mnogokotnika

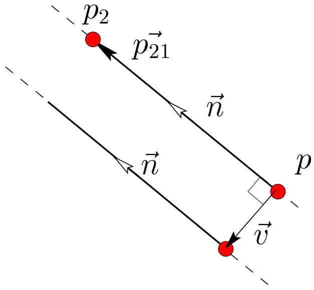
Mnogokotnik na sliki ima štiri točke za katere lahko zapišemo enačbo (7)

$$\begin{aligned} A &= \frac{1}{2}\left|\begin{array}{ccccc} 0 & 2 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array}\right| \\ &= \frac{1}{2}\left(\left|\begin{array}{cc} 0 & 2 \\ 0 & 0 \end{array}\right| + \left|\begin{array}{cc} 2 & 3 \\ 0 & 1 \end{array}\right| + \left|\begin{array}{cc} 3 & 0 \\ 1 & 1 \end{array}\right| + \left|\begin{array}{cc} 0 & 0 \\ 1 & 0 \end{array}\right|\right) \\ &= \frac{1}{2}(0 + 2 + 3 + 0) = \frac{5}{2} = 2.5 \end{aligned} \tag{8}$$

V primeru, da bi bilo zaporedje točk mnogokotnika v nasprotnem vrstnem redu, bi izračunali površino mnogokotnika z negativnim predznakom. Predznak pri izračunu površine se lahko koristno uporabi za določitev zaporedja točk pri določanju odmika.

1.3 Odmik daljice

Za izračun poti orodja je porebno poiskati enačbo premice, ki je vzporedna daljici podani s parametrično enačbo (1). Na sliki 3 je prikazan odmik premice od daljice p_{12} za razdaljo podano z vektorjem $\vec{v}$. Smerni vektor



Slika 3: Odmik premice od daljice

daljice podane s točkama $\vec{p_1} = (x_1, y_1)$ in $\vec{p_2} = (x_2, y_2)$ izračunamo z enačbo

$$\vec{n} = \frac{\vec{p_2} - \vec{p_1}}{|\vec{p_2} - \vec{p_1}|} \tag{9}$$

Če smerni vektor (9) zavrtimo za 90° in ga pomnožimo z željenim skalarjem za velikost odmika w dobimo vektor odmika $\vec{v}$, ki ga izračunamo kot

$$\begin{aligned} \vec{v} = wRot(90°, \vec{n}) &= w \begin{bmatrix} cos90° & -sin90° \\ sin90° & cos90° \end{bmatrix} \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} \\ v_x &= -wn_y \\ v_y &= w\, n_x \end{aligned} \tag{10}$$

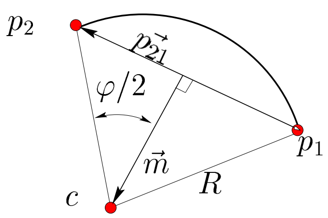
Nova parametična enačba premice, ki je za w odmaknjena od p_{12} je

$$\vec{p} = t\vec{n} + \vec{p_1} + \vec{v} \tag{11}$$

kjer je t poljubna konstanta, ki določa točko na premici, $\vec{p_1}$ katerakoli točka na osnovni premici in $\vec{v}$ vektor odmika (10).

1.4 Posplošene daljice

Daljice lahko razširimo v loke s tem, da jim dodamo lastnost ukrivljenosti $b = \tan(\varphi/4)$. S tem uvedemo enoten zapis za vse gradnike v bazi računalnika. Daljica ima ukrivljenost $b = 0$, loki pa ostale vrednosti.



Slika 4: Lok podan s točkama p_1, p_2 in ukrivljenostjo b

Podajanje četrtine celotnega loka je potrebno zaradi periodičnosti funkcije tangens, kot tudi zaradi enostavnega računanja središča loka brez uporabe kotnih funkcij.

Dolžino vektorja $\vec{m}$ izračunamo kot

$$l = |\vec{m}| = \frac{|\vec{p_2} - \vec{p_1}|}{2} \frac{1}{\tan(\varphi/2)} \tag{12}$$

Z uporabo funkcije dvojnih kotov za tangens

$$\tan 2\alpha = \frac{2\tan\alpha}{1 - \tan^2\alpha}$$

zapišemo enačbo (12) kot

$$l = \frac{|\vec{p_2} - \vec{p_1}|(1 - b^2)}{4b} \tag{13}$$

Dolžina l ima lahko tudi negativni predznak, kar pomeni, da je središče loka c na nasprotni strani. Koordinate središča loka izračunamo z enačbo

$$\vec{c} = \vec{p_1} + \frac{\vec{p_{21}}}{2} + Rot(90°, l\frac{\vec{p_{21}}}{|\vec{p_{21}}|}) \tag{14}$$

kjer je

$$\vec{p_{21}} = \vec{p_2} - \vec{p_1}$$

in

$$|\vec{p_{21}}| = \sqrt{(p_{2x} - p_{1x})^2 + (p_{2y} - p_{1y})^2}$$

Enačba (14) je problematična le, ko je ukrivljenost $b = 0$, kar je geometrijski pogoj, saj je takrat središče neskončno oddaljeno od daljice. Takrat lok degenerira v daljico.

1.4.1 Absolutni kot vektorja

Če želimo izračunati absolutni kot vektorja $\vec{p_{21}}$ uporabimo sledeči postopek:

1. Vektor normiramo $\vec{n} = \frac{\vec{p_2} - \vec{p_1}}{|\vec{p_2} - \vec{p_1}|}$
2. Koordinata n_y vektorja $\vec{n}$ je sinus kota α, n_x pa kosinus kota. Absolutni kot α izračunamo kot

$$\alpha = \begin{cases} \arccos n_x, & n_y \geq 0 \\ 2\pi - \arccos n_x, & n_y < 0 \end{cases} \tag{15}$$

pri računanju funkcije $\arccos n_x$, je potrebno kontrolirati območje krožne funkcije $n_x = [-1, 1]$, saj lahko n_x zaradi zaokrožitvenih napak zavzame vrednosti izven dovoljenih mej.

1.4.2 Kot med dvema vektorjema

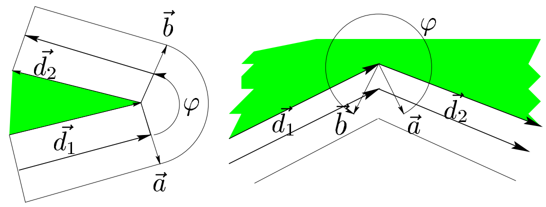
Imejmo dva *normirana* vektorja $\vec{a}$ in $\vec{b}$. Skalarni produkt dveh vektorjev je enak kosinusu kota med njima:

$$\cos\alpha = \vec{a} \cdot \vec{b} = a_x b_x + a_y b_y \tag{16}$$

Sinus kota med vektorjema je enak vektorskemu produktu dveh vektorjev

$$\sin\alpha = \vec{a} \times \vec{b} = a_x b_y + a_y b_x \tag{17}$$

S poznanima vrednostima $n_x = \cos\alpha$ in $n_y = \sin\alpha$ lahko z enačbo (15) izračunamo kot med vektorjema.



a) b)

Slika 5: Presek dveh odmaknjenih daljic. a) daljici je potrebno povezati z lokom b) daljici se skrajšata

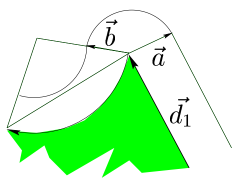
1.5 Presek dveh odmaknjenih daljic

Na sliki 5 sta prikazana dva možna primera presekov daljic. Če je kot med vektorjema $\vec{a}$ in $\vec{b}$ ali $\vec{d_1}$ in $\vec{d_2}$ med 0° in 180°, je potrebno odmaknjeni daljici povezati s krožnim lokom z ukrivljenostjo $b = \tan(\varphi/4)$. Za kot φ večje od 180° je potrebno izračunati presečišče z enačbo (3). Test vrivanja krožnega loka lahko zapišemo tudi takole:

$$w = d_{1x}d_{2y} - d_{1y}d_{2x} \ , \quad \begin{cases} w \leq 0 \ , & \text{izračun preseka} \\ w > 0 \ , & \text{zaokrožitev} \end{cases} \tag{18}$$

1.6 Presek loka in daljice

Tudi pri izračunu preseka odmaknjenega loka in daljice obstaja možnost zaokrožitve. Na sliki 6 je prikazan tak primer preseka.



Slika 6: Presek odmaknjenega loka in daljice

Pogoj vrivanja loka je podoben (18), le da moramo tu uporabiti pravokotna vektorja $\vec{a} = Rot(-90°, \vec{d_1})$ in $\vec{b}$, ki tvorita pogoj $w = a_x b_y - a_y b_x$.

V primeru, da je potrebno izračunati presečišče med lokom in daljico, nastavimo enačbo krožnice v splošni legi s središčem $\vec{c} = (c_x, c_y)$ in radijem R

$$(x - c_x)^2 + (y - c_y)^2 = R^2 \tag{19}$$

v katero vstavimo parametrično enačbo premice $\vec{p} = t\vec{n} + p_1$ v eksplicitni obliki, kar zapišemo kot

$$(tn_x + p_{1x} - c_x)^2 + (tn_y + p_{1y} - c_y)^2 = R^2 \tag{20}$$

Po razčlenitvi kvadratne enačbe (20) izračunamo

$$t_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{21}$$

kjer je

$$\begin{aligned} a &= |\vec{n}|^2 = n_x^2 + n_y^2 \\ b &= 2(\vec{n} \cdot (\vec{p_1} - \vec{c})) = 2(n_x(p_{1x} - c_x) + n_y(p_{1y} - c_y)) \\ c &= |\vec{p_1} - \vec{c}|^2 - R^2 = (p_{1x} - c_x)^2 + (p_{1y} - c_y)^2 - R^2 \end{aligned} \tag{22}$$

2 Datoteka geometrijskih podatkov

Za kostruiranje dvodimenzionalnih likov bomo uporabili programski paket Autocad v katerem bomo risali zaključene krivulje z ukazom `pline`. S programom `cnc.lsp` v autolispu kreiramo datoteko `cnc.txt` z naslednjim formatom:

```
n1
x1 y1 b1
x2 y2 b2
x3 y3 b3
n2
x1 y1 b1
x2 y2 b2
x3 y3 b3
x4 y4 b4
0
```

Številke `ni` določajo število posplošenih daljic lika. Posplošene daljice imajo koordinatame (x, y) in ukrivljenost $b = \tan\varphi/4$. `(load "cnc")` in ga zaženemo z ukazom `cnc`. S tem se narisana slika zapiše v datoteko `cnc.txt`.

3 Program za CNC stroj

Program za CNC stroj generiramo v G-kodi. Izsek daljšega programa z bistvenimi ukazi je naslednji:

```
N1      G90
N2      G17
N3      G00 X0.00 Y95.00 Z-10.02 F3000
N4      G01 X0.00 Y95.00 Z-3.02 F1000
N5      G01 X0.00 Y95.00 Z4.20 F1200
...
N39     G91
N40     G02 I-0.65 J-35.99 X35.33 Y-37.29
N41     G90
N42     G01 X190.00 Y183.43 Z35.00
...
N70     G03 I-5.00 J-8.43 X-5.00 Y1.37
N71     G90
...
N93     M30
```

Številke `N` pred ukazi označujejo programsko vrstico. Označbe vrstic za program CNC niso obvezne. Ukaz `G90` nastavi absolutno nastavljanje koordinat. `G17` nastavi interpolacijsko ravnino *x-y*. `G0` je hitri, `G1` pa delovni gib v izbrano točko [*mm*].

Vsakemu `G`-ukazu lahko priredimo hitrost gibanja `F` [*mm/min*]. Krožna interpolacija v smeri urinega kazalca (*CW*) je podana z ukazom `G2`. Koordinati *(I,J)* podajata relativno oddaljenost središča loka od zadnje točke (začetne točke loka). Koordinati *(X,Y)* podajata relativno oddaljenost končne točke loka od začetne točke loka.

Pred krožno interpolacijo z ukazom `G91` preklopimo v relativno naslavljanje koordinat. Krožna interpolacija v nasprotni smeri urinega kazalca (*CCW*) ima iste parametre, le da je ukaz `G3`. Ukaz `M30` konča program.

4 Prikaz potovanja orodja

Za grafičen prikaz rezultatov uporabite jezik GKS [3]. Program za prikaz mora uporabljati homogene transformacije v dvodimenzionalnem prostoru s kompozicijo matrik poljubnega vrstnega reda. Osnovni program, ki demonstrira uporabo jezika GKS je:

```
      program GKS skeleton
      implicit none
      real x(3) /-200.0, 350.0, 250.0/
      real y(3) /-200.0, 300.0,-250.0/
      real kx(2) /0.0, 100.0/
      real ky(2) /50.0, 120.0/
      character*80 datrec(1)
c     Odpre GKS (Gks OPen gKS)
      call gopks()
c     Nastavitev zaslonskih koordinat
c     (Gks Set WorKstation ViewPort)
      call gswkvp(1, 20.0, 50.0, 600.0, 450.0)
c     Nastavitev uporabniskega koord. sistema
c     (Gks Set WiNdow)
      call gswn(1, -300.0, -300.0, 400.0, 400.0)
c     Izris lika z dvema crtama
c     (GksSetPolyLineColorIndex, GksPolyLine)
c     Indeksi:1 modra, 2 zelena, 3 cyan, 4 rdeca,
c              5 violicna, 6 rjava, 7 bela, 8 siva,
c              9 svetlomodra, ..., 14 rumena, 15
      call gsplci(3)
      call gpl(3, x, y)
c     Izris teksta (GksSetCharacterHeight,
c     ...TextColorIndex, GksTeXt)
      call gstxci(14)
      call gschh(20)
      call gtx(-250.0, 300.0, 'GKS skeleton')
c     Izris kroga
c     (Generalized Drawing Primitive)
      call gdp(2, kx, ky, -1, 1, datrec)
c     Zapre GKS (Gks CLose gKS)
      call gclks()
      end
```

5 Poročilo

Poročilo se oddaja v dveh delih, prvi za numerični program, drugi za grafični del. V poročilu opišite problem, ki ga rešujete (zahteve), podajte teoretične osnove in bistvene dele rešitve, ki ste jih uporabili v programu.

Izpisa programa ni potrebno prilagati poročilu, potrebno pa je predstaviti pomembne dele programa na način, ki čimbolj jasno predstavlja rešitev. To je lahko z besednim opisom, diagramom poteka ali z delom programske kode. Priložite tudi rezultate,ki ste jih izračunali in.

Celotno poročilo naj ima klasično obliko (uvod, zahteve, teoretične osnove, problemi, rezultati, možne izboljšave, zaključek, literatura, priloge). Izberite si primer in na njem pokažite reševanje problema. Diskutirajte izboljšave programa, omejitve, degeneracije, dodatne izračune in napake programa, ki so posledica vgrajenih algoritmov (kontrole, velikost matrik, ...). Torej: poročilo mora biti napisano tako, da je možno iz njega rekonstrurirati program.

Pišite čitljivo; priporočen je izpis s tiskalnikom. Ovojnico za poročilo dobite pri vratarju (Računalniško podprto konstruiranje).

6 Delo na računalniku

Na računalniku PC je instalirano programsko okolje s fortranskim prevajalnikom po standardu F77. Okolje je integrirano in se imenuje *Programmers WorkBench*. Integrirano okolje z požene z ukazom pwb ime.for v katerem lahko program popravljamo, prevajamo, zaganjamo in iščemo napake. Osnovni ukazi so dosegljiv s pritiskom na tipko skupaj s tipko **ALT**. Tako je izhod iz **PWB ALT+F,X**, Prevajanje in zaganjanje programa z ukazom **Alt+R,E,B,R**. Za pomoč lahko uporabite tipko **F1**, ki je kontekstno občutljiva. Poleg okolja PWB je možno uporabiti tudi zmogljivejši 32-bitni prevajalnik *Fortran Power Station*, ki nima spominskih omejitev pri matrikah, žal pa deluje le v sistemu *Windows*, zato se priporoča najprej uporaba PWB za majhne sisteme, kasneje pa prevajanje s FPS za večje probleme in hitrejše izvajanje.

7 Splošna priporočila za delo

Noben program ne deluje na mah. Ker je problem kompleksen, se ga je potrebno lotiti sistematično. Posamezne dele program moramo sproti kontrolirati. Pomembno je tudi dokazovanje pravilnosti delovanja, kjer si logično razlagamo obnašanje programa za različne vhodne parametre. Švejkova metoda pri problemih, kot je naš, nikoli ni bila uspešna.

Fortran ima to (ne)ugodno lastnost, da so vse premenljivke ob začetku postavljene na nič. Po končanem delu brišite z (`del`) vse nepotrebne datoteke, ki jih imate na svojem področju.

Literatura

[1] Dragan Domazet. *Programiranje računarske grafike pomoću GKS sistema.* Naučna knjiga, Beograd, 1989.

[2] Divjak J. Dovedan Z. *Fortran 77 s tehnikama programiranja.* Zveza organizacij za tehnično kulturo Slovenije, Ljubljana, 1988.

[3] Kansy K. Plaff G. Enderle G. *Computer Graphics Programming.* Springer-Verlag, New York, 1984.

[4] Jon Rorke. The area of a simple polygon. In James Arvo, editor, *Graphics Gems II*, volume 2, pages 5–9. Academic Press, 1991.

Laboratorij za CAD, Soba N-17. Govorilne ure za študente so od ponedeljka do četrtka od 13^{00} do 14^{00}. Prosimo, da se držite razporeda urnika. Nadomeščanje zamujenih laboratorijskih vaj načeloma ni možno zaradi zasedenosti računalniške učilnice. Pred vajami na računalniku si pripravite gradivo za delo, da ne boste gledali v zrak (k sosedu).

Ljubljana 16. oktober 1995

Typeset in LaTeX 2ε