Laboratorijske vaje pri predmetu Računalniško podprto konstruiranje

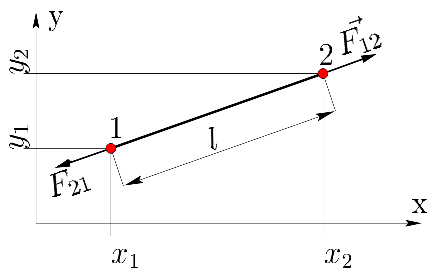
Leon Kos, Laboratorij za CAD - LECAD

Naloga

Izdelajte računalniški program za preračun prostorskih statično nedoločenih paličnih konstrukcij in program za prikaz rezultatov izračuna z uporabo grafičnega standarda GKS (level 1a). Programa morata biti ločeni zaključeni celoti (*processor*, *post-processor*), ki si posredujeta potrebne podatke preko binarne neformatirane datoteke podatkov (direktna datoteka). Prikaz konstrukcije v prostoru izvedite z homogenimi transformacijami poljubnega zaporedja. Za vsak program izdelajte poročilo z opisom delovanja programa, teoretičnimi osnovami in rezultati izračuna.

1 Deformacijska metoda končnih elementov

1.1 Element



Slika 1: Obremenjena palica konstantnega prereza

Slika 1 kaže osnovni element, ki se uporablja pri sestavljanju paličij. Obremenitev F je enakomerno porazdeljena po prerezu ploščine A. Zvezo med silo in raztezkom podaja enačba

$$F = \frac{AE}{l}\Delta l \tag{1}$$

Dolžina palice v 3D je

$$l = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2} \tag{2}$$

Če so raztezki majhni (v primerjavi z dolžino) lahko Δl nadomestimo s totalnim diferencialom dl, ki ga zapišemo kot

$$\begin{aligned} dl &= \frac{\partial l}{\partial x_1}dx_1 + \frac{\partial l}{\partial y_1}dy_1 + \frac{\partial l}{\partial z_1}dz_1 \\ &\quad + \frac{\partial l}{\partial x_2}dx_2 + \frac{\partial l}{\partial y_2}dy_2 + \frac{\partial l}{\partial z_2}dz_2 \end{aligned} \tag{3}$$

$$\begin{aligned} dl &= \frac{1}{l}(x_2-x_1)(dx_2-dx_1) \\ &\quad + \frac{1}{l}(y_2-y_1)(dy_2-dy_1) \\ &\quad + \frac{1}{l}(z_2-z_1)(dz_2-dz_1) \end{aligned} \tag{4}$$

Komponentne sile, ki delujejo v palici vozlišča 1 izračunamo tako, da enačbo (1) množimo s smernim kosinusom:

$$\begin{aligned} F_{12x} &= \frac{(x_2-x_1)}{l}\,\frac{AE}{l}dl \\ F_{12y} &= \frac{(y_2-y_1)}{l}\,\frac{AE}{l}dl \\ F_{12z} &= \frac{(z_2-z_1)}{l}\,\frac{AE}{l}dl \end{aligned} \tag{5}$$

Enačbe (5) so povezava med poljubnim pomikom vozliša (dq_i) in silo v palici. V prostoru ima vsaka palica šest možnih pomikov in šest komponentnih sil.

1.2 Enačba konstrukcije

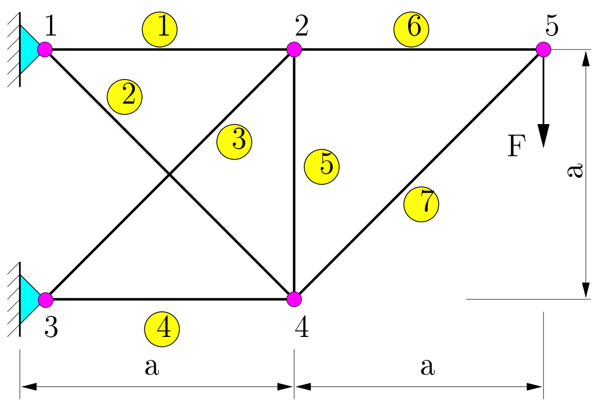
Ker imajo palice tudi skupna vozlišča lahko za vsako vozlišče napišemo ravnotežno enačbo

$$\sum_i \vec{F}_i = 0 \tag{6}$$

Če napišemo ravnotežno enačbo za vsa vozlišča, dobimo enačbo konstrukcije, ki ima obliko

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ F_{2x} \\ \vdots \\ F_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ k_{31} & k_{32} & \cdots & k_{3n} \\ k_{41} & k_{42} & \cdots & k_{4n} \\ \vdots & & \ddots & \vdots \\ k_{n1} & k_{b2} & \cdots & k_{nn} \end{bmatrix} \begin{Bmatrix} dx_1 \\ dy_1 \\ dz_1 \\ dx_2 \\ \vdots \\ d_n \end{Bmatrix} \tag{7}$$

Na levi strani enačbe so zunanje sile in neznane reakcije v podporah. Elementi k_{ij} tvorijo matriko togosti. Na desni strani so pomiki vozlišč. Nekateri pomiki v vozliščih so že vnaprej znani in jih zato ni potrebno računati v sistemu linearnih enačb.



Slika 2: Statično nedoločeno paličje; a = 1 m, F = 1 kN, AE = 4e6 N

Slika 2 kaže primer ravninskega statično nedoločenega paličja s petimi vozlišči. Velikost sistema enačb je tako $n = 2 \cdot 5 = 10$. Pripadajoča togostna matrika je

$$k_{ij} = \begin{bmatrix} -5.4 & 1.4 & 4 & 0 & 0 & 0 & 1.4 & -1.4 & 0 & 0 \\ 1.4 & -1.4 & 0 & 0 & 0 & 0 & -1.4 & 1.4 & 0 & 0 \\ 4 & 0 & -9.4 & -1.4 & 1.4 & 1.4 & 0 & 0 & 4 & 0 \\ 0 & 0 & -1.4 & -5.4 & 1.4 & 1.4 & 0 & 4 & 0 & 0 \\ 0 & 0 & 1.4 & 1.4 & -5.4 & -1.4 & 4 & 0 & 0 & 0 \\ 0 & 0 & 1.4 & 1.4 & -1.4 & -1.4 & 0 & 0 & 0 & 0 \\ 1.4 & -1.4 & 0 & 0 & 4 & 0 & -6.8 & 0 & 1.4 & 1.4 \\ -1.4 & 1.4 & 0 & 4 & 0 & 0 & 0 & -6.8 & 1.4 & 1.4 \\ 0 & 0 & 4 & 0 & 0 & 0 & 1.4 & 1.4 & -5.4 & -1.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.4 & 1.4 & -1.4 & -1.4 \end{bmatrix} \cdot 10^3 N \tag{8}$$

Matriko k_{ij} zgradimo tako, da za vse palice izračunamo sile v vozliščih in jih postavimo na ustrezno mesto (trojna zanka).

$$\begin{bmatrix} R_{1x} \\ R_{1y} \\ 0 \\ 0 \\ R_{3x} \\ R_{3y} \\ 0 \\ 0 \\ 0 \\ 1000 \end{bmatrix} \begin{bmatrix} -5.4 & 1.4 & \cdots & 0 & 0 \\ 1.4 & -1.4 & \cdots & 0 & 0 \\ 4 & 0 & \cdots & 4 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ 1.4 & -1.4 & \cdots & 1.4 & 1.4 \\ -1.4 & 1.4 & \cdots & 1.4 & 1.4 \\ 0 & 0 & \cdots & -5.4 & -1.4 \\ 0 & 0 & \cdots & -1.4 & -1.4 \end{bmatrix} \begin{bmatrix} dx_1 \\ dy_1 \\ dx_2 \\ dy_2 \\ dx_3 \\ dy_3 \\ dx_4 \\ dy_4 \\ dx_5 \\ dy_5 \end{bmatrix} \tag{9}$$

1.3 Robni pogoji

Linearni sistem enačb (9) je predoločen, saj so nakateri pomiki že vnaprej znani ($dx_1 = 0$, $dy_1 = 0$, $dx_3 = 0$, $dy_3 = 0$). Niso pa poznane velikosti reakcij. Enačbo (9) lahko preuredimo s permutacijo vrtic in kolon v obliko

$$\begin{Bmatrix} \mathbf{F}_m \\ \mathbf{F}_r \end{Bmatrix} = \begin{bmatrix} \mathbf{K}_{mm} & \mathbf{K}_{mr} \\ \mathbf{K}_{rm} & \mathbf{K}_{rr} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_m \\ \mathbf{U}_r \end{Bmatrix} \tag{10}$$

v kateri so v spodnjem delu enačb (10) poznanimi pomiki $\mathbf{U}_r = 0$. Zunanje sile v vozliščih so $\mathbf{F}_m$, reakcije v podporah pa $\mathbf{F}_r$. Zaradi ničnosti vektorja $\mathbf{U}_r$ se enačba (10) poenostavi v

$$\mathbf{F}_m = \mathbf{K}_{mm} \cdot \mathbf{U}_m \tag{11}$$

iz katere lahko izračunamo neznane pomike $\mathbf{U}_m$ z enačbo

$$\mathbf{U}_m = \mathbf{K}_{mm}^{-1} \cdot \mathbf{F}_m \tag{12}$$

Po izračunanih pomikih se reakcije v podporah izračuna kot

$$\mathbf{F}_r = \mathbf{K}_{rm} \cdot \mathbf{U}_m \tag{13}$$

Sile v palicah pa izračunamo iz osnovne enačbe elementa (1 ali 5).

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1000 \end{bmatrix} = \begin{bmatrix} -6828 & 0 & 0 & 0 & 1414 & 1414 \\ 0 & -6828 & 0 & 4000 & 1414 & 1414 \\ 0 & 0 & -9414 & -1414 & 4000 & 0 \\ 0 & 4000 & -1414 & -5414 & 0 & 0 \\ 1414 & 1414 & 4000 & 0 & -5414 & -1414 \\ 1414 & 1414 & 0 & 0 & -1414 & -1414 \end{bmatrix} \begin{bmatrix} dx4 \\ dy4 \\ dx2 \\ dy2 \\ dx5 \\ dy5 \end{bmatrix} \tag{14}$$

Za primer na sliki 2 je sistem linearnih enačb (14) dobljen s permutacijo vrstic in kolon. Po izračunu smo dobili naslednje rezultate pomikov in reakcij

$$\begin{array}{l|l} dx_4 = -0.3894\,mm & dy_4 = -0.7838\,mm \\ dx_2 = 0.3606\,mm & dy_3 = -0.6733\,mm \\ dx_5 = 0.6106\,mm & dy_5 = -2.491\,mm \\ R_{1x} = -2000\,N & R_{1y} = 557.8\,N \\ R_{3x} = 2000\,N & R_{3y} = 442.2\,N \end{array} \tag{15}$$

2 Programska modula

2.1 Procesor

Prebere podatke iz tekstne datoteke, ki jo uporabnik napiše z editorjem v predpisani obliki. Pravilnost podatkov se sproti kontrolira in izpisujejo se napake ter opozorila pri interpretiranju vhodne datoteke. Ob napaki v interpretaciji mora program izpisati vrstico v kateri se je pojavlja napaka in vrsto napake. Vhodna datoteka naj ima končnico `.TXT`. Po preračunu pa naj program naredi dve datoteki; eno v tekstni (`.PRN`) s tabelaričnim izpisom izračuna in drugo v binarni obliki (`.BIN`) z direktno datoteko za posredovanje podatkov postprocesorju. Program naj omogoča izračun ravninskega in prostorskega paličja z najmanj 100 vozlišči.

2.1.1 Vhodna datoteka

Format datoteke je podoben programom STRESS (*STRuctural Engineering System Solver*). Vhodna datoteka je tekstovna s ključnimi besedami katerim sledijo numerični podatki.

structure ***naziv*** Naziv je le kot komentar.

plane ali space določa dimenzijo koordinatnega sistema ravnina ali prostor. Če imamo ravninski problem, potem ne podajamo koordinate *z*.

joints ***število vozlov***

members ***število palic***

reactions ***število reakcij***

loads ***število obremenitev (sil)***

coordinates sledi seznam koordinat vozlišč z zacetnim vozlom št. 1 v obliki:
`št. vozla koord-x koord-y (koord-z)`

incidences sledi seznam palic (palice se začnejo šteti z 1) z navedenimi povezavami vozlišč v obliki:
`št.-palice začetni-vozel končni-vozel lastnost-`$A \cdot E$ AE je zmnožek površine prereza palice in elastičnega modula materiala palice.

supports sledi seznam vozlišč v katerih so podpore in smeri delovanja reakcij v obliki:
`številka vozlišča črka x ali y ali z`

forces sledi seznam sil v vozliščih in komponente sil v posameznih smereh v obliki:
`številka vozlišča sila-x sila-y (sila-z)`

solve pove programu, da je konec podatkov in naj konča z interpretiranjem.

Primer vhodne datoteke za statično nedoločen nosilec v ravnini (slika 2).

```
structure MDT2 p.479/16
plane
joints 5
members 7
reactions 4
loads 1

coordinates
1      0.0  1000.0
2   1000.0  1000.0
3      0.0     0.0
4   1000.0     0.0
5   2000.0  1000.0

incidences
1 1 2 4e6
2 1 4 4e6
3 2 3 4e6
4 3 4 4e6
5 2 4 4e6
6 2 5 4e6
7 4 5 4e6

forces
5 0.0 -1000.0

supports
1 x
1 y
3 x
3 y

solve
```

2.1.2 Tekstovna izhodna datoteka

Tekstovna izhodna datoteka naj izpiše vse rezultate preračuna v tabelirani obliki. Palice, ki so obremenjene na tlak imajo po dogovoru negativen predznak sile. Reakcije v podporah naj bodo tabelirane tako, da se bo vedelo v katerem vozlu je podpora in kakšna je velikost. Predznak (usmerjenost) sile v podporah se določi glede na globalni koordinatni sistem.

2.1.3 Binarna izhodna datoteka

Binarna izhodna datoteka se bo uporabila za posredovanje podatkov postprocesorju. Kateri podatki so potrebni za nadaljno obdelavo ocenite sami. (npr. kordiante vozlišč, pomiki, št. palic, vozlišč, sile v pali-cah, ...). Ime datoteke naj bo fiksno (npr. `palicje.bin`), tako da poprocesor ne bo spraševal po vhodni datoteki ampak samo preverjal če obstaja.

2.2 Postprocesor

Postprocesor je program, ki omogoča prikaz in nadaljno obdelavo izračunanih podatkov (za razliko od preprocesorja, ki bi npr. omogočal interaktiven vnos podatkov za preračun).

Za grafičen prikaz rezultatov v programu uporabite grafični jezik GKS ali PHIGS. Poprocesor naj omogoča prikaz konstrukcije v treh dimenzijah s poljubnim zaporedjem transformacj v prostoru. V ta namen uporabi homogene matrike in iz njih sestavi kompozicijsko matriko. Projekcija na zaslon naj bo ortogonalna projekcija ravnine x-y. Ravninski nosilci imajo koordinato z=0. Na zahtevo (menu) mora program izrisati na konstrukcijo še označbe vozlov, palic, sile v palicah oziroma pomike. Ker so pomiki vozlišč majhni, jih je potrebno množiti s konstanto, da dobimo pretiran prikaz deformiranega paličja. Palice, ki so obremenjene na tlak izriši v rdeči barvi, natezno obremenjene palice pa v zeleni. Risanje sil in reakcij v vozliščih je neobvezno.

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
```

3 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 primerov, ki ste jih izračunali.

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).

4 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.

5 Fortran

Tu bom prikazal standardne posebnosti v Fortranu, ki jih boste s pridom uporabili pri programiranju. Priporoča se tipkanje programa z malimi črkami, saj je tako kodiran program bolj čitljiv.

Formatne stavke vključene lahko vključimo v ukaz `read` ali `write`:

```
character ime*65
write(*, '('' Ime vhodne datoteke: '', $)' )
read(*,'(A)') ime
```

Datoteke delimo na tekstovne in binarne datokeke. Direktna datoteka je poseben primer binarne datoteke, ki ima fiksno dolžino zapisa (record). Fortran omogoča branje/pisanje v prostem formatu (*), uporabniškem formatu (format()) in brez formata (samo v binarne datoteke). Neformatirano direktno datoteko z dolžino enega zapisa 58 bytov se odpre z ukazom

```
  open(1, 'results.bin', status='NEW',
* access='DIRECT', form='UNFORMATTED',
* recl=58)
```

Dolžina enega zapisa RECL se določa z maksimalno velikostjo zapisa v datoteki. Zapisi imajo lahko različne vsebine podatkov, dolžina zapisa pa je fiksna zaradi hitrejšega določanja pozicije datotečnega kazalca. Določi se jo s seštevkom dolžine vseh spremenljivk in upoštevanjem, da zavzame realna spremenljivka 4 byte, celoštevilčna 2, znakovna pa toliko, kolikor znakov ima rezerviranih.

Za branje in pisanje enote se uporablja standardni read/write stavek brez podatka o formatu.

```
write(1, rec=2) (x(i), i = 1, n/2)
read(1, rec=2)  (x(i), i = 1, n/2)
```

Pri vseh branjih in pisanjih se lahko testira napaka ob konverziji ali napaka na vhodno/izhodno enoti s parametrom err. Primer:

```
    ...
    read(line(6:80), '(I10)', err=102) nodes
    ...
    return
102 stop 'Napaka pri branju parametra loads!'
    end
```

Fortran omogoča pisanje in branje v svoje interne enote v obliki `character` polja. To nam omogoča pretvarjanje numeričnih spremenljivk v znakovne. Največkrat se to uporablja za podprograme, ki zahtevajo kot parameter znakovno spremenljivko, na voljo pa imamo le numerično spremenljivko. Tipičen primer uporabe je izpis teksta v jeziku GKS. Primer:

```
character cnode*3
integer node
...
write(cnode, '(I3)') node
call gtx(x, y, cnode)
end
```

6 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] William H. et.al. *Numerical Recipes in Fortran; The Art of Scientific programing.* Cambridge university press, Cambridge, 1989.

[4] Ervin Prelog. *Metoda končnih elementov.* FAGG Ljubljana, 1975. FS 16528.

Ljubljana 20. februar 1996

Typeset in $\LaTeX\ 2_\varepsilon$