Laboratorijske vaje pri predmetu Postopki konstruiranja

Leon Kos, Laboratorij za CAD - LECAD

Naloga

Izdelajte računalniški program za hidravlični preračun fluovodnih mrež v obliki prstanov in program za prikaz rezultatov izračuna z uporabo grafičnega standarda GKS (level 1a).

Programa morata biti ločeni zaključeni celoti (preračun in prikaz), ki si posredujeta potrebne podatke preko datoteke podatkov, ki naj se izvede z neformatirano (binarno) direktno datoteko. Za vsak program izdelajte poročilo z opisom delovanja programa, omejitvami, teoretičnimi osnovami in rezultati izračuna.

1 Uvod

Cevovodi predstavljajo zelo primerno in ekonomično sredstvo za transport fluidov, posebno ko imamo opravka z velikimi kapacitetami fluidov in velikimi razdaljami, kar se tiče transporta. Uvrstimo jih lahko med nepremična transportna sredstva, saj ne zahtevajo energije za morebitno lastno premikanje. Črpalke in kompresorje, ki so tako kot ostali delovni stroji, ki se uporabljajo za ustvarjanje tlačne energije, nepremični, v tem smislu obravnavamo enako kot cevovode. Od vseh vrst fluidov se s cevovodi najpogosteje transportira hladna in topla voda, nafta in njeni derivati, razne kisline v kemijski in procesni industriji, naravni plin, vodna para, razne snovi iz prehrambene industrije, energetski fluidi v termoenergetskih postrojenjih, itd.

2 Vrste fluidovodnih mrež

Poznamo tri tipe mrež:

- v obliki drevesa oz. razvejan tip
- v obliki zank (prstanov)
- kombinacija zgornjih dveh tipov

2.1 Fluidovodna mreža v obliki drevesa

Tovrstna mreža se sestoji iz magistralnih oz. glavnih vodov ter velikih in malih vej. Uporablja se za oskrbovanje manjšega števila medsebojno oddaljenih uporabnikov, ki dopuščajo tudi kratkotrajna prekinjanja v dobavljanju, ki lahko nastanejo zaradi pokanja cevovodov ali kakšne druge okvare.

2.2 Fluidovodna mreža v obliki prstanov

Mreža se je zgrajena iz niza zaprtih prstanov (zank). Napajanje se izvaja na enem ali več mestih preko magistralnih vodovodov, kar je odvisno od velečine mreže, razporeditve uporabnikov in njihove porabe.

Mestne vodovodne mreže se danes gradijo v glavnem s tovrstnimi mrežami, ker omogočajo dvosmerno dobavljanje z vodo in ne zahtevajo prekinitve dela cele mreže v času izključitve njenih posameznih delov. Vodovodne mreže protipožarne zaščite se gradijo izključno v obliki prstanov.

3 Hidravlični preračun

V primeru projektiranja nove fluidovodne mreže se hidravlični preračun sestoji v glavnem iz določevanja potrebnih prerezov cevi in pritiska na mestih napajanja mreže ob upoštevanju pogoja, da skozi posamezne cevi teče določena količina fluida potrebnega pritiska.

Lahko pa naletimo tudi na drugo vrsto naloge in sicer, da so določeni posamezni premeri cevi in dotoki oz. odtoki iz mreže, naša naloga pa je določiti stanje v mreži, to pomeni določiti pretoke v ceveh in tlake v posameznih vozliščih. Potrebno je torej narediti analizo dogajanja v mreži.

Osnovna enačba hidrodinamike je seveda *Bernoulijeva enačba* za stacionarni tok nestisljivega fluida za sistem s trenjem 153

$$p_1 + \rho\frac{v_1^2}{2} + \rho g h_1 = p_2 + \rho\frac{v_2^1}{2} + \rho g h_2 + \Delta p_t \qquad (1)$$

Ker se prerez v cevi ne spreminja je hitrost fluida konstantna, tako lahko napišemo enačbo za tlak na izhodu

$$p_2 = p_1 + \rho g(h_2 - h_1) - \Delta p \qquad (2)$$

V ceveh se lahko pojavlja laminarni ali turbulenten tok 154

$$Q = vA = \frac{v\pi d^2}{4} \quad \Longrightarrow \quad v = \frac{4Q}{\pi d^2}$$

katerega razmejuje Reynoldsovo število 156

$$Re = \frac{vd}{v} = \frac{4Q}{\pi v d}$$

$$Re = \begin{cases} < 2320 & - \quad \text{laminarni tok} \\ > 2320 & - \quad \text{turbulentni tok} \end{cases},$$

kjer je Q pretok v cevi premera d in υ kinematična viskoznost fluida (voda pri 293K : $\upsilon = 1.01 mm^2/s$).

Ne glede na to pred kakšno nalogo stojimo in s kakšnim tipom mreže imamo opravka, morata biti izpolnjena sledeča dva zakona hidravlike:

1. vsota vseh tokov, ki pritekajo v vozlišče in vseh tokov, ki odtekajo iz vozlišča mora biti enaka nič.
$$\sum_i Q_i = 0 \tag{3}$$
Podobno lahko vzamemo celotno omrežje kot eno vozlišče, zato mora biti vsota vseh dotokov in odtokov prav tako enaka nič. Če to ne bi bilo tako, bi najverjetneje omrežje nekje puščalo.

2. vsota padcev tlaka po zaključeni zanki mora biti prav tako enaka nič. Statični tlak ne vpliva na zaključeno zanko, kot tudi ne višinske razlike, zato je zančna enačba odvisna le od izgub v ceveh.
$$\sum_i \Delta p_i(Q) = 0 \tag{4}$$

Izgubo tlaka zaradi odpora pri toku fluida z gostoto ρ in hitrostjo v izračunamo z enačbo

$$\Delta p = \frac{\zeta \rho v^2}{2}$$

ζ je koeficient izgub, ki je pri ravnih ceveh s krožnim prerezom

$$\zeta = \frac{\lambda l}{d}$$

λ je koeficient trenja, l je dolžina cevi, d je premer cevi.

Enačbo za izgubo (padec) tlaka v cevi lahko tako
157 napišemo

$$\Delta p(Q) = \frac{8\lambda l \rho}{\pi^2 d^5} Q^2$$

Če bi napisali sistem enačb za mrežo tokov, bi dobili nelinearni sistem enačb saj je padec tlaka kvadratično odvisen od pretoka $\Delta p = f(Q^2)$ in s tem praktično nerešljiv. Pomagamo si z linearizacijo enačb(4). V okolici delovne točke Q_d se nelinearna funkcija obnaša linearno. Pretok lahko napišemo kot

$$Q = Q_d + \Delta Q = Q_d\left(1 + \frac{\Delta Q}{Q_d}\right) \tag{5}$$

Funkcijo $\Delta p(\Delta Q)$ razvijemo v Taylorjevo vrsto z
45 uporabo enačbe

$$(1+x)^n = 1 + \binom{n}{1} x + \binom{n}{2} x^2 + \binom{n}{3} x^3 + ...$$

$$(1+x)^2 = 1 + 2x + x^2 + ... \quad , \quad |x| < 1$$

in upoštevamo le linearni člen, saj naj bi bil neznan odmik ΔQ od fiksne delovne točke majhen in s tem napaka pri linearizaciji dovolj majhna.

$$\Delta p(\Delta Q) = \frac{8\lambda l \rho}{\pi^2 d^5} Q_d^2 \left(1 + 2\frac{\Delta Q}{Q_d}\right) \tag{6}$$

Drugače zapisana enačba (6) pokaže linearno odvisnost padca tlaka Δp od male spremembe ΔQ

$$\Delta p(\Delta Q) = C Q_d^2 + 2 C Q_d \Delta Q \tag{7}$$

$$C = \frac{8\lambda l \rho}{\pi^2 d^5}$$

3.1 Koeficient trenja λ

1. V laminarnem področju ($Re < 2320$) je λ odvisen le od Reynoldsovega števila in je
$$\lambda = \frac{64}{Re} = \frac{64\pi \upsilon d}{4Q} \tag{8}$$

2. Pri turbulentnem toku se v praksi največ uporablja Colebrookova v prehodnem področju 157
$$\frac{1}{\sqrt{\lambda}} = -2log_{10}\left(0.269\frac{k}{d} + \frac{2.51}{Re\sqrt{\lambda}}\right) \quad ,$$
in je pri hitrosti $Re > 10^3 d/k$ praktično konstantna. Računanje točne vrednosti λ transcendentne enačbe lahko enostavno izvedemo z iterativno formulo
$$\lambda_{n+1} = \frac{1}{\left(-2log_{10}\left(0.269\frac{k}{d} + \frac{2.51}{Re\sqrt{\lambda_n}}\right)\right)^2} \tag{9}$$
in začetnim približkom $\lambda = 0.04$. Iterativni proces končamo ko je relativna napaka dovolj majhna $\epsilon = 0.001$
$$\frac{|\lambda_{n+1} - \lambda_n|}{\lambda_{n+1}} < \epsilon$$

λ je kot vidimo iz enačb v praktičnih primerih, ko je $k/d \approx 0.005$ in $Re \approx 10^5$ skoraj konstantna, zato lahko λ v enačbah (7) vzamemo kot konstanto. V najslabšem primeru, ko je tok laminaren je $\Delta\lambda(\Delta Q = 10\%) < -9\%$. V turbulentnem področju je to odstopanje največ $\Delta\lambda(\Delta Q = 10\%) < -2\%$, v praksi pa približno -0.1%, kar je skoraj konstantno. Z λ v enačbah (7) lahko računamo kot konstanto po celem obsegu (laminarno in turbulentno) tudi zato, ker je napaka $\Delta\lambda$ vedno negativna in majhna, kar v najslabšem primeru le zmanjšuje ΔQ in s tem nekoliko poveča število iteracij, nikakor pa to ne vnaša divergence v linearizirann sistem enačb.

4 Sistem enačb

Iz osnovnih enačb (3 in 4) z uporabo lineariziranih enačb (5 in 7) zapišemo nov sistem enačb

$$\sum_i \Delta Q_i = Q_o - \sum_i Q_{di} \tag{10}$$

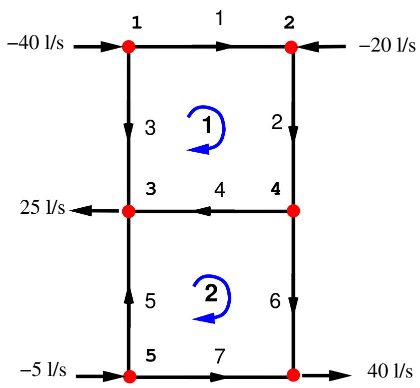
$$\sum_i 2C_i Q_{di} \Delta Q_i = -\sum_i C Q_{di}^2 \tag{11}$$

Q_o je odtok v vsakem vozlišču za katerega napišemo vrstico sistema enačb. Kako določiti začetne delovne točke? Ena od možnosti je, da nastavimo sistem enačb s predpostavko o laminarnem pretoku v vseh ceveh. Tak sistem je linearen in enostavno rešljiv. Izkaže se, da ni potrebno natančno računati delovnih točk pretokov v ceveh, ampak je dovolj, če predpostavimo, da v vseh ceveh teče povprečje vseh dotokov v vozliscih. Ker je reševanje lineariziranega sistema enačb **iterativno** se Q_{di} po nekaj iteracijah postavi v pravo lego. Začetne delovne točke se lahko izračuna kot

$$Q_d{}^0 = \frac{1}{n}\sum_i Q_{oi} \tag{12}$$

4.1 Tvorjenje sistema enačb

Prikažimo tvorjenje sistema enačb na primeru podanem s sliko 1. Vse cevi so premera d = 100 mm, relativne hrapavosti k = 0.5 mm in dolžine l = 50 m. Fluid je voda pri 293 K; $\rho = 1000 kg/m^3$, $\upsilon = 1.0110^{-6} m^2/s$.



Slika 1: Omrežje s prstani

Vozliščna enačba v prvem vozlišču izgleda takole:

$$\Delta Q_1 + \Delta Q_3 = Q_{o1} - (Q_{d1} + Q_{d3})$$

Prva zančna enačba pa je sledeča:

$$\begin{aligned} &2C_1 Q_{d1}\Delta Q_1 + 2C_2 Q_{d2}\Delta Q_2 + \\ &-2C_3 Q_{d3}\Delta Q_3 + 2C_4 Q_{d4}\Delta Q_4 = \\ &= -(C_1 Q_{d1}^2 + C_2 Q_{d2}^2 - C_3 Q_{d3}^2 + C_4 Q_{d4}^2) \end{aligned}$$

Podobno bi lahko napisali enačbe še za ostala vozlišča. Tako je možno napisati šest vozliščnih enačb in tri zančne enačbe. Ker imamo le sedem neznanih pretokov so nekatere enačbe v sistemu linearno odvisne. Ena odvečna enačba je pri skupini vozliščnih enačb saj je odtok v šestem vozlišcu popolnoma določen, če poznamo ostale odtoke v vozliščih. Odvečna zančna enačba pa je tista, ki ne prinaša v sistem enačb nobenega novega podatka o pretokih v ceveh. V našem primeru je to zanka ki obide vse zunanje cevi. Linearni sistem je rešljiv, če imamo n neznank in n linearno neodvisnih enačb.

$$\text{Zank} \equiv \text{Cevi} - \text{Vozlov} + 1$$

4.2 Smerna matrika

Pri tvorjenju sistema enačb si lahko pomagamo z celoštevilčno smerno matriko, ki nam pove smeri pretokov in logično smer pretoka v zanki. V našem primeru je prvih pet vrstic osnova za vozliščne enačbe, zadnji dve pa sta osnovi za zančni enačbi.

$$M = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Ta matrika se ne spreminja preko iteracij in je osnova za gradnjo matrike sistema. Vozliščne enačbe so že kar pravilne, zančne pa je še potrebno pomnožiti z ustreznimi konstantami.

4.3 Iteracija

Ker delovne točke po izračunu sistema enačb niso bile dovolj točne, jih je potrebno popraviti na pravo vrednost. Za koliko je bil začetni približek napačen pa nam pove ravno ΔQ_i, ki je nekakšna ocena napake računa sistema nelinearnih enačb.

$$Q_{dn+1} = Q_d + \Delta Q$$

Delovne točke popravljamo toliko časa, dokler ne bo napaka dovolj majna. Za oceno majhnosti celotne napake lahko izberemo enačbo

$$error = \frac{1}{n}\sum_i \left|\frac{\Delta Q_i}{Q_i}\right| < \epsilon$$

Za nas bo rezultat dovolj točen, če bo absolutna napaka manjša od 1% ($\epsilon < 0.01$) in takrat bomo prekinili iteracijski proces.

5 Program

Program 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. Vhodna datoteka naj ima končnico .TXI. Po preračunu pa naj program izpiše na zaslon rezultate obdelave, kot so: vhodni podatki, pretoki, hitrosti, Reynoldsovo število, število iteracij ... Kasneje pa bo potrebno rezultate zapisati še v binarni obliki z direktno datoteko za posredovanje podatkov postprocesorju. Ne trudite se izpisovati rezultate še v tekstovno datoteko, saj to omogoča že sam operacijski sistem z ukazom kot je

```
echo podatki.txi | flow > rezultati.txt
```

Vhodna datoteka naj ima sledeč format:

```
constants 1000 1.01e-6  # gostota, viskoznost
junctions 6             # i, x, y, z, Qi
1   0.0 100.0 1.0 -0.040
2  50.0 100.0 2.0 -0.020
3   0.0  50.0 0.5  0.025
4  50.0  50.0 1.5  0.000
5   0.0   0.0 1.7 -0.005
6   0.0  50.0 0.0  0.040
pipes 7 # i, za\v{c}, kon, l[m], d[m], k[m]
1 1 2 50.0 0.1  0.0005
2 2 4 50.0 0.1  0.0005
3 1 3 50.0 0.1  0.0005
4 4 3 50.0 0.1  0.0005
5 5 3 50.0 0.1  0.0005
6 4 6 50.0 0.1  0.0005
7 5 6 50.0 0.1  0.0005
loops 2 # \v{s}t. cevi {smer cevi ...}
4  1  2  4 -3
4 -4  6 -7  5
solve
```

Začetne delovne točke se izračuna kot povprečje vseh pretokov $Q_d^0 = 1/6(|-0.04| + |-0.02| + |0.025| + |-0.005| + |0.04| = 0.018571$ kar je osnova za začetne približke pretokov.

$$Q_{d_i} = 0.01857 \qquad i = 1, 2, ..7$$

Celoten začetni sistem enačb z upoštevanjem enačb (10 in 11) ima obliko

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ R & R & -R & R & 0 & 0 & 0 \\ 0 & 0 & 0 & -R & R & R & -R \end{bmatrix} \cdot \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \Delta Q_3 \\ \Delta Q_4 \\ \Delta Q_5 \\ \Delta Q_6 \\ \Delta Q_7 \end{bmatrix} = \begin{bmatrix} 0.003 \\ 0.020 \\ 0.030 \\ -0.01 \\ -0.03 \\ -1596 \\ 0.000 \end{bmatrix}$$

Konstanta $R = 85927$ ima to vrednost le pri prvem izračunu sistema enačb, kasneje pa se matrika pri iteraciji spreminja (le zančni del enačb). Sistem enačb je linearen. Neznane vrednosti so ΔQ_i, ki jih izračunamo z eno od metod za reševanje sistema linearnih enačb.

Rezultat sedmih iteracij so pretoki Q_d delovne točke z relativno natančnostjo 1%

$$Q_d = (0.0097, 0.0297, 0.0302, 0.0083, -0.0136, 0.0213, 0.0186)$$

5.1 Prikaz rezultatov

Za grafičen prikaz rezultatov v programu uporabite GKS [3] grafični jezik. Postprocesor naj omogoča prikaz omrežja v dveh dimenzijah s poljubnim zaporedjem transformacj. V ta namen uporabi homogene matrike.

Na zahtevo mora program izrisati na omrežje še tekstovne označbe vozlov, cevi, pretokov v ceveh ali hitrosti v ceveh. Cevi, ki imajo napačno predpostavljeno smer gibanja fluida izriši v rdeči barvi, ostale pa v zeleni. Risanje smeri pretoka je neobvezno.

6 Poročilo

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

Diskutirajte probleme pri izdelalavi bolj uporabnega programa z kombinirano metodo cevi (absolutni tlaki, izgube v vozlih in na liniji), omejitve programa (velikost sistema) in algoritma (natančnost, stisljivost, temperatura, konvergenca, kavitacija).

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 kode v fortranu. Priložite tudi rezultate izračuna mrež, ki ste jih izračunali. Na konkretnem primeru je potrebno prikazati delovanje programa in dokazati, da so rezultati točni.

Celotno poročilo naj ima klasično obliko (uvod, zahteve, teoretične osnove, problemi, rezultati, možne izboljšave, zaključek, literatura, priloge). Izberite si primer omrežja in na njem pokažite reševanje problema. Pišite čitljivo; priporočen je izpis s tiskalnikom. Ovojnico za poročilo dobite pri vratarju (Računalniško podprto konstruiranje).

7 Delo na računalniku

Na računalniku PC je instalirano programsko okolje s fortranskim prevajalnikom po standardu F77.

Okolje je integrirano in se kliče *Programmers WorkBench*. Ob začetku dela na računalniku je potrebno najprej narediti uporabniški direktorij z ukazom mkdir priimek.ime. Sledi sprememba tekočega direktorija z ukazom cd priimek.ime Integrirano okolje z požene z ukazom pwb flow.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.

8 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.

Če želimo interpretirati (parsing) vhodne podatke ki imajo (na)ključne besede skupaj s številčnim parametrom je potrebno najprej prebrati celotno vrstico kot niz znakov in ga nato interpretirati glede na vrsto ključne besede. Naslednji primer programa prebere standardno vhodno enoto (tipkovnica) in izpiše parameter, če je na začetku vrstice vtipkana ključna beseda `pipes`.

```
      program stringReadDemo
C     demonstrira uporabo branja iz ene vrstice
C     in primerjanje klju\v{c}ne besede z parametrom
      character*80 line
      read(*,'(A)') line
      if(line(1:5) .eq. 'pipes') then
         read(line(6:80), '(I10)') pipes
         print *, 'Pipes=', pipes
      endif
      stop
      end
```

Navedeni primer vsebuje formatne stavke vključene kar v ukaz read, kar je ugodno, če imamo male formatne stavke. Podobno bi lahko napisali formatni stavek za ukaz read/write:

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

Datoteke delimo na tekstne 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) cev,x,y,z,Q,v
       read(1, rec=2)  cev,x,y,z,Q,v
```

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) pipes
      ...
      return
102   stop 'Napaka pri branju parametra pipes!'
      end
```

Celoten program za izračun mreže bi lahko vseboval sledeče zaporedje podprogramov:

- Interpretiranje vhodnih podatkov in pisanje v ustrezne spremenljivke
- Kontrola določenosti sistema (rešljivost)
- Tvorjenje smerne matrike in začetnih delovnih točk
- Iterativen proces z izračunom sistema linearnih enačb
- Izpis podatkov na zaslon in prepis vseh potrebnih podatkov v direktno datoteko.

9 Reševanje sistema enačb

Sistem linearnih enačb lahko rešujemo na različne načine. Klasična Gaussova eliminacijska metota se pri konkretnih problemih pokaže kot počasna (N^3 operacij). Predlagana metoda, ki se v praksi tudi največ uporablja za reševanje sistema enačb je Lower/Upper dekompozicija, ki ima časovno zahtevnost $1/3N^3$. Reševanje sistema po tej metodi se sestoji iz dveh korakov:

1. **decomposition** razdeli matriko **M** na dve matriki (zgornja / spodnja), katerih produkt je **M**.) Obe matriki sta shranjeni v matriki M, le da je zgornji del matrike **M** matrika **U**, spodnji pa matrika **L**.

2. **backsubstitution** množi desno stran enačbe z zgornjo matriko in pri tem izračuna neznane linearne spremenljivke.

9.1 Lower/Upper dekompozicija

Rešujemo sistem enačb velikosti N. Matrika A predstavlja levo stran sistema enačb. Pri fortranu so lahko matrike le statične in zato parameter NP opisuje velikost matrike A, ki je običajno večja kot pa sistem enačb ($N < NP$). INDX je celočtevilčni vektor permutacji v matriki A in se prenaša naprej tako kot parameter D v podprogram $LUBKSB$, kateri zahteva še desno stran sistema enačb v vektorju B. Po izračunu se rezultat nahaja v vektorju B.

Potek izračuna sistema linearnih enačb bi potekal nekako takole:

```
parameter (NP=100)
real a(NP,NP), dQ(NP), d
integer indx(NP)
...
...
n = pipes
...
...
call ludcmp(a, n, NP, indx, d)
call lubksb(a, n, NP, indx, dQ)
...
```

Potrebna podprograma sta izpisana iz knjige [4]

```
      SUBROUTINE LUDCMP(A,N,NP,INDX,D)
      PARAMETER (NMAX=100,TINY=1.0E-20)
      DIMENSION A(NP,NP),INDX(N),VV(NMAX)
      D=1.
      DO 12 I=1,N
        AAMAX=0.
        DO 11 J=1,N
          IF (ABS(A(I,J)).GT.AAMAX)
     *          AAMAX=ABS(A(I,J))
11      CONTINUE
        IF (AAMAX.EQ.0.)
     *        PAUSE 'Singular matrix.'
        VV(I)=1./AAMAX
12    CONTINUE
      DO 19 J=1,N
        IF (J.GT.1) THEN
          DO 14 I=1,J-1
            SUM=A(I,J)
            IF (I.GT.1)THEN
              DO 13 K=1,I-1
                SUM=SUM-A(I,K)*A(K,J)
13            CONTINUE
              A(I,J)=SUM
            ENDIF
14        CONTINUE
        ENDIF
        AAMAX=0.
        DO 16 I=J,N
          SUM=A(I,J)
          IF (J.GT.1)THEN
            DO 15 K=1,J-1
              SUM=SUM-A(I,K)*A(K,J)
15          CONTINUE
            A(I,J)=SUM
          ENDIF
          DUM=VV(I)*ABS(SUM)
          IF (DUM.GE.AAMAX) THEN
            IMAX=I
            AAMAX=DUM
          ENDIF
16      CONTINUE
        IF (J.NE.IMAX)THEN
          DO 17 K=1,N
            DUM=A(IMAX,K)
            A(IMAX,K)=A(J,K)
            A(J,K)=DUM
17        CONTINUE
          D=-D
          VV(IMAX)=VV(J)
        ENDIF
        INDX(J)=IMAX%
        IF(J.NE.N)THEN
          IF(A(J,J).EQ.0.)A(J,J)=TINY
          DUM=1./A(J,J)
          DO 18 I=J+1,N
            A(I,J)=A(I,J)*DUM
18        CONTINUE
        ENDIF
19    CONTINUE
      IF(A(N,N).EQ.0.)A(N,N)=TINY
      RETURN
      END

      SUBROUTINE LUBKSB(A,N,NP,INDX,B)%
      DIMENSION A(NP,NP),INDX(N),B(N)
      II=0
      DO 12 I=1,N
        LL=INDX(I)
        SUM=B(LL)
        B(LL)=B(I)
        IF (II.NE.0)THEN
          DO 11 J=II,I-1
            SUM=SUM-A(I,J)*B(J)
11        CONTINUE
        ELSE IF (SUM.NE.0.) THEN
          II=I
        ENDIF
        B(I)=SUM
12    CONTINUE
      DO 14 I=N,1,-1
        SUM=B(I)
        IF(I.LT.N)THEN
          DO 13 J=I+1,N
            SUM=SUM-A(I,J)*B(J)
13        CONTINUE
        ENDIF
        B(I)=SUM/A(I,I)
14    CONTINUE
      RETURN
      END
```

Literatura

[1] Kraut Bojan. *Strojniški priročnik*, pages 152–158. Strojniški vestnik, Ljubljana, 1981.

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

[5] Teague Robert. *Computing problems for FORTRAN solution.* Canfield Press, San Francisco [etc.], 1972.

[6] Šašić Mane. *Transport fluida cevima.* Mašinski fakultet, Beograd, 1982.

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 9. oktober 1995

Typeset in LaTeX