

# Laboratorijske vaje pri predmetu Računalniško Podprto Konstruiranje Hidravlični model talnega gretja

Leon Kos in Simon Kulovec, Laboratorij za CAD - LECAD

## Naloga

Izdelajte računalniški program za hidravlični preračun talnega gretja in program za prikaz rezultatov izračuna z uporabo grafičnega jezika OpenGL. V hidravličnem preračunu je potrebno določiti pretoke v zaprtem omrežju plastičnih (PEX) cevi. Tlačne izgube v ceveh določamo iz empiričnih enačb oz. diagramov. Poleg izgub v ceveh moramo upoštevati še izgube v ventilih. Obratovalno točko obtočne črpalke določite iz Q-h diagrama. Grafični prikaz v jeziku OpenGL mora omogočati 3D (prostorski) prikaz celotnega omrežja in pobarvati veji omrežja z največjim pretokom (rdeče) in najmanjšim pretokom (modro) in prikaz posameznih vej za kontrolo vhodnih podatkov preračuna. Za enostavnejše modeliranje talnega gretja se predvideva enostaven vnos podatkov sob z polaganjem cevi v krožni obliki. Poleg šolskega primera je potrebno prikazati delovanje na vsaj še dveh dodatnih zahtevnejših primerih, ki ju prikažete v poročilu na wiki strani.

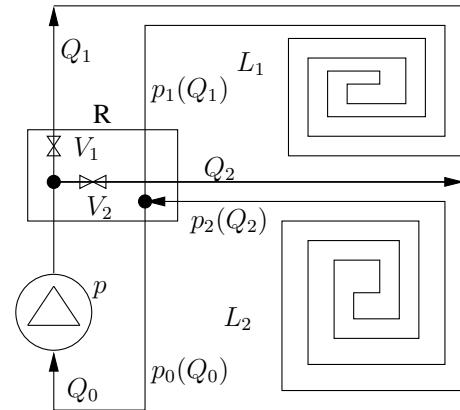
## 1 Uvod

Talno gretje se je v preteklosti obravnavalo kot cenovno neugodna tehnologija ogrevanja. Z novimi materiali so se razmere spremenile in tako lahko opazimo, da klasične tehnologije ogrevanja izgublajo svoje prednosti na vseh področjih. Prednosti talnega gretja so predvsem v (i) boljši izrabi prostora, (ii) nizkotemperaturnem režimu obratovanja, (iii) ugodnem temperaturnem profilu ogrevanja in (iv) manjšem dvigovanju prahu, kar ugodno vpliva na zdravje. Poraba zaradi nizko-temperaturnega obratovanja je lahko do 10% nižja.

Z novimi materiali kot so (i) plastične cevi iz navzkrižno vezanega polietilena (PEX), (ii) plastični razdelilci, ki nadomeščajo drage kovinske izvedbe in (iii) sistemske plošče za lažje polaganje cevi na enakomerne razdalje, talno gretje postaja najugodnejša rešitev ogrevanja za vse namene. Za pravilno delovanje ogrevanja je poleg osnovnega toplotnega preračuna potrebno zagotoviti tudi hidravlični preračun s katerim predvidimo pravilnost delovanja sistema kot celote. Zaradi uporabe nizko-cenovnih komponent, ki nimajo možnosti diagnostike pretokov je potrebno izračun in izris izdelati vnaprej. Na vajah RPK bomo uporabili splošno znanje iz strojništva in izdelali program s katerim bo mogoče z zadovoljivo natančnostjo in enostavnostjo uporabe izdelati hidravlični preračun.

## 2 Hidravlični preračun

Slika 1 prikazuje poenostavljen model talnega gretja. Črpalka  $P$  poganja toplo vodo do razdelilca  $R$  v katerem



Slika 1: Prikaz talne napeljave v dveh vejah z eno črpalko zaprtega sistema.

se pretok  $Q$  razdeli na dva pretoka  $Q_1$  in  $Q_2$ . Kolikšna sta ta dva pretoka je odvisno od upora cevi in nastavitve ventilov  $V_1$  in  $V_2$  v razdelilcu. Za vsako vejo je predviden v razdelilcu ventil s katerim nastavljamo pretok. Podatki za cevi, ventile in črpalke se podajajo v diagramih, ki kažejo osnovne značilnosti komponente.

Ne glede na to pred kakšno nalogo stojimo in s kakšnim tipom mreže imamo opravka, morata biti izpolnjena naslednja (Kirchoffova) zakona hidravlike:

1. Vsota vseh tokov, ki pritekajo v vozlišče in vseh tokov, ki odtekaajo iz vozlišča mora biti enaka nič.

$$\sum_i Q_i = 0 \quad (1)$$

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.

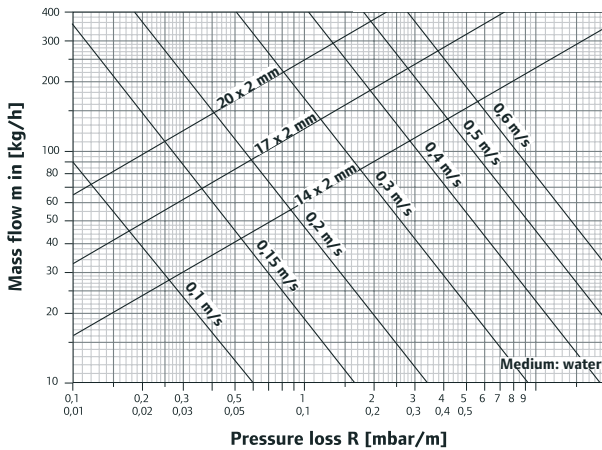
2. Vsota padcev tlaka po zaključeni zanki mora biti prav tako enaka nič. Statični tlak ne vpliva na zaključeno zanko, kot tudi ne višinske razlike, zato je zanka enačba odvisna le od izgub v ceveh.

$$\sum_i \Delta p_i(Q_i) = 0 \quad (2)$$

V sistemu kot je na sliki 1 so osnovne neznane količine pretoki v ceveh, kot tudi skupen pretok  $Q$ , ki je odvisen od črpalke  $P$ . Če primer poenostavimo tako, da vemo skupen pretok  $Q$  lahko napišemo, da je vsota tokov  $Q = Q_1 + Q_2$  in da je padec tlaka na obeh vejah enak  $\Delta p_1 = \Delta p_2$ .

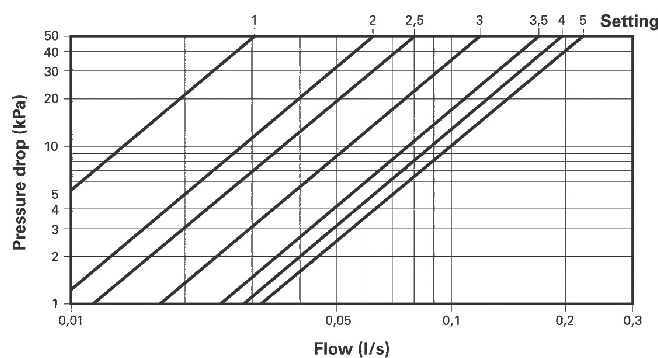
## 2.1 Tlačne izgube v ceveh in ventilih

Prvi diagram na sliki 2 v dvojno logaritmični skali prikazuje padec tlaka v odvisnosti od pretoka fluida in debeline cevi ter obratno. Diagrame poda proizvajalec cevi. Modeliranje enačbe odvisnosti tlaka od pretoka, je potrebno v programu izvesti tako, da funkcija izračuna padec tlaka za izbrano cev ter za sleherni pretok, ki je vhodni podatek. Iz diagrama je razvidno, da predvideva tri tipe



Slika 2: Diagram padca tlaka v odvisnosti od pretoka in debeline cevi.

cevi s premerom in debelino stene 20x2, 17x2 in 14x2. Ker enostavni razdelilci omogočajo hkrati le en tip cevi, predpostavimo da se zanke razlikujejo le v dolžini cevi. Zaporedno cevem pa so postavljeni ventili s katerimi nastavimo pretok. Iz diagrama na sliki 3, ki je v ravno tako v dvojno logaritmični skali se določi padec tlaka glede na položaj ventila (1=zaprt, ..., 8=odprt) in pretoka za vodo. Modeliranje diagrama se sestoji iz enačbe s katero napovemo padec tlaka za izbrano stopnjo zaprtja ventila in velikosti pretoka. Ker modeliramo predvsem s pretoki



Slika 3: Tlačne izgube v odvisnosti od zaprtja ventila zanke (1...6) in pretoka.

v vejah ( $Q_i$ ) lahko napišemo linearizirano enačbo padca tlaka tako za cev kot ventil

$$R = n Q^k \quad (3)$$

v logaritemski obliki kot

$$\log R = k \log Q + \log n \quad (4)$$

Za dve poljubni točki diagrama

$$\log R_1 = k \log Q_1 + \log n, \quad (5)$$

$$\log R_2 = k \log Q_2 + \log n \quad (6)$$

izrazimo neznani konstanti

$$k = \frac{\log R_1 - \log R_2}{\log Q_1 - \log Q_2}, \quad (7)$$

$$n = \exp\left(\frac{\log R_1 \log Q_2 - \log R_2 \log Q_1}{\log Q_2 - \log Q_1}\right). \quad (8)$$

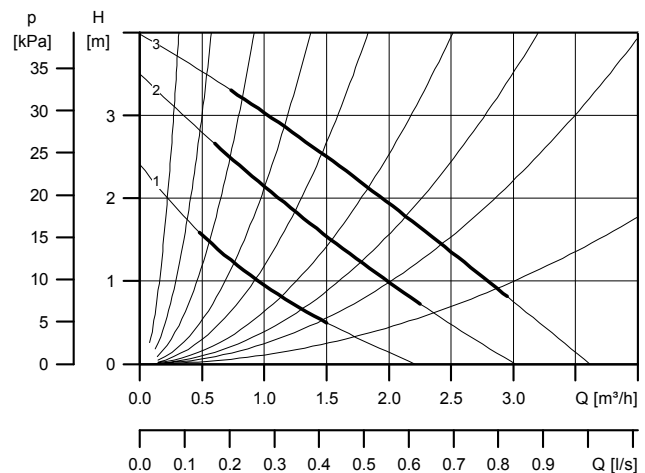
Skupen padec tlaka v veji  $i$  na cevi dolžine  $L_i$  in ventilom s konstantama  $k_{v,i}$  in  $n_{v,i}$  je tako ob enotnem pretoku  $Q_i$

$$\Delta p(Q_i) = L_i n_i Q_i^{k_i} + n_{v,i} Q_i^{k_{v,i}}. \quad (9)$$

Za naš primer lahko predpostavimo, da je padec tlaka v eni zanki enak dvigu tlaka na črpalki. Padci tlakov so tako na vseh vejah enaki in tako lahko napišemo za vsako  $i$ -to vejo enačbo (2), ki pa je zaradi nelinearnih členov (9) tudi sama nelinearna.

## 2.2 Tlačna karakteristika črpalke

Poleg nelinearnosti v ceveh in ventilih je nelinearen še tlak črpalke kot je prikazano v Q-H diagramu na sliki 4. Funkcijo tlaka črpalke  $p(Q)$  lahko podamo na različne



Slika 4: Karakteristika črpalke za tri različne položaje moči (1, 2, 3)

načine: (i) odsekoma zvezna interpolacija, (ii) aproksimacija empirične enačbe, (iii) polinomska interpolacija, (iv) polinomska aproksimacija, ...

Tako kot pri cevi in ventilu, bi želeli podati interpolacijsko funkcijo še za črpalko. Najenostavnejše je, da predpostavimo interpolacijski polinom. Če ne poznamo funkcije, vendar imamo možnost izbire interpolacijskih točk, so najbolj pametna izbira Chebysheve točke v intervalu  $[a, b]$

$$x_i = \frac{a+b}{2} + \frac{a-b}{2} \cos\left(\frac{2i+1}{2n+2} \pi\right), \quad i = 0, \dots, n. \quad (10)$$

Za primer izberimo 4 točke v diagramu 4 in izbranem položaju stikala 1 v katerem je razpon funkcije  $[0, 2.2]$ .

Čebiševe točke so tako  $x_i = \{0.08, 0.68, 1.52, 2.11\} [m^3/h]$ . Za te vrednosti razberemo še vrednosti tlakov  $f_i = \{21.2, 12.7, 4.59, 0.76\} [\text{kPa}]$ . Skozi štiri točke poteka en sam polinom, ki pa ga lahko zapišemo z različnimi baznimi funkcijami. Za interpolacijo je najbolj uporabna Newtonova shema deljenih diferenc interpolacijskega polinoma  $p_n(x)$ , ki jo v splošnem napišemo kot

$$\begin{aligned} p_n(x) &= \sum_{j=0}^n f[x_0, x_1, \dots, x_j] \prod_{i=0}^{j-1} (x - x_i) \\ &= f[x_0] + f[x_0, x_1](x - x_0) \\ &\quad + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\ &\quad + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned} \quad (11)$$

kjer je  $c_j = f[x_0, x_1, \dots, x_j]$  deljena diferenca, ki je definirana kot

$$\begin{aligned} f[x_i] &= y_i, \\ f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \quad (12) \\ f[x_i, \dots, x_{i+k}] &= \frac{f[x_i, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned}$$

Funkcija za štiri točke našega primera je kubični polinom

$$\begin{aligned} p_3(x) &= c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) \\ &\quad + c_3(x - x_0)(x - x_1)(x - x_2) \end{aligned} \quad (13)$$

za katerega izračunamo deljene diference  $c_i$  kot

$$\begin{aligned} c_0 = f[x_0] &= f(0.08) = 21.2 \\ f[x_1] &= f(0.68) = 12.7 \\ f[x_2] &= f(1.52) = 4.59 \\ f[x_3] &= f(2.11) = 0.76 \\ c_1 = f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{12.7 - 21.2}{0.68 - 0.08} \\ f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{4.59 - 12.7}{1.52 - 0.68} \\ f[x_2, x_3] &= \frac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{0.76 - 4.59}{2.11 - 1.52} \\ c_2 = f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ f[x_1, x_2, x_3] &= \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} \\ c_3 = f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}. \end{aligned}$$

Prednost Newtonove interpolacije je, da lahko dokaj enostavno izračunamo odvod enačbe (13)

$$\begin{aligned} p'_3(x) &= c_1 + c_2\{(x - x_0) + (x - x_1)\} \\ &\quad + c_3\{(x - x_1)(x - x_2) + (x - x_0)(x - x_2) \\ &\quad + (x - x_0)(x - x_1)\}, \end{aligned} \quad (14)$$

kar nam pride prav pri linearizaciji karakteristike črpalke v delovni točki in linearizaciji sistema enačb (1) in (2).

## 2.3 Linearizacija problema

Vsako funkcijo je mogoče razviti v potenčno vrsto okoli poljubne točke  $x = a$  s Taylorjevimi polinomom

$$f(x) = f(a) + f'(a)(x - a) + 1/2 f''(a)(x - a)^2 + O(x - a)^3. \quad (15)$$

Če raje pišemo z odmikom od točke  $h = x - a$  je enačba (15) krajša

$$f(h) = f(a) + f'(a)h + 1/2 f''(a)h^2 + O(h)^3, \quad (16)$$

s tem da pravimo točki  $a$  delovna točka,  $h$  pa odmik od nje. Enačbo padca tlaka (9) v eni sami veji (brez indeksa  $i$ ) bi torej lahko napisali kot

$$\begin{aligned} p(h) &= LnQ^k + n_v Q^{k_v} + Ln_k Q^{k-1} h + n_{v,k_v} Q^{k_v-1} h \\ &\quad + 1/2(Lnk(k-1)Q^{k-2} + n_{v,k_v}(k_v-1)Q^{k_v-2})h^2, \end{aligned} \quad (17)$$

kjer je  $Q$  začetni približek za pretok in  $h$  poljuben odmik pretoka. Člen drugega reda s spodnji vrstici enačbe (17) ni težko izračunati. Prav tako ne bi bilo težko izračunati drugi odvod tlačne karakteristike črpalke z dodatnim odvajanjem enačbe (14). Na točnost reševanja sistema enačb (1) in (2) členu drugega in višjega reda ne vplivajo in jih lahko zanemarimo, saj bomo reševali sistem iterativno s sprotnim popravljanjem delovnih točk  $Q_i$ . Vpliv členov bi se poznal le pri hitrosti konvergence. Za primer napišimo sistem dveh enačb z eno črpalčko, cevjo in ventilom v eni zanki. Imamo neznana dva pretoka  $Q_1 + h_1$  in  $Q_2 + h_2$  za katera lahko napišemo, da sta enaka z prvo Kirchoffovo enačbo (1)

$$Q_1 + h_1 - Q_2 - h_2 = 0, \quad (18)$$

kjer sta  $Q_1$  in  $Q_2$  poljubno izbrana delovna pretoka črpalke in cevi,  $h_1$  in  $h_2$  pa neznana popravka obeh pretokov, ki bosta zadovoljila enačbo (18). Vsota padcev tlakov (En. (2)) mora biti

$$-p_1(Q_1) - p'_1(Q_1)h_1 + p_2(Q_2) + p'_2(Q_2)h_2 = 0, \quad (19)$$

kjer je  $p_1(Q_1)$  tlak črpalke izračunan z enačbo (13) in  $p'_1(Q_1)$  odvod tlaka črpalke v točki  $Q_1$  izračunan z enačbo (14). Tlak  $p_2(Q_2)$  in odvod tlaka  $p'_2(Q_2)$  v cevi in ventilu za začetni pretok  $Q_2$  izračunamo po enačbi (17). Neznana odmika pretokov  $h_1$  in  $h_2$ , ki zadovoljita enačbi (18) in (19) napišemo v matrični obliki kot

$$\begin{bmatrix} 1 & -1 \\ -p'_1 & p'_2 \end{bmatrix} \begin{Bmatrix} h_1 \\ h_2 \end{Bmatrix} = \begin{Bmatrix} -Q_1 + Q_2 \\ p_1 - p_2 \end{Bmatrix}. \quad (20)$$

Primer bi lahko rešili tudi s substitucijo. Večje primere pa je potrebno reševati programsko. Večji problem je prikazan na sliki 1, kjer imamo tri neznane pretoke. Lahko postavimo dve zanki enačbi in eno tokovno. Izbor zank mora biti tak, da enačbe niso med sabo odvisne in jih je potrebno pisati tako, da vsaka podaja različen obhod. Prav tako velja za vozlišča saj je razvidno, da sta dva vozlišča enakovredna in da je vozliščna enačba ena sama. Za zanke je najenostavneje, da se pišejo tako, da gre vsaka ogrevalna veja v zanki skozi črpalčko. Pri tem se le našteje cevi in ventile v zanki s pripadajočimi pretoki.



### 3.2 Postavitev sistema enačb

Program prebere podatke iz tekstovne 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 .TXT. Po preračunu pa naj program izpiše na zaslon rezultate obdelave, kot so: vhodni podatki, pretoki, hitrosti, število iteracij ... Za kontrolo vhodnih podatkov in ponazoritev omrežja v prostoru bo skrbel grafični del programa. Grafični program bo ravno tako kot program za preračun, prebral vhodno datoteko in omogočil prikaz posameznih zank z barvanjem.

Pri določanju formata vhodne datoteke naj bo osnovno vodilo enostavnost uporabe, saj se bo ta datoteka vnašala za vsak primer ročno. Vhodna datoteka naj ima naslednji format:

```
underfloor: 1.0 1 2
17 10 1
17 55 2.5
17 70 2
3 0 1 2
2 0 1
2 0 2
```

V prvi vrstici podamo format datoteke, število vozliščnih enačb in število zračnih enačb. Format datoteke podaja prepoznavni niz (*magic string*) in verzijo datoteke s katero določimo, kakšno obliko lahko pričakujemo v nadaljevanju. S spremembo verzije datoteke na npr 2.0 lahko specifikacijo dopolnimo z grafičnimi podatki. Sledijo podatki o ceveh za katere je potrebno izračunati pretoke. Cev št. 0 je dovodna cev in vsebuje poleg premera in dolžine še podatek o položaju stikala za karakteristiko črpalke. Vse naslednje cevi imajo poleg premera in dolžine še položaj ventila s katerim reguliramo pretok. Če je ta številka 0, potem se ventila ne upošteva. Podatkom o ceveh sledijo vozliščne enačbe in zračne enačbe tako, da prva številka v vrsti najprej pove koliko cevi je v enačbi in nato so našteje cevi. Pri vozliščnih enačbah je za cev št. 0 predpostavljeno, da je dovodna, zato je smer pretoka negativna. Dogovor je, da se pišejo zračne enačbe tako, da gre zanka vedno skozi črpalko. Vrstica 3 0 1 2 tako predstavlja vozliščno enačbo v linearizirani obliki

$$-Q_0 - h_0 + Q_1 + h_1 + Q_2 + h_2 = 0. \quad (23)$$

Sledita dve zračni enačbi

$$-p_3 - p'_3 h_0 + p_0 + p'_0 h_0 + p_1 + p'_1 h_1 = 0, \quad (24)$$

$$-p_3 - p'_3 h_0 + p_0 + p'_0 h_0 + p_2 + p'_2 h_2 = 0, \quad (25)$$

kjer smo s  $p_3$  označili tlačno karakteristiko črpalke, katera je odvisna od pretoka  $p_3(Q_0 + h_0) = p_3(Q_0) + p'_3(Q_0)h_0$  in je aproksimirana z enačbama (13) in (14). Na ceveh (razen dovodnih) je v vsaki veji serijsko vezan ventil kot je to podano v enačbi (17) brez drugega reda. Postavitev sistema enačb zahteva, da se vsi členi, ki vsebujejo  $h_i$  pripišejo matriki  $\mathbf{m}$ , ostali pa se prestavijo v vektor  $\mathbf{b}$ . Za izračun členov matrike  $\mathbf{m}$ , je najbolje, da se iz vhodnih podatkov najprej izdelata smerna matrika, ki vsebuje ničle

in enice s predznaki in je za podan primer

$$e = \begin{bmatrix} -1 & 1 & 1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \quad (26)$$

Enice v vozliščnih enačbah pomenijo, da je na tem mestu potrebno izračunati pretoke. Podobno je v zračnih enačbah, le da se tam računajo padci tlaka. Ob prisotnosti enice se v matriki  $\mathbf{m}$  na teh mestih izračunajo odvodi tlaka in hkrati odštejejo tlaki v vektorju  $\mathbf{b}$ . Cev št. 0 je posebnost saj ima črpalko in nima ventila. Zaradi dogovora, da pišemo zračne enačbe vedno skozi črpalko, bodo členi v stolpcu 0 zračnih enačb vedno prisotni in bodo vsebovali negativni predznak za karakteristiko črpalke, ki tlak dviguje in padec tlaka na cevi št. 0. Kot je razvidno iz enačbe (20) se na levi strani sistema zračnih enačb pojavljajo odvodi, na desni pa negativni tlaki v delovni točki. V splošnem bi lahko zapisali lineariziran sistem enačb z  $n$  vozliščnimi in  $p = m - n$  zračnimi enačbami kot

$$\begin{bmatrix} v_{0,0} & v_{0,1} & \dots & v_{0,m-1} \\ \vdots & \vdots & & \vdots \\ v_{n-1,0} & v_{n-1,1} & \dots & v_{n-1,m-1} \\ z_0 & z_{n,1} & \dots & z_{n,m-1} \\ z_0 & z_{n+1,1} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ z_0 & z_{m-1,1} & \dots & z_{m-1,m-1} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_n \\ \vdots \\ h_{m-2} \\ h_{m-1} \end{bmatrix} = \begin{bmatrix} sv_0 \\ sv_1 \\ \vdots \\ sv_{n-1} \\ sz_n \\ \vdots \\ sz_{m-2} \\ sz_{m-1} \end{bmatrix}$$

Elementi vozliščnih enačb matrike so enaki smerni matriki

$$v_{i,j} = e_{i,j}, \quad i = [0, n-1], \quad j = [0, m-1] \quad (27)$$

Vektor  $\mathbf{b}$  vozliščnih enačb je enak vsoti

$$sv_i = - \sum_{j=0}^{m-1} e_{i,j} Q_j \quad i = [0, n-1]. \quad (28)$$

V zračnih enačbah je posebnost le element, ki hkrati vsebuje črpalko in cev št. 0. Element

$$z_0 = p'_0(Q_0) - p'_c(Q_0) \quad (29)$$

je tako enak za vse zračne enačbe in vsebuje odvod tlaka črpalke  $p_c$  v obratovalni točki  $Q_0$ . Ostali elementi matrike zračnih enačb se izračunavajo kot

$$z_{n+i,j} = e_{n+i,j} (L_j n_j k_j Q_j^{k_j-1} + n_{v,j} k_{v,j} Q_j^{k_{v,j}-1}), \quad (30)$$

$$i = [0, p-1], \quad j = [1, m-1].$$

in so po stolpcih enaki in odvisni le od smerne matrike. Celoten prispevek v vektorju  $\mathbf{b}$  zračne enačbe je

$$sz_{n+i,j} = -p_0(Q_0) + p_c(Q_0) - \sum_{j=1}^{m-1} e_{n+i,j} (L_j n_j Q_j^{k_j} + n_{v,j} Q_j^{k_{v,j}}). \quad (31)$$

Za prikazani primer s slike 1 je začetna oblika sistema z začetnimi pretoki v  $Q_i = 100$  l/h in tlaku v kPa v naslednji obliki

$$\begin{bmatrix} -1 & 1 & 1 \\ 0.028 & 0.19 & 0 \\ 0.028 & 0 & 0.28 \end{bmatrix} \begin{Bmatrix} h_0 \\ h_1 \\ h_2 \end{Bmatrix} = \begin{Bmatrix} -100 \\ 9.6 \\ 5.7 \end{Bmatrix},$$

ki že po nekaj iteracijah privede do rešitve  $Q_0 = 231.3$  l/h,  $Q_1 = 125.8$  l/h in  $Q_2 = 105.5$  l/h. Iz pretočkov lahko brez težav izrazimo tudi padce tlaka na ceveh, ventilih in obratovalno točko črpalke.

## 4 Grafični prikaz

Za grafičen prikaz rezultatov v programu uporabite grafični jezik OpenGL. Za uporabniški vmesnik uporabite knjižnico GLUT. Z nadgradnjo vhodne datoteke (verzija 2.0) lahko dopolnimo vhodne podatke, ki popisujejo fizikalni sistem z grafičnimi podatki, ki predstavljajo cevi. Primer take razširitve predstavlja naslednja datoteka:

```
underfloor: 2.0 1 2
17 10 1
17 55 2.5
17 70 2
3 0 1 2
2 0 1
2 0 2
1 22
0 0 0
0 0 -2.7
...
```

S spremembo verzije iz 1.0 na 2.0 smo programu dopovedali, da sledijo še grafični podatki, ki podrobneje popisujejo cevi v takem zaporedju, kot so bile vnesene. Ker pa so cevi lahko sestavljene tudi iz dveh delov najprej s številko 1 povemo, da je cev 0 sestavljena iz ene lomljenke in da sledi 22 točk v formatu  $x, y, z$ . Točke lomljenk bomo vpisovali ročno s pomočjo programov, ki si jih bomo sami izdelali, da si bomo olajšali vnos v urejevalnik.

Osnovni program, ki demonstrira uporabo jezika OpenGL in knjižnice uporabniškega vmesnika GLUT in omogoča vnos z miško:

```
#include <stdio.h>
#include <stdlib.h>
#include <GL/glu.h>
#include <GL/glut.h>

#define MAXN 100
GLint n;
GLfloat *vertex;

void redraw()
{
    int i;
    glClearColor(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINE_STRIP);
    for (i = 0; i < n; i++)
        glVertex2fv(&vertex[2*i]);
    glEnd();
    glutSwapBuffers();
}

void mouse(int button, int state, int x, int y)
{
    GLint viewport[4];
    GLdouble mvmatrix[16], projmatrix[16];
    GLdouble wx, wy, wz; /*x, y, z coords*/
    GLdouble px, py, pz; /*picked window coords*/
    int status;
```

```
if (button == GLUT_LEFT_BUTTON )
    if (state == GLUT_DOWN) {
        glGetIntegerv (GL_VIEWPORT, viewport);
        glGetDoublev (GL_MODELVIEW_MATRIX, mvmatrix);
        glGetDoublev (GL_PROJECTION_MATRIX, projmatrix);
        /* note viewport(4) is height in pixels */
        px = x;
        py = viewport[3] - y - 1;
        pz = 0.0;
        fprintf (stderr,
                "Coordinates at cursor are %f, %f\n",
                px, py);
        status = gluUnProject (px, py, pz, mvmatrix,
                               projmatrix, viewport, &wx, &wy, &wz);
        fprintf(stderr,
                "World coords at z=0.0 are %f, %f, %f\n",
                wx, wy, wz);
        if (n < MAXN) {
            vertex[n*2] = wx; vertex[n*2+1] = wy;
            n ++;
        } else
            fprintf(stderr,
                    "Dosezeno maksimalno stevilo tock!\n");
        glutPostRedisplay();
    }
}

int main(int argc, char *argv[])
{
    vertex = (GLfloat *)
        malloc(2 * MAXN * sizeof (GLfloat));
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (500, 500);
    glutInitWindowPosition (100, 100);
    glutCreateWindow("Click in window");
    glutDisplayFunc(redraw);
    glutMouseFunc(mouse);
    glutMainLoop();
    return 0;
}
```

Na vajah so prikazani tudi primeri, kako se algoritmično zgradi lomljenka, ki popisuje zanko. Za označevanje in izpis rezultatov v grafični obliki v prostoru je najenostavneje uporabiti rastrske (bitmap) fonte. Ker pa GLUT omogoča izpis le enega znaka v prostoru, se predlaga naslednji podprogram, ki omogoča izpis niza znakov, podobno kot funkcija `printf(format, ...)`, le da podamo še začetno točko v prostoru.

```
#include <stdarg.h>
#include <GL/glut.h>

void print3D(GLfloat x, GLfloat y, GLfloat z,
             const char *format, ...)
{
    glRasterPos3f(x, y, z);
    char buf[80];
    char* ch = buf;
    va_list args;
    va_start(args, format);
    vsnprintf(buf, 80, format, args);
    va_end(args);
    while (*ch)
        glutBitmapCharacter(GLUT_BITMAP_HELVETICA_12, *ch++);
}
```

Uvod v OpenGL s primeri se nahaja na spletni strani <http://lecad.si:8000/vaje/wiki/opengl-intro>

## 5 Zahteve projekta

Na laboratorijskih vajah se po uvodnih vajah in domačih nalogah se predvideva samostojno delo študentov na projektni nalogi. Študent s samostojnim delom prikaže absorbirano znanje na nekoliko širšem problemu. *Sodelovanje pri delu med študenti ni dovoljeno*, saj je bistveni del ocene pridobljen prav s takim delom. Težave se lahko diskutirajo le z asistentom na vajah ali preko listkov.

Ker je projekt sestavljen iz dveh delov lahko programa prenašata podatke preko datoteke. Lahko pa se izdelava celoviti program. Poleg navedenih zahtev se lahko vključi še dodatne pomožne programe za lažje tvorjenje vhodne datoteke za grafični del, kar pa je potrebno podrobneje opisati v poročilu.

**Numerični del.** V numeričnem delu naloge je potrebno iz vstopnih podatkov, ki jih preberemo iz vhodne tekstovne (.txt) datoteke izračunati pretoke, padce tlakov na ceveh in ventilih, tlake, obratovalno točko črpalke in jih izpisati na zaslon za poljubno napeljavo. Vhodne podatke je potrebno nadgraditi tako, da se dolžine cevi izračunajo neposredno iz 3D podatkov lomljenk. Poleg šolskega primera je potrebno pripraviti vsaj še dva obsežnejša primera (realna) preračuna in ustrezno nastavitve ventilov. Minimalno št. razdelilcev je 2. Minimalno število zank 10.

**Grafični del.** Za grafični del naloge je potrebno iz vhodnih in izračunanih podatkov izdelati 3D model poljubne talne napeljave, na katerem so prikazani vsi podatki (pretoki, tlaki, ...). Izdelati je potrebno vsaj dve unikatni vhodni datoteki, ki tudi grafično prikazujeta sistem talnega gretja. Eden mora biti v vsaj dveh nadstropjih. Minimalno število razdelilcev je 2. Minimalno št. zank je 4. Grafični vmesnik mora omogočati vrtenje okoli vseh treh osi in spremembo velikosti pogleda s tipkami ali miško. Vozlišča naj se obarvajo v drugačni barvi od cevi. Na sredino vsake cevi je potrebno napisati številko cevi in zahtevane rezultate preračuna. Ob zagonu programa naj program izpiše kratko navodilo, katere tipke oz. miš se lahko uporabi pri interakciji s programom. Tipka h pa naj to kadarkoli tudi izpiše.

**Poročilo.** Zaradi lažjega ocenjevanja in doseganja enotnih standardov, se predvideva enoten način izdelave poročila o nalogi. Poročilo o vajah se oddaja na WIKI strani vašega projekta. Prvi del poročila naj opisuje numerični program, drugi pa naj predstavi 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 uporabljati 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.

Celotno poročilo naj ima klasično obliko (povzetek, uvod, kratek opis teoretičnih osnov brez ponavljanja besedila iz predloge, rezultati, možne izboljšave, zaključek, literatura, priloge). Izberite si primer in na njem pokažite reševanje problema.

Diskutirajte izboljšave programa, ki ste jih dogradili in niso del minimalnih zahtev. Glede na poročilo se bo tudi preverjalo te izboljšave. Nestandardne formate vhodne datoteke podrobneje komentirajte. Diskutirajte tudi omejitve, ki so posledica vgrajenih algoritmov (velikost matrik, stabilnost, ...). Torej: poročilo mora biti napisano tako, da je možno iz njega rekonstruirati program.

Spletni vmesnik TRAC omogoča strukturirano pisanje dokumentov, ki spremljajo projekt. Vsak študent je tako na strani WikiStart svojega projekta, dolžan predstaviti svoje delo v obliki poročila v katerem predstavi projekt, komentira svoje delo in delovanje programa. Strukturirano obliko poročila je možno doseči z logičnimi ukazi, ki so podrobneje predstavljeni na strani *WikiFormatting*. Predvideno je samostojno učenje opisnega jezika za oblikovanje strani, za kar je predviden prostor v peskovniku (*SandBox*). Učenje na strani SandBox se ne shranjuje in je predviden ravno zaradi tega, da se z njim ne smeti časovni potek dela.

Poročilo je ravno tako, kot kodo možno slediti v Timeline. Ravno tako mora biti iz Timeline razviden potek izdelave poročila. V poročilu naštejete tudi revizije programa iz Timeline in kaj ste dosegli z določeno spremembo! Pomemben del ocene je tudi sprotnost dela in zmožnost razdelitve problema na podprobleme.

## 6 Delo na računalniku

Delo (datoteke) je tako potrebno shranjevati na strežnik. Za hranjenje in sledenje spremembam je na strežniku nameščen sistem *subversion*. Prikaz, primerjanje in komunikacija pri izvedbi vaj pa poteka s sistemom *TRAC* na namenskem strežniku <http://www.lecad.fs.uni-lj.si:8000/>. Vsa potrebna navodila in dodatna so dostopna na projektu <http://www.lecad.fs.uni-lj.si:8000/vaje>.

**SVN Dostop do primerov v projektu vaje in rpk**  
Primere lahko poleg brskanja na WWW strani poberete tudi z ukazom

```
svn cp --username vaje \
    svn://lecad.si/rpk/vaje/lupack.c .
```

Podajanje uporabniškega imena vaje v navideznem računalniku ni potrebno, saj je to ime že privzeto. Geslo za dostop je ravno tako vaje.

## 7 Terminski načrt

Po sedmih uvodnih vajah na katerih so prikazani potrebni postopki za uspešno izdelavo naloge, se predvideva samostojno delo študentov za katerega je do konca semestra predvideno še sedem terminov na katerih bo prisoten asistent s katerim lahko rešujete težave pri izdelavi aplikacije. Na vajah bodo prikazane osnove programskega je-

zika C s primeri, katere bo možno razširiti ali uporabiti v končnem programu.

Po uvodnih vajah je obisk v laboratoriju LECAD je še vedno obvezen, vendar je pričakovati različen napredek, ki je odvisen od posameznikove usposobljenosti reševanja večjih problemov. Obisk laboratorija je možen tudi izven predvidenega urnika, če so računalniki učilnice prosti. Možna je tudi namestitev navideznega računalnika z vsemi potrebnimi orodji in s tem delo od doma.

Vaje se zaključijo 13. januarja in takrat je možno oddati končno verzijo tako, da zaprosite asistenta za pregled in oceno vašega izdelka, ki se je oddajal sproti, kar bo tudi razvidno iz poteka oddajanj (*TracTimeline*). V dogovoru s študenti bo določen tudi rok za oddajo, ki mora biti pred pričetkom letnega semestra.

## 8 Ocenjevanje

Programska koda naj vsebuje komentarje, ki bodo opisovali pomen posameznih odsekov programa. Uporaba MakeFile-a je obvezna, saj je v osnovi program napisan na virtualnem računalniku, kjer je operacijski sistem Linux. Pri oddaji poročila in programa mora biti tudi testni primer, s katerim se dokaže pravilnost delovanja programa.

Ocena naloge bo sestavljena iz naslednjih kriterijev:

- 10% Sprotnost dela in sledenje razvoju s sistemom TRAC iz katerega mora biti jasno razviden postopek, s katerim ste prišli do končne verzije izdelka. Iz vsake opombe ob shranitvi na strežnik mora biti jasno predstavljena vsebina sprememb! To velja tako za izvorno kodo, kot za poročilo.
- 10% Jasnost sporočil pri shranjevanju dela in opis le teh v poročilu.
- 5% Prisotnost in izvedba domačih nalog s katerimi si pridobimo delovni cikel.
- 35% Delovanje programa za preračun, izris hitrosti v točkah notranjosti, jasnost kode. Uporaba programa `make` in datoteke `Makefile` je obvezna.
- 20% Oblika in vsebina poročila. Poročilo v elektronski obliki se vnaša s sistemom Trac na strani WIKI.
- 20% Neobvezni del funkcionalnosti programa, ki obsega:
  - Izračun Newtonove interpolacije za več točk z dvojno `for` zanko (5%).
  - Integracija v zaključeno aplikacijo. S tako aplikacijo je možno iterativno ponavljati vnos, preračun in prikaz brez izhoda iz aplikacije (5%).
  - Upoštevanje zaokrožitev v vejah (2%). Risanje zaokrožitev (3%).
  - Povečana stopnja interaktivnosti z uporabo miške in menujev GLUT (5%).
  - Riši meje sob (5%).
  - Interaktivno risanje sobe v mreži točk (10%)
  - Program za mreženje različnih tipov sob, ki niso samo pravokotne (7%).

- Avtomatsko risanje zanke z željvo grafiko (10%)
- Uporabi konfiguracijsko datoteko v kateri se podajo karakteristike cevi in ventilov in se tako omogoči poljubne proizvajalce cevi, razdelilicev in črpalk.

### Rang ocen:

50% - 60%	=	6
61% - 70%	=	7
71% - 80%	=	8
81% - 90%	=	9
91% - 100%	=	10

Predstavljeni kriteriji niso dokončni in se bodo lahko dopolnili ali popravili do konca koledarskega leta, če bodo v tem času ugotovljena nova dejstva. Timeline in Wiki-Start strani posameznih projektov so po uvodnih vajah vidni le z geslom, kar onemogoča neavtoriziran dostop do vašega izdelka. Sodelovanje študentov na način, kjer bi se programska koda prepisovala na kakršen koli način, ni dovoljeno. Če se problematika diskutira, potem naj se to izvaja tako, da se v kodo ne gleda oziroma se preverja kodo na abstraktnem primeru. Izdelki vseh študentov (poročila in programi) so ob ocenjevanju primerjani med seboj s sistemom MOSS (*Measure Of Software Similarity*), s katerim je možno enostavno analizirati kodo in detektirati morebitni plagiarizem in povod za negativno oceno.

**Kako zaprositi za oceno?** Poleg osebnega kontakta in e-pošte je za vse najbolj priročno, če zaprosite kar v Tracu, tako da odprete nov Ticket. Tam napišite kar želite in asistenta bosta preko e-pošte obveščena, da je odprt nov listek. Če želite tudi Vi spremljati dogodke na listku je najbolje, da pod Settings napišete svoje e-poštni naslov in boste tako obveščeni, ko se bo pojavil odgovor. Vsi projekti bodo imeli spremljanje ocenjevanja preko listka. Možne so seveda tudi pripombe.

Datotek ni potrebno prilagati v "Ticket". Dovolj je, da zaprosite za oceno. Pogledala bova SVN verzijo, ker je lažje dobiti izdelek naenkrat, kot pa vsako datoteko pobirati v začasni imenik.

## Literatura

- [1] Kraut Bojan. *Strojniški priročnik*, pages 152–158. Strojniški vestnik, Ljubljana, 1981.
- [2] Jože Duhovnik, Milan Kljajin, and Milan Opalič. *Inženirska grafika*. Univerza v Ljubljani, Fakulteta za strojništvo, 2009.
- [3] Šašič Mane. *Transport fluida cevima*. Mašinski fakultet, Beograd, 1982.

Ljubljana 22. december 2009