

Proračunska aerodinamika

Nedelja 1

A. Simonović & J. Svorcan

Mašinski fakultet, Katedra za vazduhoplovstvo

2020/2021.



Sadržaj

Opšte informacije

Rad na Linux-u

Uvod u FORTRAN

Uvod u MATLAB/QtOctave

Uvod u Python

Primer 1

Primer 2

Primer 3

Primer 4

Primer 5

Zadaci za vežbu



Termin i mesto održavanje nastave

Utorkom od 14h u učionici 452 (SimLab) ...



Osnovni cilj kursa

- Upoznavanje studenata vazduhoplovstva sa osnovama vazduhoplovnih proračuna prevashodno primenom metode konačnih razlika (MKR).
- Predstavlja kombinaciju nekoliko disciplina:
 - fizike (mehanike, mehanike fluida),
 - matematičkog proračuna (numerike)
 - i programiranja (najmladje grane stare nekoliko decenija),što ga čini složenim ali i veoma moćnim inženjerskim alatom.
- Fizički zakoni koji opisuju promene veličina, npr. strujanje fluida, prenos toplote ili složena naponska stanja, po nepravilnim oblastima/domenima izražavaju se parcijalnim diferencijalnim jednačinama (PDJ) i potrebno je orudje za njihovo uspešno rešavanje.



Osnovni plan kursa

- Uvod u modeliranje strujnih problema.
- Pregled osnovnih numeričkih metoda za rešavanje PDJ.
- Pregled osnovnih šema za aproksimaciju/diskretizaciju problema.
- Prepoznavanje tipa strujnog problema i zadavanje početnih i graničnih uslova.
- Implementacija prostijih programa za proračun.
- Uočavanje strukture simulacije (koja se sastoji iz: preprocesiranja, proračuna i pregleda i analize rezultata).
- Korišćenje kompleksnijih dopunskih alata.
- Samostalno rešavanje domaćih zadataka/projekata.



Dodatna literatura



Z. Petrović & S. Stupar.

Projektovanje računarom: metod konačnih razlika.
MF, 1996



J.D. Anderson.

Computational Fluid Dynamics: The Basics with Applications.
McGraw-Hill, 1995



J.H. Ferziger & M. Perić.

Computational Methods for Fluid Dynamics.
Springer, 2002



Uslovi, zahtevi

| Studentske obaveze | Mogući broj poena |
|----------------------------------|-------------------|
| Prisustvo | (+5) |
| Kolokvijum | 15 |
| Samostalni radovi, domaći zadaci | 15 + 20 + 20 |
| Ispit | 30 |
| Uslov za izlazak na ispit | 30 |



OS Linux

- Rad na operativnom sistemu UNIX započeo je 60-ih godina (i dalje traje).
- Odlikuju ga “otvorenost i dostupnost”. Postoje različite varijante (mi radimo na GNU/Linux).
- Sada ima i GUI sličan Windows-u (što većini korisnika olakšava rad) ...
- Sastoji se iz tri dela (kernel - osnovni nivo, shell - vrsta interfejsa i programi).
- UNIX sadrži samo datoteke i procese.

Korisni linkovi:

- <http://www.ee.surrey.ac.uk/Teaching/Unix/>
- <http://www.computerhope.com/unix.htm>
- <http://mally.stanford.edu/~sr/computing/basic-unix.html>



Organizacija sistema

- /boot – kernel,
- /etc – konfiguracijski fajlovi,
- /bin, /usr/bin – programi potrebni za OS,
- /sbin, /usr/sbin – fajlovi i programi za administraciju,
- /usr/doc – dokumentacija o instaliranom softveru,
- /usr/src – source fajlovi,
- /usr/include – header fajlovi (struktura, kompajliranje),
- /lib – biblioteke za programe koji se dinamički linkuju,
- /home – folder za korisničke fajlove,
- /root – home folder root user-a,
- /var – promenljivi podaci,
- /var/log – sistemski log fajlovi,
- /dev – uređaji.



Rad na Linux-u

- Struktura direktorijuma je hijerarhijska:

root(/) → home → student...

Primer putanje: *"/home/student..."*

- Otvaranje Linux terminala: kliknuti na ikonicu **Terminal** u toolbar-u **Applications...**
- Da bi se videlo šta se nalazi u folderu, kucati **ls** (list).
- **% ls -a** – prikaz i skrivenih direktorijuma (**-a** je ovde dodatna opcija komande **ls**).
- **% ls -l** – detaljan listing.
- Prilikom zadavanja imena fajlovima i folderima treba izbegavati specijalne znakove **/ * & %**, kao i blanko znak (bolje koristiti **_**).



Rad na direktorijumima

- Pravljenje novog foldera **mkdir** (make directory):
% **mkdir student**
% **ls**
- Promena radnog foldera **cd** (change directory):
% **cd student**
% **ls** (trebalo bi da je prazan)
% **cd ..** – vraća korisnika u prethodni folder (ovde **home**)
% **cd** – vraća korisnika u **home** folder
- Ispis putanje foldera **pwd** (print working directory):
% **pwd**



Rad na direktorijumima i datotekama

- Kopiranje datoteka **cp** (copy):
% cp /student05/.../pr1.txt . – kopira datoteku **pr1.txt** u trenutni folder (označen **.**)
- Pomeranje fajlova **mv** (move):
% mv file1 file2 – odgovara naredbama **cut** i **rename**
- Uklanjanje datoteka i direktorijuma **rm** (remove), **rmdir** (remove directory):
% cp pr1.txt pomocni.txt
% ls
% rm pomocni.txt
% ls



Rad na direktorijumima i datotekama

- Brisanje terminala **clear** (clear screen): **% clear**
- Ispis/prikaz fajla **cat** (concatenate): **% cat pr1.txt**
- Ispis sadržine fajla **less**: **% less pr1.txt**
- Ispis prvih deset linija fajla **head**: **% head pr1.txt**
- Ispis poslednjih deset linija fajla **tail**: **% tail pr1.txt**
- Broj reči **wc** (word count):
 - % wc -w pr1.txt** – broj reči
 - % wc -l pr1.txt** – broj linija
- Gornja strelica izlistava prethodno kucane naredbe.



Rad na direktorijumima i datotekama

- Specifikacija fajla **chmod** (changing a file mode):
u – user, **g** – group, **o** – other, **a** – all,
r – read, **w** – write, **x** – execute,
+ – dati dozvolu, **-** – oduzeti dozvolu,
% chmod go-rwx pr1.txt
% chmod a+rw pr1.txt
- Informacije o procesu **ps** (process): **% ps**
(foreground **% fg**, background **&**, suspended **% bg**)
% sleep 10 – čeka 10s
% sleep 10 & – čeka 10s u pozadini



Rad na direktorijumima i datotekama

- `% ^z` ([Ctrl] + z) – prekid izvršavanja programa
- Lista trenutnih procesa **jobs**:
`% jobs`
`% fg %1` – pokreće proces pod rednim brojem 1
- Potpuni prekid izvršavanja:
`% ^c` ([Ctrl] + c) – za *foreground* procese,
`% kill %i` – za *suspended* i *background* procese, **i** je redni broj



Još neke korisne naredbe ...

- Tekstualni editor, npr. **emacs** ili **gedit**:
% **emacs pr1.txt** ili
% **gedit pr1.txt**
- Zipovanje **gzip**:
% **ls -l**
% **gzip pr1.txt** – zipuje i pravi **pr1.txt.gz**
% **ls -l**
% **gunzip pr1.txt.gz** – proširuje fajl
(% **tar -xvf ime.tar** – ekstrahuje sadržinu *tar* fajla)
- Istorija kucanih komandi **history**:
% **history**



Uvod u FORTRAN

FORTRAN (Formula Translating System)

Proceduralni programski jezik razvijen 50-ih godina naročito pogodan za numeričke proračune i naučno programiranje.

- Bilo koji FORTRAN-ski program sastoji se iz niza linija koje mogu sadržati 72 znaka (karakter i od 73-80 služe za numeraciju).
- Prvih 6 mesta linije (prvih 6 kolona) koristi se za oznaku naredbe, signalizaciju komentara (**C** na prvom mestu) ili nastavka linije (***** na 6. mestu).
- Program počinje naredbom **PROGRAM**, a završava se naredbom **END**. Prekid izvršavanja koda – **STOP**.
- Ulaz: **READ(*,*)** ili **READ ***
- Izlaz: **WRITE(*,*)**, **WRITE *** ili **PRINT(*,*)**, **PRINT ***



Kompajliranje

- Svi programi napisani u programskom jeziku višeg nivoa moraju biti prevedeni na osnovni/bazični nivo koji računar razume.
- Za programe napisane u FORTRAN-u, koristi se:
% gfortran imeFajla.for -o imelzvrsnogFajla.out
- Pokretanje izvršnog fajla:
% ./imelzvrsnogFajla.out



Tipovi promenljivih

- Celobrojne promenljive – **INTEGER** i, j
- Realne promenljive – **REAL** a, b
- Slovnne promenljive – **CHARACTER** c
- Kompleksne promenljive – **COMPLEX** m, n
- Logične promenljive – **LOGICAL** p, q
- Višedimenzione promenljive – **INTEGER** aa(10), bb(5,10)



Naredbe, operatori

Naredba dodele =

Aritmetički operatori + - * / ** ()

Osnovne funkcije **sin(x), cos(x), tan(x), atan(x)**

abs(x), sqrt(x)

exp(x), log(x)

Relacije **== .lt. .le. .gt. .ge.**

Uslovna naredba **IF if (a == 5) then**

b = 2

else

b = 3

end if



Naredbe, operatori

Ciklična naredba **DO** **do i = 0,20,2**
 write(*,*) i
 end do

Ciklična naredba **DO WHILE** **x = 20**
 do while (x.gt.0)
 x = x-1
 write(*,*) x
 end do



Rad sa datotekama

- Da bi podaci trajali i nakon izvršenja programa, korisno je čitati ih iz ili upisivati u datoteku. Pre korišćenja, potrebno je otvoriti datoteku:

OPEN(unit=1, file='niz.dat', status= 'unknown')

- Može i **OPEN(1, file='niz.dat')**.
- Status može biti **'unknown', 'new'** ili **'old'**.
- Čitanje iz datoteke: **READ(1,*) ...**
- Upis u datoteku: **WRITE(1,*) ...**
- Na kraju je potrebno zatvoriti datoteku:
CLOSE(unit=1) ili **CLOSE(1)**



Za dalji rad u FORTRAN-u

Korisni linkovi:

- <http://www.cs.mtu.edu/~shene/COURSES/cs201/NOTES/fortran.html>
- <http://jupiter.ethz.ch/~pjt/FORTRAN/FortranClass.html>
- <http://www.egr.unlv.edu/~ed/fortran.html>



MATLAB/QtOctave

- Veoma zgodno okruženje za numeričke proračune.
- Lako manipulisanje matricama.
- Brzo skiciranje.
- Mogućnost pisanja sopstvenih funkcija, skriptova, grafičkog interfejsa ...
- Postoje brojni dodatni moduli.
- Veoma korišćen na univerzitetima i u industriji.
- QtOctave je open-source varijanta.



Tipovi promenljivih

- Celobrojne promenljive – **int8** ($-2^7 \div 2^7 - 1$), **int16**, **int32**, **int64**, **uint8**, ...
- Realne promenljive – **single**, **double**
- Slozne promenljive – **char**, **string**
- Kompleksne promenljive – **complex**, $z = \text{complex}(a,b)$
- Logične promenljive – **logical**, $y = 1 < 2$
- Višedimenzione promenljive
- Strukture
- Objekti
- Datumske promenljive
- ...



Naredbe, operatori

Naredba dodele =

Aritmetički operatori + - * / ^ ()

Osnovne funkcije **sin(x), cos(x), tan(x), atan(x)**

abs(x), sqrt(x)

exp(x), log(x)

Relacije == < <= > >=

Uslovna naredba **if** **if a == 5**

b = 2;

else

b = 3;

end



Uvod u Python

Python

Programski jezik visokog nivoa, podržava imperativni (proceduralni), objektno-orijentisani i funkcionalni stil programiranja. Razvijen početkom 90-ih godina.

- Popularan zbog: preglednosti koda, raznovrsnosti pristupa, jednostavnosti, brzog učenja.
- Veoma razvijene biblioteke različitih modula ...
- Više na <https://www.python.org>
- Potreban softver:
 - Python (verzija 2 ili 3)
 - NumPy (Numerical Python) – proračunski/numerički modul,
 - Matplotlib – za skiciranje grafika,
 - (SymPy – simbolički račun),
 - (SciPy – napredno naučno programiranje).



Ispis poruke na ekran

FORTTRAN:

```
% emacs pr1.for
```

```
program pr1
```

```
print *, "Moj prvi program"
```

```
end
```

```
ili
```

```
% emacs pr1b.for
```

```
program pr1b
```

```
write(*,*) "Moj prvi program"
```

```
end
```

```
% gfortran pr1.for -o pr1.out
```

```
% ./pr1.out
```

MATLAB/QtOctave:

(kucati u dokumentu ili
komandnom prozoru)

```
pr1.m
```

```
disp('Moj prvi program')
```



Suma dva broja

FORTRAN:

```
% emacs pr2.for
```

```
program pr2
```

```
real a, b, c
```

```
write(*,*) "Unesite dva broja"
```

```
read(*,*) a,b
```

```
c = a+b
```

```
write(*,*) "Suma je: ", c
```

```
end
```

```
% gfortran pr2.for -o pr2.out
```

```
% ./pr2.out
```

MATLAB/QtOctave:

```
pr2.m
```

```
clear all, clc
```

```
prompt = 'Unesite prvi broj';
```

```
a = input(prompt);
```

```
prompt = 'Unesite drugi broj';
```

```
b = input(prompt);
```

```
c = a + b;
```

```
disp(['Suma je ', num2str(c)])
```



Maksimalni član niza (FORTRAN)

pr3.for

```

program pr3
real a(10), mx
write(*,*) "Unesite članove niza"
read(*,*) (a(i), i=1,10)
mx = a(1)
do i=2,10
    if (mx .lt. a(i)) then
        mx = a(i)
    end if
end do
write(*,*) "Najveci član niza je: ", mx
end
    
```



Maksimalni član niza (MATLAB/QtOctave)

pr3.m

```
niz = 1:2:10;          % definicija niza  
[mx,j] = max(niz);    % najveći član i njegov redni broj  
disp(mx)
```



Ispis u datoteku (FORTRAN)

pr4.for

```

program pr4
real a(10)
write(*,*) "Unesite clanove niza"
read(*,*) (a(i), i=1,10)
open(1,file="promenjeniNiz.txt")
do i=1,10
    a(i) = a(i)*2+3
end do
write(1,*) (a(i), i=10,1,-1)
close(unit=1)
end
    
```



Ispis u datoteku (MATLAB/QtOctave)

pr4.m

```
clear all, clc
x = 0:.1:1;
A = [x; 2*x+3];
% otvaranje datoteke
fileID = fopen('promenjeni.txt', 'w');
% ispis prvog reda
fprintf(fileID, '%6s %12s\n', 'x', '2*x+3');
% ispis oba niza
fprintf(fileID, '%6.2f %12.8f\n', A);
% zatvaranje
fclose(fileID);
```

Odgovarajuća naredba za čitanje: **fscanf**. Probati i **load/save**.



Ispis korenova u datoteku (FORTRAN)

pr5.for

```
program pr5
real a(10), b(10)
open(1, file='niz.txt')
read(1,*) (a(i), i=1,10)
close(1)
do i=1,10
    b(i) = sqrt(a(i))
end do
open(2, file='nizB.txt')
write(2,*) (b(i), i=1,10)
close(2)
end
```



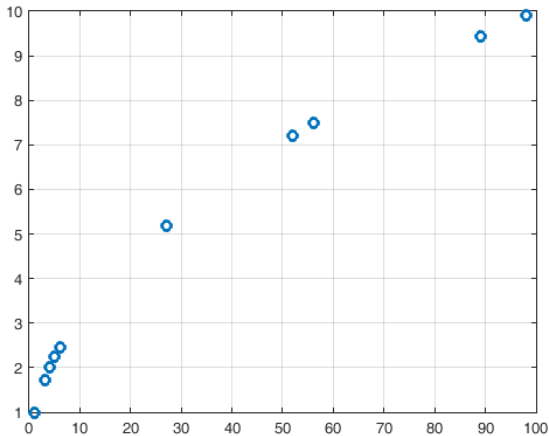
Ispis korenova u datoteku (MATLAB/QtOctave)

pr5.m

```
clear all, clc
% otvaranje datoteke za citanje
id1 = fopen('niz.txt', 'r');
a = fscanf(id1, '%f'); % citanje, ne mora 10x!
fclose(id1);          % zatvaranje
b = sqrt(a);
% crtanje
figure, plot(a,b,'o','linewidth',2)
% otvaranje datoteke za ispis
id2 = fopen('nizB2.txt', 'w');
fprintf(id2, '%f', b); % ispis niza b, ne mora 10x!
fclose(id2);          % zatvaranje
```



Ispis korenova u datoteku (MATLAB/QtOctave)



Slika: $y = \sqrt{x}$



Zadaci za vežbu

1. Izračunati površinu P i zapreminu V :
 - lopte zadatog poluprečnika R ,
 - kvadra zadatih dužina stranica a , b i c .
2. Izračunati vrednost faktoriijela $n!$.
3. Izračunati sumu brojeva u datoteci (brojeve učitavati dok se ne učitava 0).
4. Odrediti vrednost sume S pri uslovu: $a_n \geq \epsilon$, a broj x se unosi sa tastature.

$$S = \sum_n a_n = \sum_n (-1)^n \frac{x^n}{n!}$$

5. Proveriti da li se tačka $A(x, y)$ nalazi unutar kruga.

$$(x - x_c)^2 + (y - y_c)^2 = R^2$$



Zadaci za vežbu

6. Odrediti sumu:

- svih brojeva u matrici,
- brojeva u k -toj vrsti,
- brojeva na glavnoj dijagonali.

Unos: **read(*,*) ((A(i,j), j=1,N), i=1,M)**

7. Odrediti vrednost fje f u tački (x, y) gde je:

$$f(x, y) = \ln(x^2 + y^2) + \sqrt{x^4 + 5y^2} - \sin(xy), x > 0, y > 0$$

