

Füüsika üldkursus on juba läbitud ja meie kursus põhineb selle kursuse teadmistel. Kursus on rohkem praktiline kui teoreetiline. See tähendab et me lahendame konkreetseid füüsikalised ülesanded numbrilistel meetoditel. Teisisõnu kirjutame programme programmeerimiskeeltes Matlab, Fortran ja C. Programmeerimiskeel Fortran on suhteliselt lihtne ja väga efektiivne. Üle 90 protsente teaduse arvutustel on teostatud fortranikeeles. Ja ärge unustage et kõik programmeerimise keelte on väga sarnased. Fortranil kirjutatud programm saab lihtsalt konverteerida C keele ja vastupidi.

Korraldame iga loengu nii: 1) Me võtame konkreetne ülesande füüsikakursuselt, 2) Siis sõnastame selle matemaatiliselt, 3) Siis kirjutame Fortrani ja Matlabi koodi, 4) Seejärel tehakse arvutused, 5) Ja lõpus joonistame graafikuid ja analüüsime tulemusi.

Semestril toimub üks või kaks kontrolltööd. Kaheksandal nädalal ja semestri lõpus.

Otsene eksam ei toimu aga aprilli kuu lõpus ma anna igale tudengile lõputööteemat (mingi konkreetne füüsika ülesanne). Lõputöö tuleb esitada ja kaitsta enne sessiooni lõppu. Lõplik hinne sõltub kaitsmise tulemustest.

Lõputöö peab sisaldama:

1. Tiitelleht
2. Teoreetilised alused. Kõik matemaatilised väljendid koos teie selgitamisega.
3. Matlab, Fortran ja C koodid (kommentaaridega).
4. Tulemused (andmeid, graafikud jne.)
5. Kokkuvõte

Saab seda teha PDF formaadis ja saata mulle e-mailiks. Siis lepime kokku millal saab arutamise teha.

Alustame MATLABi uurimist liht programmidest. MATLABi skripti (programme) salvestame loodud skripti-failides. Ärge unustage et MATLAB on juhtumitundlik (case sensitive).

I. Matlabi põhioperaatorid ja käskud.

1. Põhilised matemaatilised operaatorid ja funktsioonid

clear – kustutab muutujad

whos – näitab muutujad

1.1 Maatriksite loomine

Näidised:

$$a=[1,2;3,4]; \quad a \rightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$b=[1,2,3,4]; \quad b \rightarrow [1 \ 2 \ 3 \ 4]$$

$$d=[1;2;3;4]; \quad d \rightarrow \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

d' - transposeerimine

inv(a) või d⁻¹ -pöördmatriks

$a*b \quad a'*b \quad a*b'$

det(a) – determinandi arvutamine

1.2 Numbrijada loomine

$x=alg:samm:lõpp$

$x=alg:lõpp$

Näidis:

$c=1:0.2:2;$

$[\quad c \rightarrow [11.2 \ 1.4 \ 1.6 \ 1.8 \ 2.0]$

1.3 Matemaatilised operaatorid ja väljendid

$+, -, /, ^, *, .^, .* , ./, \backslash$

1.4 Kompleksne numbrite loomine

$a=2+3*i$

$b=complex(2,3)$

conj(b) - kaaskompleksne

imag(b) – imaginaar osa

real(b) – reaalarv osa

1.5 Standartsed matemaatilised funktsioonid

1.5.1 Trigonomeetrilised funktsioonid

sin,sind,cos,cosd,tan,tand,acos,acosd,asin,asind,atan,atand

1.5.2 Eksponentsiaalsed ja logaritmilised funktsioonid

exp,sinh,cosh,tanh, asinh,acosh,atanh,log,log10,log2

1.5.3 Mõned teised funktsioonid

factorial(a) — faktoriaali arvutamine

a — täisarvuline massiiv

Näidis:

factorial(3)

power(a,n) — aste funktsioon

a — reaalarv, n — aste

Näidis: power(3,2)

sqrt(a) — ruutjuur

a - reaalarv

Näidis: sqrt(2.)

zeros -nulli maatriksi loomine

ones — maatriks koosneb ühikutest

rand — juhusliku numbri loommine vahemikus 0 kuni 1.

rand (n,m) – loob maatriksi juhuslikke numbritega

sum(a) — arvude jada summeerimine

a — matriks

max(a) – maksimaalne matriksi element

a — matriks

min(a)-maatriksi minimaalne element

a — matriks

norm(a)- maatriksi normi arvutamine

$$\text{norm}(a) = \sqrt{\sum_{i,j=1}^n a_{i,j}^2}$$

length(a) – matriksi pikkuste arvutamine

1.5.3.1 Funktsioonid kasutatakse andmete ümardamiseks

round, ceil, fix

1.5.4 Andmete esituse täpsuse muutumine

format *arg*

argumendiks saab kasutada:

short kuni 4 arvu

long kuni 15 arvu

Näidis:

short *e* või *g* või *eng*

long *e* või *g* või *eng*

1.5.5 Andmete lugemine failist ja salvestamine failiks

save option file_to_save data_for_save

a=**load**('failinimi')

Näidis:

a=rand(4);

save -ascii test.dat a

Näidis:

a=rand(4);

```
save -ascii test.dat a
b=load(a)
```

fscanf

Näidis:

```
a=rand(4);
save -ascii test.dat a
fid=fopen('test.dat');
b=fscanf(fid,'%f',[4 4])
```

fprintf

Näidis:

```
a=rand(4);
fid=fopen('test.dat');
fprintf (fid,'%f',a)
```

input(PROMPT,'s')

Näidis:

```
a=input('enter data=','s')
```

1.5.6 Andmete väljaprintimine kuvarile

disp — andmete väljundamiseks ekraanile

num2str — number -> tekst konverter

Näidis:

```
a=23.456;
disp(['a = ',num2str(a)])
```

1.6.1 Tsükkel

for i=n1:n2:n3

.....

operaatorid

.....

end

i – tsüklioperaatori muutuja

n1-alg väärtus

n2-lopp väärtus

n3- samm

1.6.2 IF lause (tingimussiirtelause)

log-loogiline avaldis

if log1

operaatorid

end

```
if log1  
operaatorid  
elseif log2  
operaatorid  
end
```

```
if log1  
operaatorid  
elseif log2  
operaatorid  
else  
operaatorid  
end
```

Loogikaavaldised

>	gt	<i>greater than</i>
<	lt	<i>less than</i>
>=	ge	<i>greater or equal</i>
<=	le	<i>less or equal</i>
~=	ne	<i>not equal</i>
==	eq	<i>equal</i>
	or	
&&	and	

1.6.4 Funktsioonid

```
function [y1,...,yn] =function_name (x1,...,xm)  
operaatorid  
end
```

```
function [y1,...,yn] =function_name (x1,...,xm)  
operaatorid  
return  
end
```

x₁,...,x_m - siseparametrid
y₁,...,y_n - väljundparametrid

Näidis:
Summerib matriksi elementid ja arvutab keskmist väärtust.

```
function [summa,kesk]=sum_func(a)  
summa=0.;  
n=length(a);  
for i =1,n  
summa=summa+a[i];  
end  
kesk=summa/n;
```

end

LIHT PROGRAMMID

Programm 1.

“hello word” väljundi väljumiseks.

disp – käsku kasutatakse andmete väljundamiseks.

disp ('hello world')

Programm 2

summeerime kaks real arvu.

num2str() - käsku arvu konverteerimiseks teksti.

a=2.;b=3;

c=a+b;

disp([num2str(a),'+',num2str(b),'=',num2str(c)])

Programm 3

Kahe real arvu summeerimine. Andmete lugemine klaviatuurist.

input(“text”) -käsku kasutatakse andmete lugemiseks klaviatuurist.

“text” - lisa seletus.

a=input('enter a=');

b=input('enter b=');

c=a+b;

disp([num2str(a),'+',num2str(b),'=',num2str(c)])

Programm 4

Loeme andmeid failist nimega test3.in.

load(faliname) – käsku kasutatakse andmete lugemiseks failist. Eeldatakse et andmed esitatud ASCII formaadis.

a – ühemõõtmiline massiv pikkusega 9, “load” käsku loob seda automaatselt.

sum(a) – MATLABi käsku massiivi elemendi summeerimiseks.

Meie juhul faili test3.in sisu on sellist:

1 2 3 4 5 6 7 8 9

a=load('test3.in')

c=sum(a);

disp(['sum=',num2str(c)])

Programm 5

Numbrite summeerimiseks. Andmed on salvestatud massiivis nimega **a**.

Massiivi pikus on 7 aga elementid nummeeritatakse 00 kuni 6-ni (sama nagu C,Python-keeletes).

```

a=[1,2,3,4,5,6,7];
n=length(a);
c=sum(a);
disp(['sum=',num2str(c)])

```

Programm 6

```

a=load('test4b.in');
n=length(a)
c=sum(a);
disp(['sum=',num2str(c)])

```

Programm 7

```

a=load('test4b.in');
[sum,av]=mysum(a)

```

```

function [sum,av]=mysum(a)
n=length(a);
sum=0.;
for i=1:n
sum=sum+a(i);
end
av=sum/n;
end

```

Programm 8

```

myfact(5)
function myfact(n)
if fix(n) ~= n
disp('argument not integer' )
return
end
if (n==0 || n==1)
disp(['0!=',num2str(1)]);
return
elseif n<0
disp('argument is negativ');
return
end
ff=1;
for i=1:n
ff=ff*i;
end
disp([num2str(n),'!=',num2str(ff)]);
end

```

Programm 9

```

myroot(1,3,1)
function [x1,x2]=myroot_1( a,b,c)
diskr=b*b-4*a*c;

```

```

if(diskr>0)
x1=(-b+sqrt(diskr))/(2.*a);
x2=(-b-sqrt(diskr))/(2.*a);
return
end
if(diskr==0)
x1=-b/(2.*a);
disp(['x1=x2=',num2str(x1)]);
return
end
x1=-b/(2.*a);
x2=(sqrt(-diskr))/(2.*a);
end

```

```

function myroot_2( a,b,c)
diskr=b*b-4*a*c;
if(diskr>0)
x1=(-b+sqrt(diskr))/(2.*a);
x2=(-b-sqrt(diskr))/(2.*a);
disp(['x1=',num2str(x1),' x2=',num2str(x2)]);
return
end
if(diskr==0)
x1=-b/(2.*a);
disp(['x1=x2=',num2str(x1)]);
return
end
x1=-b/(2.*a);
x2=(sqrt(-diskr))/(2.*a);
disp(['x1,x2=',num2str(x1),' +/-i',num2str(x2)]);
end

```

```

function [x1,x2]=myroot_3( a,b,c)
diskr=b*b-4*a*c;
x1=(-b+sqrt(diskr))/(2.*a);
x2=(-b-sqrt(diskr))/(2.*a);
end

```

Program 10
a=[1,2,3];
b=[0,-3,2];
myangle(a,b)

```

function myangle( a,b )
ang=acosd(dot(a,b)/(norm(a)*norm(b)));
disp(['angle=',num2str(ang)]);
end

```

Program 11
fun = @(x) power(x,2);

```
format long  
res=integral(fun,0,1)
```

Program 12

```
a = integral(@func,0,1)*4  
function [y] = func(x)  
y = sqrt(1-power(x,2)) ;  
end
```

Program 13

```
function [max,min ] = mymaxmin(a)  
max=-1.e-32;  
min= 1.e+32;  
n=length(a);  
for i=1:n  
if (a(i) < min)  
min=a(i);  
end  
if (a(i) > max)  
max=a(i);  
end  
end
```

Program 14

```
x=-720:10:720;  
y=-720:10:720;  
[X,Y]=meshgrid(x,y);  
R=sqrt(power(X,2)+power(Y,2));  
Z=sind(R);  
surf(X,Y,Z)
```

Program 15

```
theta = linspace(-pi,pi,1000);  
xc = cos(theta);  
yc = -sin(theta);  
plot(xc,yc);  
axis equal  
xt = [-1 0 1 -1];  
yt = [0 0 0 0];  
hold on  
t = area(xt,yt); % initial flat triangle  
hold off  
for j = 1:length(theta)-10  
xt(2) = xc(j); % determine new vertex value  
yt(2) = yc(j);  
t.XData = xt; % update data properties  
t.YData = yt;  
drawnow limitrate % display updates  
end
```

II. Programmeerimiskeel FORTRAN

1. Muutujad

Muutujate deklareerimiseks Fortranis kasutatakse järgmised operaatorid (muutujate nimed ei ole “case sensitive” ja kõik tühikud ignoreeritakse):

integer
integer*4
integer*8
integer*16
real
real*4
real*8
real*16
complex
complex*8
complex*16
complex*32
logical
character*n

Eksisteerib ka teine(Fortran 90) variant:

real:: a,b,c

Näidis:

real a,b,c või **real*4 a,b,c** – loodud reaali muutuja pikkusega 4 baiti
real *8 a,b,c – loodud reaali muutuja pikkusega 8 baiti
real *16 a,b,c – loodud reaali muutuja pikkusega 16 baiti
integer i,j,k – loodud täisarvuline muutuja pikkusega 4 baiti
integer*8 i,j,k või **integer*4 i,j,k** – loodud täisarvuline muutuja pikkusega 8 baiti
character*6 a – loodud teksti muutuja pikkusega 6-baiti
complex*16 abc – loodud kompleksne muutuja **abc** kahekordse täpsusega. Kompleksne arv koosneb kahest reaalarvust iga pikkusega 8 baiti.

Piirväärtused:

					mantiss (positsioonide arv)
real*4	1.175494 10 ⁻³⁸	kuni	3.402823 10 ⁺³⁸		~8
real*8	2.225074 10 ⁻³⁰⁸	kuni	1.797693 10 ⁺³⁰⁸		~16
real*16	3.362 10 ⁻⁴⁹³²	kuni	1.20 10 ⁺⁴⁹³²		~32
integer *4	-2147483648	kuni	2147483647		~8
integer*8	-9223372036854775808	kuni	9223372036854775807		~16

2. Maatriksid

real*4 a(10), b(3,4) – loodud massivid: ühemõõtmeline massiv **a** pikkusega 10 ja kahemõõtmeline massiv **b** pikkustega 3 (reade arv)x4(tilpade arv)

3. Matemaatilised operaatorid ja funktsioonid

=,+,*,/

** -aste

sqrt,sin,cos,tan,cotan,acos,atan,asin,exp,alog

Peaaegu kõik standardfunktsioonid leiate siit:

<https://www.nsc.liu.se/~boein/f77to90/a5.html>

4. Andmete lugemine ja salvestamine

Näidis:

*print *a,b,c,"hello world"*

read () a,b,c*

read ()

open(n, file="filename") n – täisarv(faili viite) vahemikus 1...9999

*read(n, *) a,b,c*

*write(n, *) a,b,c*

Näidis:

open(12, file="input.dat")

input faili viite on loodud

open(20, file="output.dat")

output faili viite on loodud

*read(12, *) a,b*

loeme kaks reaalarvu input failist

*write(20, *) a,b*

salvestame kaks reaalarvu output failis

4. IF laused

if (log) then

operaatorid

endif

if (log) then

operaatorid

else

operaatorid

endif

if (log) then

operaatorid

else if (log)

operaatorid

else

operaatorid

endif

Näidis:

if(a>2) then

b=a

else

b=a*3

endif

Loogikaavaldised

> gt

< lt

>= ge

<= le

!= ne

== eq

|| or

&& and

Näidis:

$a > b \parallel 2 \leq 3 \equiv a.gt.b.or.2.le.3$

5. Tsükli operaatorid

do n=n_alg,n_lõpp,n_samm

operaatorid

enddo

do n=n_alg,n_lõpp

operaatorid

enddo

while (log) do

operaatorid

enddo

do while (log)

operaatorid

enddo

FORTTRAN-i programmide mitmesuguseid näiteid on salvestatud eraldi failiks!