Numeričke metode

Aleksandar Maksimović
IRB

vektori

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Vektor u 3D prostoru.

C:
```
int v1[3];
v1[0]=a;v1[1]=b;v1[2]=c;
```

Fortran:
```
INTEGER V1(3)
V1(1)=a
V1(2)=b
V1(3)=c
```

Skalarni produkt dva vektora
npr. v1 . v2

```
{a, b, c} . {d, e, f}
```

$$a\,d + b\,e + c\,f$$

C: v1[0]*v2[0]+v1[1]*v2[1]+v1[2]*v2[2];

Fortran: v1(1)*v2(1)+v1(2)*v2(2)+v1(3)*v2(3);

library

```
   subroutine readmatrix(a,n,m,unit)
   integer n,m,unit
   real a(n,m)
   Do i=1,n
   read (unit,*)  (a(i,j),j=1,m)
   end do
   end

   subroutine displaymatrix(a,n,m)
   integer n,m,unit
   real a(n,m)
   Do i=1,n
   write (6,*)  (a(i,j),j=1,m)
   end do
100  Format (5(2x,f10.5))
   end
```

lib1.f

lib1.c

```
#include <stdio.h>
#include <stdlib.h>
#include "lib1.h"
void readmatrix(float a[][50], int n, int m,FILE* fp)
{
 int k, l;
 for (k=0;k<n;k++)
   for (l=0;l<m;l++) fscanf(fp,"%f ",&a[k][l]);
}
void displaymatrix(float a[][50], int n, int m)
{
 int k, l;
 for (k=0;k<n;k++)
   { for (l=0;l<m;l++) printf("%f ",a[k][l]);
     printf("\n");
   }
}
```

program

```
void readmatrix(float a[][50]  , int , int ,FILE* );
void displaymatrix(float a[][50] , int , int );
```

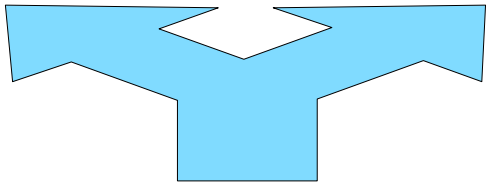
lib1.h

```
#include <stdio.h>
#include <stdlib.h>
#include "lib1.h"

int main()
{
        int j,k,l,m,n;
        float b[50];
        FILE *fp;
```

mainlib1.c

```
program ReadDisplay

Implicit real (a-h,o-z)
integer n,m,unt,l
Dimension b(50)
```

mainlib1.f



Deklaracije, polja odnosno vektora b cijelih brojeva, file pointera

program

```
if ((fp = fopen("matrixVector.dat","r")) == NULL)
            error("Data file matrx1.dat not found\n");

          while (!feof(fp)) {
           fscanf(fp,"%d %d ",&n,&m);
           readmatrix(a,n,m,fp);

           fscanf(fp,"%d ",&l);
           for (k=0;k<l;k++) fscanf(fp,"%f ",&b[k]);
          }
```

Otvori file

n, m, Matrica

n, vektor

```
      open(unit=8,file="matrixVector.dat",status="unknown")

       read (8,*) n,m
       unt=8
       call readmatrix(a,n,m,unt)

c   vector
       read (8,*) l
       read (8,*)  (b(j),j=1,l)
```

matrixVector.dat

```
2 3
1.0 2.0 3.0
1.0 1.0 1.0
4
1.0 2.0 3.0 4.0
```

C: fp je FILE POINTER

F77: 8 unit

n, m - cijeli brojevi

a - dvodimenzionalno polje - matrica

b- jednodimenzionalno polje - vektor

program

Napravili smo program readAnddisplay modularnim, tj. funkcije su u posebnom file-u napisane kao podprogrami.

C EXE: gcc -o mainlibC mainlib1.c lib1.c

./mainlibC

F77 EXE: f77 -o mainlibF mainlib2.f lib1.f

./mainlibF

Izvršavanje programa ispisuje vektor i matricu definiranu u ulaznom file-u.

nove funkcije za vektor

Dodajemo 3 nove funkcije.

Kompletan header file za C "lib1.h"

```
void readmatrix(float a[][50]  , int , int ,FILE* );
void displaymatrix(float a[][50] , int , int );
void readvektor(float a[], int, FILE* );
void displayvektor(float a[],int);
void readline(FILE*);
```

readmatrix : (1) 2d polje A dimenzija (2,3) NxM, (4) file pointer ili unit (integer)

displaymatrix : (1) 2d polje A dimenzija (2,3) NxM

readvektor : (1) 1d polje A dimenzije (2) N, (3) file pointer ili unit (integer)

displayvektor : (1) 1d polje A dimenzije (2) N

readline: (1) file pointer ili unit (integer)

read(matrix/vektor): pročitaju matricu vektor iz otvorenog file-a

readline: pročita liniju (u C 80 max, duljina) iz file-a

tok programa

- Program
  - deklaracija varijabli (polja)
    - za svaku veličinu iz file-a 1 varijabla, za rezultat nova varijabla
  - Otvori file
  - preskoči komentar/ pročitaj dimenziju u deklariranu varijablu / pročitaj vektor ili matricu
  - izvrši operacije
  - prikaži rezultat

modularni proram

mainlib2.c

```
if ((fp = fopen("VM.dat","r")) == NULL)
  error("Data file matrx1.dat not found\n");

  readline(fp);
  fscanf(fp,"%d ",&L);
  readvektor(b,L,fp);

  displayvektor(b,L);

  readline(fp);
  fscanf(fp,"%d %d ",&n,&m);
  readmatrix(a,n,m,fp);

  displaymatrix(a,n,m);
  fclose(fp);
```

mainlib2.f

```
open (unit=8,file="VM.dat",status="unknown")
      fp=8
c komentar
      call readline(fp)
c    vector
      read (8,*) L
      call readvektor(b,L,fp)
      call displayvektor(b,L)
      call readline(fp)
c matrica
      read (8,*) n,m
      call readmatrix(a,n,m,fp)

      call displaymatrix(a,n,m)
      close(8)
```

Zadatak

- Pročitati V2M2.dat
  - Format: komentar, 1. vektor, komentar, 2. vektor, komentar, 1. matrica, komentar, 2. matrica
- Izračunati
  - Skalarni produkt 2 vektora
  - produkt 1. vektora i 1. matrice
  - produkt 2 matrice

deklaracije

1. Trebamo 2 1d polja za vektore V1, V2 i dva skalara za dimenzije L1, L2
2. Trebamo 2 2d polja za matrice MA1, MA2 i 4 cijela broja za dimenzije n1,m1,n2,m2. Matrice ne mogu biti u f77 M1 i M2, zbog dimenzija.
3. Skalarni produkt je skalar, SPROD
4. Matrica puta Vektor (2, 3) x (3) = (2) vektor dimenzije 2, MatVec
5. Matrica puta Matrica je opet Matrica MM dimenzije (2,3)x(3,2)=(2,2)

3. SPROD = V1(i)*V2(i), i=1,2,3,..,L1, L1 mora biti jednak L2
4. MatVec(i)=M1(i,j)*V1(j), j=1,2,...L1, L1 mora biti jednak m1
   i=1,2,..n1
5. MM(i,j)= M1(i,k)*M2(k,j), k =1,2,.. m1, m1=n2. dimenzija n1xm2
   i=1,..n1. j=1,..m2

deklaracije

Dodatne varijable:

File pointer fp

Cijeli brojevi i,j,k,l za petlje

float tmp dodatna varijabla, ako zatreba itd...

mainlib3.c

```
#include <stdio.h>
#include <stdlib.h>
#include "lib1.h"

int main()
{
  int L1, L2, n1,m1, n2,m2;
  float V1[10],V2[10], SPROD, MatVec[10];
  float MA1[10][50],MA2[10][50],MM[10][50];
  FILE *fp;
  int i,j,k;
  float tmp;

  if ((fp = fopen("V2M2.dat","r")) == NULL)
    error("Data file matrx1.dat not found\n");

  /* read all data */
      readline(fp);
      fscanf(fp,"%d ",&L1);
      readvektor(V1,L1,fp);
      displayvektor(V1,L1);
      readline(fp);
učitavanje 2. vektora
```

```
 readline(fp);
fscanf(fp,"%d %d ",&n1,&m1);
readmatrix(MA1,n1,m1,fp);

displaymatrix(MA1,n1,m1);

readline(fp);
fscanf(fp,"%d %d ",&n2,&m2);
readmatrix(MA2,n2,m2,fp);

displaymatrix(MA2,n2,m2);
/* close file */

fclose(fp);
/* COMPUTE */
tmp=0;
for( i=0;i<L1;i++)
SPROD=tmp;
printf("Skalarni produkt je %f\n",SPROD);
```

```
 /* Matrix Vector */
     tmp=0;
     for (i=0; i< n1;i++)
      {
        MatVec[i]=0;
         for (j=0; j< m1;j++)
MatVec[i]=MatVec[i]+MA1[i][j]*V1[j];
      }
     printf("Matrica x Vektor \n");
     displayvektor(MatVec,n1);

      /* Matrix Matrix */
     for (i=0; i< n1;i++)
      {
        for (j=0; j< m2;j++)
      {
        MM[i][j]=0;
tmp=tmp+V1[i]; for (k=0; k< m1;k++)
MM[i][j]=MM[i][j]+MA1[i][k]*MA2[k][j];
      }
      }
      printf("Matrica x Matrica \n");
     displaymatrix(MM,n1,m2);
     return 0;
}
```

mainlib3.f

```
 program vektorMatMat
      integer  L1, L2, n1,m1, n2,m2
      real  V1(10),V2(10), SPROD, MatVec(10)
      real  MA1(10,50),MA2(10,50),MM(10,50)
      integer i,j,k,fp
      real tmp
      open (unit=8,file="V2M2.dat",status="unknown")
      fp=8
c komentar
      call readline(fp)
c   vector
      read (8,*) L1
      call readvektor(V1,L1,fp)
      call  displayvektor(V1,L1)
.......
      call readline(fp)
      read (fp,*) n2,m2
      call readmatrix(MA2,n2,m2,fp)
      call displaymatrix(MA2,n2,m2)
      close(fp)

c Calculate
      SPROD=0
      do i=1,L1
        SPROD=SPROD+V1(i)
      end do
      print *, "Skalarni produkt"
      print *, SPROD

      call multMatrixVector(MA1,V1,MatVec,n1,m1)
      print *, "Matrica x Vektor"
      call displayvektor(MatVec,n1)

c Pomnozi matrice
       print *, "Matrica x Matrica"
      call multMatrixMatrix(MA1,MA2,MM,n1,m1,n2,m2)

       call displaymatrix(MM,n1,m2);
      end program
```

mainlib3.f

```
subroutine multMatrixMatrix(MA1,MA2,MM,n1,m1,n2,m2)
    INTEGER n1,m1,m2,n2,i,j,k
    REAL MA1(n1,m1),MA2(n2,m2),MM(n1,m2)

    Do i=1,n1
      Do j=1,m2
        MM(i,j) = 0.0
        Do k=1,m1
          MM(i,j) = MM(i,j) + MA1(i,k)*MA2(k,j)
        End Do
      End Do
    End Do
    end
```

```
subroutine multMatrixVector(a,b,c,n,m)
    integer n,m
    real b(m),c(m)
    real a(n,m)

    do i=1,n
      c(i)=0.0
      do j=1,m
        c(i)=c(i)+a(i,j)*b(j)
      end do
    end do
    end
```

EXE programi

Fortran: f77 -o mainlib3F mainlib3.f lib1.f

C: gcc -o mainlib3C mainlib3.c lib1.c

```
./mainlib3F
 1.  2.  3.
 3.  2. -1.
 1.  2.  3.
 1.  1.  1.
 1.  2.
 1.  1.
 3.  1.
Skalarni produkt
 6.
Matrica x Vektor
 14.  6.
Matrica x Matrica
 12.  7.
 5.  4.
```

```
./mainlib3C
1.000000 2.000000 3.000000
3.000000 2.000000 -1.000000
1.000000 2.000000 3.000000
1.000000 1.000000 1.000000

1.000000 2.000000
1.000000 1.000000
3.000000 1.000000

Skalarni produkt je 6.000000
Matrica x Vektor
14.000000 6.000000
Matrica x Matrica
12.000000 7.000000
5.000000 4.000000
```

Zadaci

- VMV.dat ima slijedeći format:
  - vektor duljine 3
  - matricu 3x2
  - vektor duljine 3
- Napiši program koji ispiše skalarni produkt ova dva vektora
- Napiši program koji množi transponiranu matricu i vektor
- Pošaljite mail na Aleksandar.Maksimovic@irb.hr sa programima u attachmentu ili iskopirajte u /home/maks/vjezbe/VasDir i obavijestite me mailom