INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Inovace a zvýšení atraktivity studia optiky

reg. č.: CZ.1.07/2.2.00/07.0289

Úvod do programování

Lekce 7

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

Octave, Matlab – vyšší programovací jazyky pro numerické výpočty

základní charakteristiky

- oproti C zjednodušená syntaxe, proměnné není nutno explicitně definovat
- uzpůsobeny pro snadnou práci s vektorovými a maticovými objekty
- obsahují funkce pro grafický výstup a rozsáhlé knihovny pro numerické výpočty
- program, které plně nevyužívá maticovou aritmetiku běží ve srovnání s obdobným programem napsaným v jazyku C velmi pomalu

příklad: program vynásobí dvě matice 3x3 a výslednou matici vypíše na obrazovku

```
#include<stdio.h>
#define DIM 3  //  dimenze matice

main(){
  // reprezentuji matice jako dvourozmerne pole
  double a[DIM][DIM]={{0,0,1},{0,-1,0},{-1,0,0}};
  double b[DIM][DIM]={{0,0,-1},{0,-1,0},{1,0,0}};
  double c[DIM][DIM];
  int i,j,k;
  for(i=0;i<DIM;i++){
    for(j=0;j<DIM;j++){
      c[i][j]=0;
      for(k=0;k<DIM;k++){   // radek "a[i]" nasobi sloupec "b[j]"
     c[i][j]+=a[i][k]*b[k][j];
      }
    }
  }
  for(i=0;i<DIM;i++){
    for(j=0;j<DIM;j++)
      printf("%f ",c[i][j]); // tisk prvku matice
    printf("\n");
  }
}
```

totéž v Octave/Matlabu

```
function nasobeni
  a=[0,0,1;0,-1,0;-1,0,0]; # strednik potlacuje vystup
  b=[0,0,-1;0,-1,0;1,0,0];
  c=a*b                    # automaticky zobrazi vysledek
endfunction
```

práce s prvky vektoru/ matice

```
a=[0,0,1;0,-1,0;-1,0,] # matice
v=[1;0;-1]             # sloupcový vektor
dolni:krok:horni       # rozsah (range)
-1:0.5:1  # generuje: [-1,-0.5,0,0.5,1]
a(1,3)    # prvek na prvním řádku a třetím sloupci: 1
a(2,:)    # druhý řádek: [0,-1,0]
a(1:2,1:2)# submatice 2x2 v levém horním rohu: [0,0;0,-1]
a*v       # součin matice a vektoru: [-1;0;-1]
v.*v      # součin po elementech: [1;0;1]
v'        # transponovaný vektor: [1,0,-1]
v'*v      # skalární součin: 2
```

```
zeros(dim1,dim2)  # nulová matice dim1 x dim2
ones(dim1,dim2)   # matice vyplněná jednotkami
eye(dim1,dim2)    # diagonální jednotková matice

inverse(a) # vypočte inverzní matici (totéž co a^-1)
trace(a)   # stopa
det(a)     # determinant
eig(a)     # vlastní čísla
logm(a)    # logaritmus matice
sqrtm(a)   # odmocnina matice
sum(a,dim) # součet vzhledem k dimenzi dim(1 = po sloupcích)
```

přehled základních řídících struktur

Octave/Matlab

příkaz `if`

```
if (podmínka)
  tělo
endif
```

příkaz `while`

```
while (podmínka)
  tělo
endwhile
```

příkaz `for`

```
for var=výraz # postupně dosa-
  tělo        # zuje sloupce
endfor
```

příklad na `for`

```
soucet=0;
for i=1:2:100
  soucet=soucet+i;
endfor
```

totéž ale mnohem efektivněji

```
soucet=sum(1:2:100)
```

jazyk C

```
if (podmínka){
  tělo
}


while (podmínka){
  tělo
}


for(start,podmínka,příkaz){
  tělo
}


soucet=0;
for(i=1;i<=100;i+=2){
  soucet+=i;
}
```

příklad: výpočet histogramů pro soubor `5x10000` náhodných čísel

```
function histogramy
  MAX=20
  fr=fopen("random.txt","r");
  data=fscanf(fr,"%f",[10000,5]); # nacita matici po sloupcich
  fclose(fr);
  data=data';           # data bude mit 5 radku a 10000 sloupcu
  histogram=zeros(5,MAX+1);  # cetnosti nuly az MAX
  for i=0:MAX                # pocitam cetnost hodnoty "i"
    kde=(data==i);      # kde_i ma "1" na miste "i" jinak nuly
    histogram(:,i+1)=sum(kde,2);  # scita jednicky na radcich
```

```
  endfor
  for i=1:5
    bar(0:MAX,histogram(i,1:MAX+1)); # kresli histogram
    pause(2);           # ceka 2 sekundy
  endfor
endfunction
```

příklad: počítá Fraunhoferovu difrakci na čtvercovém otvoru

```
function difrakce
  pupila=zeros(100,100);  # propustnost 0=clona, 1=otvor
  ctverec=ones(10,10);
  pupila(1:10,1:10)=ctverec; # vkladame otvor do pupily
  vystup=fft2(pupila);    # Fraunhoferova aproximace
  obrazec=abs(vystup);    # budeme kreslit abs. hodnotu
  # ted posuneme stred obrazce do stredu matice
  z=shift(obrazec,50);    # posun radku
  z=shift(z',50)';        # posun sloupcu
  # generujeme matice souradnic x a y
  [x,y]=meshgrid(1:100);
  mesh(x,y,z);            # kreslime plochu
endfunction
```

příklad: modifikace pro případ kruhového otvoru

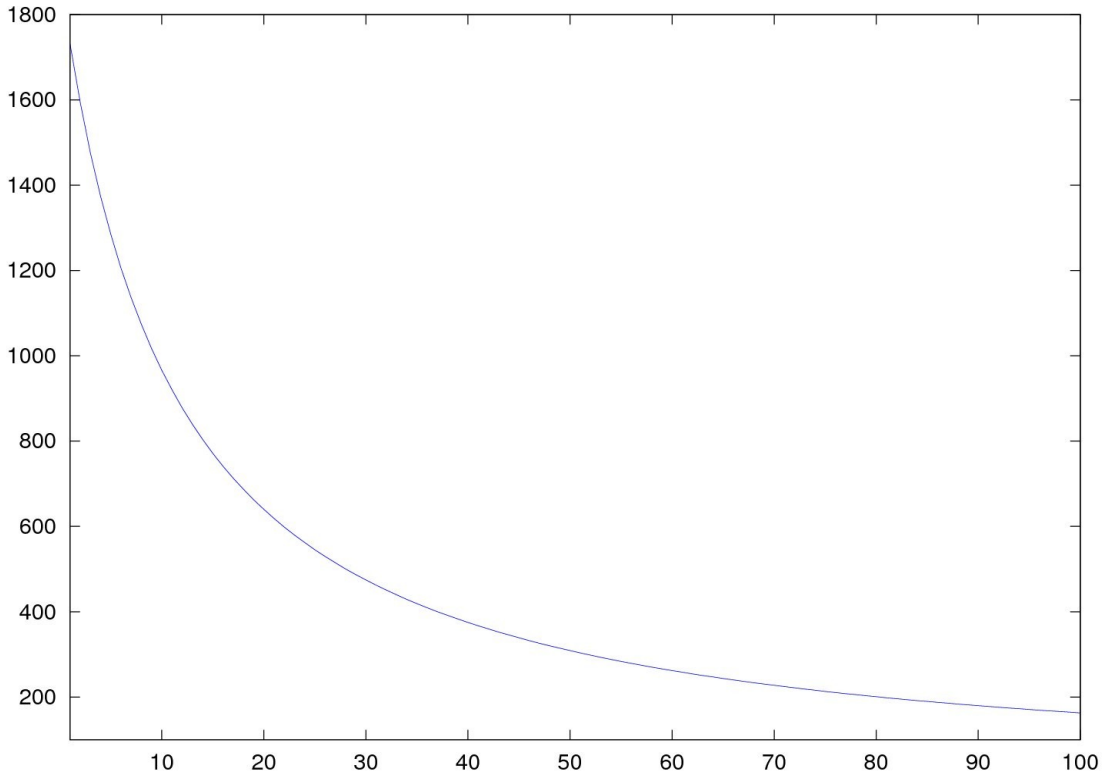
```
function difrakce2
  [x,y]=meshgrid(-50:50);
  pupila=sqrt(x.^2+y.^2)<5; # uvnitr kruhu 1 jinak 0
  obrazec=abs(fft2(pupila));
  z=shift(shift(obrazec',50)',50);
  mesh(x,y,z);
endfunction
```

příklad: vypočte ohniskovou vzdálenost objektivu, parametry čte ze souboru

```
function ohnisko
  fr=fopen("objektiv.txt","r");
  pocet=fscanf(fr,"%d",[1,1]);
  fprintf("pocet rozhrani: %d\n",pocet);
  data=fscanf(fr,"%f",[3,pocet])';
  fclose(fr);
  dt=1:100;         # mezera
  focal=zeros(1,100);
  for j=dt
    abcd=eye(2,2);
    n=1;           # paprsek prichazi ze vzduchu
    for i=1:pocet
      nn=data(i,2);       # index lomu za rozhranim
      r=data(i,1);        # polomer krivosti
      t=data(i,3)+dt(j); # vzdalenost rozhrani
      diopt=(nn-n)/r;     # mohutnost rozhrani
      refr=[1,0;-diopt,1]; # refrakcni matice
      tran=[1,t/nn;0,1];   # translacni matice
      abcd=tran*refr*abcd; # abcd matice
      n=nn;       # "nn" bude index lomu pred dalsim rozhranim
    endfor
    focal(j)=-1/abcd(2,1);  # ohniskova vzdalenost pro mezeru dt
  endfor
```

```
  plot(dt,focal);
  print -deps graf.eps    # tiskne graf do souboru
endfunction
```

např. objektiv ze dvou menisků

```
objektiv.txt
4
-80 1.5 10
-90 1 100
80 1.5 10
90 1 0
```

funkce `ohnisko` vytvoří výstup