G N U   O C T A V E

% označava početak komentara

; na kraju naredbe (retka) = ne ispisuj rezultat na ekran

Osnovne računske operacije možemo provoditi sa +, -, *, / kad god su stvari dobro definirane i na brojevima, vektorima, matricama

Potenciranje možemo koristiti a^b (baza je a, a eksponent b)

drugi korijen od x: sqrt(x)

n-ti korijen od x = nthroot(x, n)

clear all; % čisti sve varijable i konstante iz radnog prostora (memorije)

clc; % *clear screen* u komandom prozoru

x = 1; % definiramo varijablu *x* kojoj postavljamo vrijednost na 1

x = [1, 0, 1, 3]; % vektor redak (1, 0, 1, 3), isto i bez zareza
x(i); % za vektor x, pristupamo njegovoj i-toj komponenti

Za x = (1, 2, 3, 4, 5, 6) je x(1)=1, x(2) = 2, …, x(6) = 6. Primijetite da se indeksiranje vrši počevši od 1, a ne kao u C-u od 0
x = [0:1:5]; % kreiramo vektor x(0, 1, 2, 3, 4, 5), tj. 0 je početna vrijednost (x(1)), korak je 1, a kraj 5 (x(6))

' % znak za transponiranje (hermitsko adjungiranje ako radimo sa kompleksnim brojevima) ako ga koristio iza vektora ili matrice A' = A^T

A = [1 2 3; 4 5 6; 7 8 9]; % matrica $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, tj. unutar [ ] znak ; znači novi redak
A(i, j) % pristupamo elementu na mjestu (i, j)
A(2, 3) = 6 za gore definiranu matricu A
A(i : j, k : l) označava blok matrice A gdje uzimamo retke od i do j, a za stupce od k do l
A(1:3, 2:3) = $\begin{bmatrix} 2 & 3 \\ 5 & 6 \\ 8 & 9 \end{bmatrix}$, A(1, 1:2) = $\begin{bmatrix} 1 & 2 \end{bmatrix}$

y = lagrange(x, x0, y0) % pozivamo funkciju lagrange kojoj su ulazni podaci x, x0 i y0 (moramo ih definirati prije poziva), a izlazni podatak y, nema ; iz naredbe pa će se y odmah ispisati u komandnom prozoru

[n, m] = size (A); % n = broj redaka matrice/vektora A, m = broj stupaca matrice/vektora A

n= size(A, 1); % broj redaka matrice/vektora A

m = size(A, 2); % broj stupaca matrice/vektora A

```
for  i = 1 : n
        naredbe
end;
```
% petlja for, i = 1, 2, 3, 4, …, n

% for i = 1:2:n   znači početna vrijednost je 1, korak 2, a kraj je u n, tj. i = 1, 3, 5, 7, …, n ako je n neparan, inače zadnja vrijednost je n-1 za n paran

---

```
If  i == j
        naredbe
end;
```

% if uvjet, primijetite da također koristimo == za jednakost

% za različito koristimo ~=, veće >, manje <

---

y = y + z je isto kao y+=z,   y = y*z je isto kao y*=z

---

x = A\b;   % x je rješenje sustava $Ax = b$

[L, U] = lu(A); % dobivamo LU dekompoziciju matrice A, gdje je L donjetrokutasta, a U gornjetrokutasta matrica

% za dodatne opcije pogledati help lu

[Q, R] = qr(A)   % QR faktorizacija matrice A, vraća matrice ortogonalnu Q i gornjetrokutastu R

% za dodatne opcije pogledati help qr

[U, S, V] = svd(A)   % singularna dekompozicija matrice A (A = USV), dobivamo unitarne matrice U i V, te dijagonalnu S koja sadrži singularne vrijednosti na dijagonali

% za dodatne opcije pogledati help svd

I = eye(n); % jedinična matrica reda n (npr., za n = 3 $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$)

A = zeros(n,m); % nul - matrica A s n redaka i m stupaca

A = diag([1 2 3]); % kreiramo dijagonalnu matricu A koja na dijagonali sadrži vektor definiran u []

---

disp(' ')   % ispisujemo tekst unutar ' ' u komandnom prozoru

---

x = abs(y); % x postaje apsolutna vrijednost (modul) od y

---

norma = norm(x, 2); % računa 2-normu vektora ili matrice x

% za dodatne opcije pogledati help norm

---

y = max(x);   % y je maksimalna vrijednost u vektoru x

y = max(A); % y je vektor kojemu je y(i) maksimalna vrijednost u i-tom stupcu

y = max (max(A)); % y je masimalna vrijednost u matrici A
