# OCTAVE

## Kompleksni brojevi, suma, proizvod, integrali, diferencijali, simbolički račun

Autor: Nikola Žarić

# Kompleksni brojevi

- Fleksibilnost MATLAB-a proizilazi iz činjenice da su dozvoljene operacije sa kompleksnim brojevima.
- Kompleksni brojevi se mogu generisati kao:

  ***z=a+b*i***, a, b-realni i imaginarni dio

  ili u obliku:

  ***w=r*exp(i*fi)***, r, fi- moduo i argument kompleksnog broja.

- Zbog različite notacije u literaturi, imaginarna jedinica je u MATLAB-u prethodno definisana kao permanentna veličina (kao što je to ura|eno sa *eps*, *pi* i sl.), i označena sa *i* i *j*. Korišćena je klasična definicija, tako da je:

  ***»i=sqrt(-1)***

  dok drugi više vole oznaku *j*:

  ***»j=sqrt(-1)***

- Mi ćemo ovdje koristiti oznaku *i*.

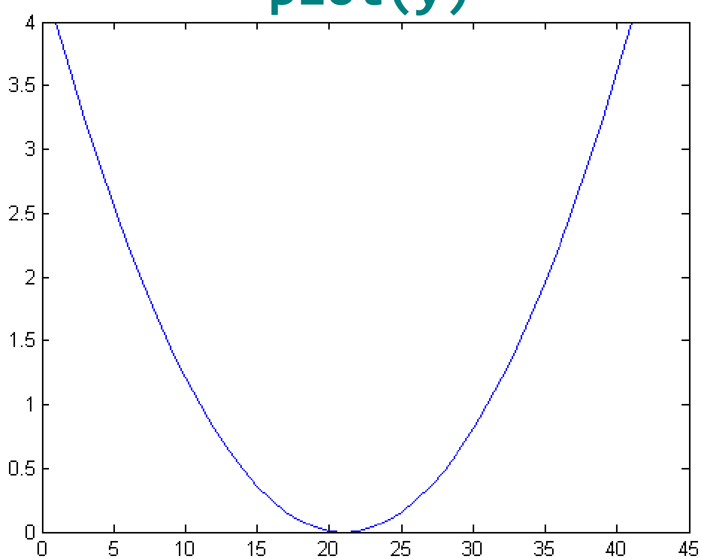
# Kompleksni brojevi

- `plot(y)` – crtanje vektora **y** u zavisnosti od rednog broja elementa.
- `plot(x,y)` – crtanje funkcije **y** u zavisnosti od nezavisno promenljive **x**.
- Ukoliko grafički prozor nije otvoren, `plot` otvara novi grafički prozor.
- Primjer:

```
x = -2 : 0.1 : 2;
y = x .^ 2;
```

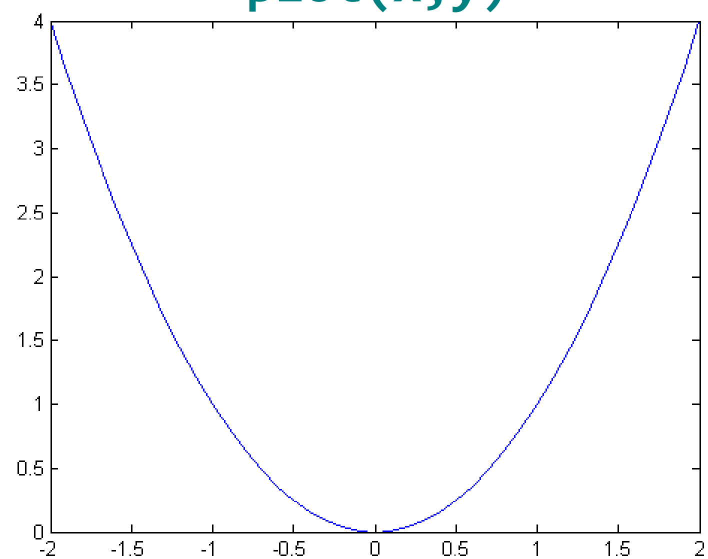
Obratiti pažnju na operator tačka, koji je neophodan za definisanje vektora funkcije.

**plot(y)**




Redni broj odbirka po x-osi

**plot(x,y)**




Vrijednost promjenljive x po x-osi

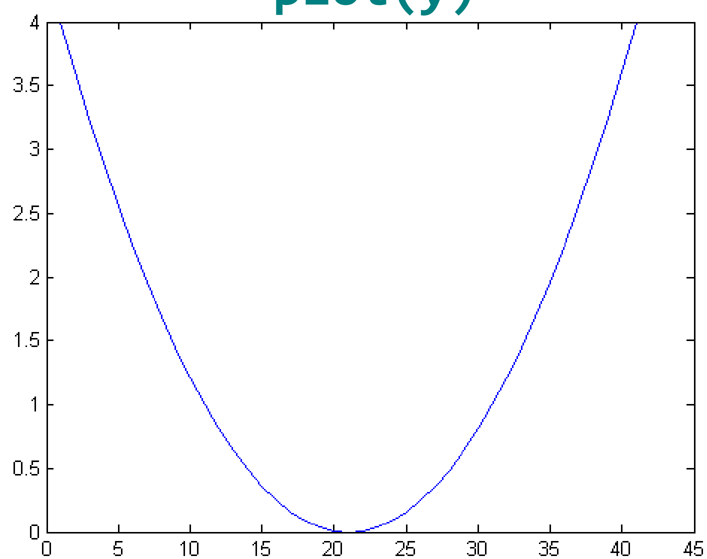
# Kompleksni brojevi

- **plot(y)** – crtanje vektora **y** u zavisnosti od rednog broja elementa.
- **plot(x,y)** – crtanje funkcije **y** u zavisnosti od nezavisno promenljive **x**.
- Ukoliko grafički prozor nije otvoren, `plot` otvara novi grafički prozor.
- Primjer:

```
x = -2 : 0.1 : 2;
y = x .^ 2;
```

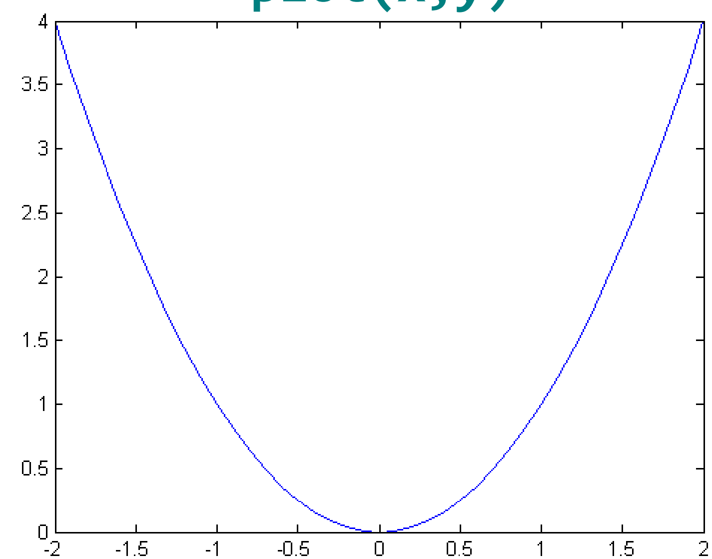
Obratiti pažnju na operator tačka, koji je neophodan za definisanje vektora funkcije.

**plot(y)**




Redni broj odbirka po x-osi

**plot(x,y)**




Vrijednost promjenljive x po x-osi

# Kompleksni brojevi

*<u>Primjer:</u>* Unošenje

»z=4+5*i

rezultira u

z =4.0000 + 5.0000i

dok izraz

»w=5*exp(2.5*i)

daje

»w =-4.0057 + 2.9924i

Postoje najmanje dva načina za unošenje kompleksne matrice:

-elementi se unose kao kompleksni, i

-posebno se unose realni i imaginarni dio.

# Kompleksni brojevi

<u>*Primjer:*</u> Za unošenje matrice sa kompleksnim elementima možemo ravnopravno koristiti sljedeće izraze:

»a=[-1,2;3,4],b=[5,-6;7,8],Z=a+b*i

što daje

$$a = \begin{matrix} -1 & 2 \\ 3 & 4 \end{matrix}$$

$$b = \begin{matrix} 5 & -6 \\ 7 & 8 \end{matrix}$$

Z= -1.0000 + 5.0000i 2.0000 - 6.0000i
3.0000 + 7.0000i 4.0000 +8.0000i

# Kompleksni brojevi

ili

»Z1=[-1+5*i 2-6*i;3+7*i 4+8*i]

sa istim rezultatom:

Z1 =

| | |
|---|---|
| -1.0000 + 5.0000i | 2.0000 - 6.0000i |
| 3.0000 + 7.0000i | 4.0000 + 8.0000 |

# Korijen kompleksnog broja

- Operator **sqrt**(*X*) daje kvadratni korijen elemenata matrice *X*, pri čemu se kompleksni rezultat dobije za negativne elemente, po definiciji

$$\sqrt{z} = \sqrt{r}\left(\cos(\frac{\alpha}{2} + k\pi) + i\, sin(\frac{\alpha}{2} + k\pi)\right), \qquad k = 0,1$$

- Proizvoljan korijen kompleksnog broja računa se prema formuli:

$$\sqrt[n]{z} = \sqrt[n]{r}\left(\cos(\frac{\alpha}{2} + k\pi) + i\, sin(\frac{\alpha}{2} + k\pi)\right),\ k = 0,1...\text{n-1}$$

- r=abs(z),
- α=angle(z)

# Suma i proizvod elemenata matrice

- Za vektor *x*, izraz *y*=**sum**(*x*) daje skalar y koji predstavlja zbir elemenata *x*-a. Za matricu *X*, izraz *s*=**sum**(*X*) daje vektor vrstu *s* koji sadrži sume elemenata pojedinih kolona matrice *X*. Na potpuno analogan način operator **prod** daje proizvod elemenata vektora ili matrice.

*Primjer:* Za vektor *x*=[1 5 0 -2 3] i matricu *A*

```
A =       -1.0000 2.0000 3.0000
           4.0000 3.1000 2.0000
           1.0000 5.0000 6.0000
```

imamo:

```
» y=sum(x)
  y =  7
» s=sum(A)
  s =  4.0000 10.1000 11.0000
```

Očigledno je da se zbir svih elemenata matrice *A* dobija sa **sum(sum**(*A*)), tako da u našem slučaju izraz

```
» S=sum(sum(A))
```

daje

```
S = 25.1000
```

# Suma i proizvod elemenata matrice

Proizvod elemenata pojedinih kolona matrice *A* dobićemo sa

```
» p=prod(A)

p =
        -4 31 36
```

dok se proizvod svih elemenata matrice *A* dobija sa

```
» P=prod(prod(A))
P =
        -4464
```

# Suma i proizvod elemenata matrice

*<u>Primjer:</u>* Za $k$=10, izračunati sume redova $\sum_{n=1}^{k} \frac{1}{n^3}$ i $\sum_{n=1}^{k} \ln(n)/n$

```
» n=1:10;
» s=sum(1./n.^3);
» s
s =
1.1975

» S=sum(log(n)./n)
S =
2.6922
```

# Kumulativne sume i prozvodi

- Kumulativne sume, odnosno proizvodi, dobijaju se pomoću operatora **cumsum** i **cumprod**. Naime, za vektor *x* izrazi *s*=**cumsum**(*x*) i *p*=**cumprod**(*x*) daju vektore *s* i *p* iste dimenzije kao *x*, čiji su elementi definisani sa
- $S_i = \sum_{j=1}^{i} x_j$, odnosno $P_j = \prod_{j=1}^{i} x_j$
- Ako je *X* matrica, izrazi *S*=**cumsum**(*X*) i *P*=**cumprod**(*X*) daju matrice *S* i *P* istih dimenzija kao *X*, čije kolone se sastoje od kumulativnih suma odnosno proizvoda elemenata kolona matrice *X*.

# Kumulativne sume i prozvodi

- *Primjer 4.4.3* Za vektor *x*=[1 2 3 4 5 6 7] imamo

  » s=cumsum(x)

  s =

  1 3 6 10 15 21 28

  » p=cumprod(x)  =

  Columns 1 through 6

  1 2 6 24 120 720

  Column 7

  5040
- Očigledno je da vektor *p* sadrži faktorijele brojeva od 1 do 7.

# Primjena sum i cumsum za računanje integrala

- Za funkciju $f(x)$ određeni integral može se, po pravougaonom pravilu, približno izraziti kao

$$I = \int_a^b f(x)dx \approx \sum_{i=1}^{n-1} f(x_i)\Delta x_i$$

- pri čemu je interval $(a,b)$ podijeljen na $n$ jednakih podintervala.

# Primjena sum i cumsum za računanje integrala

- Na osnovu ovih izraza očigledno je da se približno integraljenje može vršiti pomoću operatora **sum**.
- Za date granice $a$ i $b$, prvo ćemo usvojiti broj sektora $n$ i definisati korak $k=(b-a)/n$. Nezavisno promjenljiva $x$ definiše se u opštem slučaju kao $x=a:k:b-k$, pa se za datu funkciju $y=f(x)$ odre|eni integral jednostavno dobija kao $I$=**sum**($y$)*$k$.

# Primjena sum i cumsum za računanje integrala

- Primjer: Za funkciju y=sin x+4 naći integral od -2 do 2. Uzećemo n=40, pa za tu vrijednost n imamo:

```
» k=(2-(-2))/40;
» x=-2:k:2-k;
» y=sin(x)+4;
» I=sum(y)*k
I =
     15.9091
```

# Razlika elemenata i približno diferenciranje funkcija

- Za $x$=[$x$1 $x$2 $x$3 .... $xn$] izraz $y$=**diff**($x$) daje vektor $y$ čiji su elementi definisani sa $y^i = x^{i+1} - x^i$, $i$=1,2,..,$n$-1. Znači, elementi vektora $y$ predstavljaju razlike susjednih elemenata vektora $x$. Vektor $y$ je za jedan element "kraći" od vektora $x$, tj. **length**($y$)=**length**($x$)-1. Ovo važi bez obzira da li se radi o vektoru vrsti ili koloni. Ako je $X$ matrica, opercija $Y$=**diff**($X$) daje matricu $Y$ čije kolone se sastoje od razlika susjednih elemenata kolona matrice $X$, tako da $Y$ ima jednu vrstu manje od matrice $X$. Navedena operacija može se ponoviti $m$ puta, dodavanjem opcionog ulaznog argumenta. U ovakvoj verziji, operator **diff** ima oblik **diff**($X$,$m$).

# Razlika elemenata i približno diferenciranje funkcija

- Primjer: Za vektor

  x =

  1 8 27 64 125

  izrazi

  » y1=diff(x)

  » y2=diff(x,2)

  » y3=diff(x,3)

  daju

  y1 = 7 19 37 61

  y2 = 12 18 24

  y3 = 6 6

Očigledno je da važi: *y*2=**diff**(*y*1), *y*3=**diff**(*y*2), itd.

# Simbolički račun

- Šta je simbolički račun?
- Simbolički račun ili algebarski račun je naučna oblast koja se bavi proučavanjem i razvojem algoritama i softvera za manipulisanje matematičkim izrazima.
- Za razliku od numeričkog računa, u kome figurišu konkretne numeričke vrjednosti promjenljivih, u simboličkom računu se manipuliše izrazima u kojima se promjenljive tretiraju kao simboli bez konkretne vrijednosti.
- Simbolički račun odgovara rješavanju izraza „na papiru“, npr.

$$(a+b)2 – 3b2 = a2 + 2ab + b2 – 3b2 = a2 + 2ab – 2b2.$$

- Prednost simboličkog računa u odnosu na numerički je da se dobija tačna vrijednost izraza, dok je numerički podložan greškama zaokruživanja usljed ograničenja broja decimalnih mjesta kod decimalnih brojeva.

# Simbolički račun

- U MATLAB-u, simbolički račun obezbjeđuje Symbolic toolbox, koji omogućava:
  - Unos izraza u simboličkom obliku sa simboličkim tipovima podataka
  - Razvoj i pojednostavljenje simboličkih izraza
  - Određivanje simboličkih korena, graničnih vrednosti, minimuma, maksimuma funkcija itd.
  - Određivanje izvoda i integrala funkcija
  - Razvoj funkcija u Taylor-ov red
  - Rješavanje algebarskih i diferencijalnih jednačina
  - Rešavanja sistema jednačina
  - Grafički prikaz simboličkih funkcija

# Simbolički račun

- Simboličke promjenljive se kreiraju koristeći funkciju sym:

```
>> x = sym('x');
```

  ili pomoću naredbe syms (skraćeni oblik funkcije sym):

```
>> syms x;
```

- Više simboličkih promjenljivih se deklariše na sljedeći način:

```
>> syms x y z;
```

- Kreiranje izraza sa postojećom simboličkom promjenljivom:

```
>> y = (x+1)^2;
>> z = exp(2*x);
```

# Simbolički račun

- Funkcija expand razvija simbolički izraz u obliku zbira proizvoda.
- Najviše se koristi kod polinoma, ali može i kod trigonometrijskih, eksponencijalnih i logaritamskih funkcija.

```
>> syms x y
>> expand(4*(x+1)^2)
4*x^2 + 8*x + 4
>> expand(sin(x+y))
cos(x)*sin(y) + cos(y)*sin(x)
>> expand(exp(2*x-3*y))
exp(2*x)*exp(-3*y
```

# Simbolički račun

- Faktorizacija predstavlja razlaganje nekog objekta (npr. broja, polinoma ili matrice) u obliku proizvoda nekih drugih objekata, tzv. faktora. Na primer, faktori polinoma x^2+3x+2 su x+1 i x+2, jer važi x^2+3x+2 = (x+1)(x+2).
- Funkcija factor faktoriše simbolički izraz.
- Najviše se koristi kod polinoma, ali može i kod trigonometrijskih, eksponencijalnih i logaritamskih funkcija.

```
>> factor(x^3+2*x^2+x)
ans = [ x, x + 1, x + 1]
>> factor(x^2 * y^2, x)
ans = [y^2, x, x]
```

# Simbolički račun

- Funkcija simplify pojednostavljuje simbolički izraz.

```
>> simplify((x+2)^2-4*x)
ans = x^2 + 4
>> simplify(sin(x)^2 + cos(x)^2)
ans = 1
>> simplify(exp(log(3*x)))
ans = 3*x
```

# Simbolički račun

- Funkcija solve rešava jednačinu ili sistem jednačina.

```
>> y = x^2 - 9
>> solve(y)
ans = -3 3
>> syms a b c
>> solve(a*x^2+b*x+c)
ans =
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)
-(b - (b^2 - 4*a*c)^(1/2))/(2*a)
>> solve(a*x^2+b*x+c,a)
ans =
-(c + b*x)/x^2
```

# Simbolički račun

- Kod rješavanja sistema jednačina, prvo se jednačine zadaju kao simbolički izrazi, a zatim se te jednačine proslijede funkciji solve.
- Pri rješavanju sistema, izrazi se izjednačavaju sa 0.

```
>> prva = 3*x+2*y-z-10;
>> druga = -x+3*y+2*z-5;
>> treca = x-y-z+1;
>> [x0,y0,z0] = solve(prva, druga, treca)
x0 = -2
y0 = 5
z0 = -6
```

# Simbolički račun

- Funkcija subs mijenja simbol u izrazu nekim drugim simbolom ili dodjeljuje vrijednost promjenljivoj.
- Pri rješavanju sistema, izrazi se izjednačavaju sa 0.

```
>> izraz = x^2 + 3*x*y - 2*y^2;
>> subs(izraz, y, x)
ans = 2*x^2
>> subs(izraz, y, 1)
ans = x^2 + 3*x – 2
>> subs(izraz, {x,y}, {1,2})
ans = –1
```

# Simbolički račun

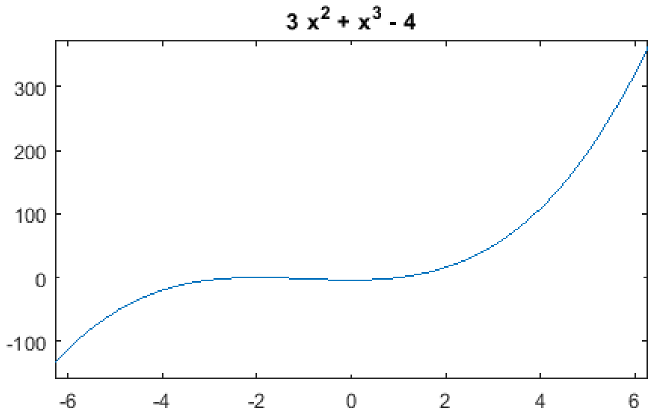
- Za traženje integrala funkcije, koristi se funkcija int.
- Može se računati neodređeni (ne navode se granice) i određeni integral (navode se granice).

```
>> izraz = x^3 + 3*x^2 - 4;
>> int(izraz)
ans = (x*(x^3 + 4*x^2 - 16))/4
>> int(izraz,0,3)                       <- Određeni integral
ans = 141/4
>> izraz = x^3 + 3*sin(y) - x*y;
>> int(izraz, y);                       <- Integral po promenljivoj
ans = y*x^3 - (y^2*x)/2 - 3*cos(y)
```

# Simbolički račun

- Za crtanje simboličkih funkcija, koristi se funkcija ezplot.
- Ukoliko se ne zada interval, podrazumijevani je [-2π, 2π].

```
>> fun = x^3 + 3*x^2 - 4;
>> ezplot(fun)
```




```
>> ezplot(fun,0,10)
```