

# PRIMJENA INŽINJESKIH SOFTVERSKIH PAKETA

## domaća zadaća 2

### Mahir Suljić

#### Zadatak 1

Napisati MATLAB funkciju `z1.m` čiji je ulaz znak `a`, `b` ili `c`, a koja ne vraća ništa.

✓ Ako je ulaz u funkciju znak `a` funkcija simbolički računa:

a) sistem algebarskih jednačina

$$x^2 + y^2 - 5 = 0 \text{ i } -3x + 2y + 1 = 0$$

b) diferencijalnu jednačinu

$$3y''(x) + 5y(x) = \cos(2x) \text{ za } y(0) = 1 \text{ i } y'(0) = 0$$

Funkcija ispisuje realna rješenja sistema algebarskih jednačina, a zatim grafički prikazuje rješenja pod a) i b) u istom grafičkom prozoru sa dva axisa jedan pored drugog (u lijevom axisu rješenja sistema jednačina označava crnim kvadratićima).

✓ Ako je ulaz u funkciju znak `b` funkcija simbolički računa i ispisuje nulu, asimptote i ekstrem funkcije  $f(x) = \frac{2+\ln x}{\sqrt{3x}}$  ( $x \in [-1, 8]$  i  $y \in [-4, 3]$ ) i grafički prikazuje funkciju i izračunata rješenja: nulu funkcije - plavim kvadratom, asimptote - crvenom linijom i ekstrem funkcije - crnim krugom.

✓ Ako je ulaz u funkciju znak `c` funkcija vrši simboličku analizu električnog kola koristeći `amatrixa.m` i `zmatrixa.m`, te ispisuje vrijednost napona na krajevima otpornika `R3` i vrijednost struje kroz naponski izvor `V1`.

#### Rješenje:

##### `z1.m`

```
function z1(symbol)
    if nargin ~= 1
        error("Funkcija ima jedan ulazni argument");
    else
        if symbol == 'a'
            z1_a;
            return;
        elseif symbol == 'b'
            z1_b;
            return;
        elseif symbol == 'c'
            z1_c;
            return;
        end
        error("Pogrešan ulazni argument")
    end
```

##### `z1_a.m`

```
syms x y real;

alg_eq1 = x^2+y^2-5==0;
alg_eq2 = -3*x+2*y+1==0;

[sx, sy] = solve(alg_eq1, alg_eq2);

dif_eq = '3*D2y + 5*y = cos(2*x)';
```

```

dif_conds = 'y(0)=1, Dy(0)=0';

dif_y = dsolve(dif_eq, dif_conds, 'x');

disp('x: ');
pretty(sx);
disp('y: ');
pretty(sy);

subplot(1, 2, 1);

ezplot(alg_eq1);
hold on;
grid on;
ezplot(alg_eq2);
hold on;
plot(sx, sy, 'ks');
title('Algebarske');

subplot(1, 2, 2);

fplot(dif_y);
title('Diferencijalna');

```

### **z1\_b.m**

```

syms x real;

y = (2+log(x))/(sqrt(3*x));

% izvodi
y_dif = diff(y);
y_dif2 = diff(y_dif);

% nule funkcije
zero_x = solve(y);
zero_y = subs(y, {x}, zero_x);
nula = zero_y;

% ekstremi
extr_x = solve(y_dif);
extr_y = subs(y, {x}, extr_x);
conv = subs(y_dif2, {x}, extr_x);

a = find(conv<0);
b = find(conv>0);

max = [extr_x(a)', extr_y(a)'];
min = [extr_x(b)', extr_y(b)'];

T = extr_x;

% horizontalna asimptota
ha = limit(y, inf);

% vertikalna asimptota
va = solve(sqrt(3*x));
limit(y, x, va, 'right');

% iscrtavanje

ezplot(y, [-1 8, -4 3]);

```

```

hold on;
grid on;
plot(zero_x, zero_y, 'bs');
plot(extr_x, extr_y, 'ko');
plot([-1 8], [ha ha], 'r');
plot([va va], [-4 3], 'r');

```

nula, ha, va, T

### z1\_c.m

```
syms R1 R2 R3 R4 R5 R6 R7 V1 V2 Is real;
```

```

R = [R1 4 5;
      R2 2 3;
      R3 3 4;
      R4 0 3;
      R5 0 4;
      R6 4 5;
      R7 0 1];
V = [V1 3 4;
      V2 2 0];
I = [Is 1 5];

```

```

kolo.V = V;
kolo.R = R;
kolo.I = I;
kolo.simb = 1;

```

```

[G, B, C, D, A] = amatrica(kolo);
Z = zmatrica(kolo);
x = (A\Z);

```

```

disp('Napon na R3 = ');
pretty(x(4)-x(3));
disp('Struja kroz V1 = ');
pretty(x(6));

```

### Primjer:

```

>> z1
Error using z1
Funkcija ima jedan ulazni argument

```

```

>> z1('x')
Error using z1
Pogrešan ulazni argument

```

```
>> z1('a')
```

```

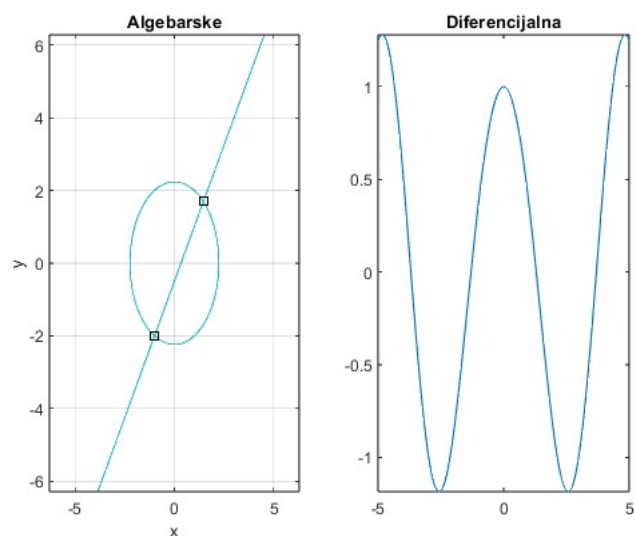
x:
/ -1 \
|     |
| 19  |
| --  |
\ 13 /

```

```

y:
/ -2 \
|     |
| 22  |
| --  |
\ 13 /

```



```
>> z1('b')
```

```
nula =
```

```
0
```

```
ha =
```

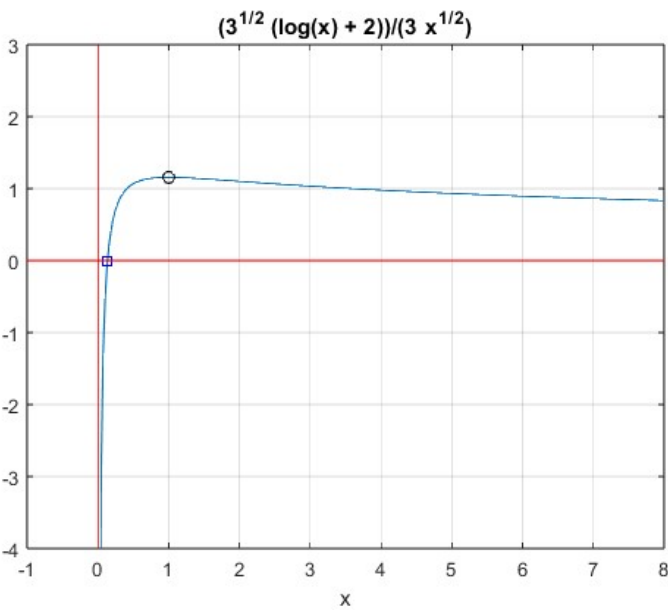
```
0
```

```
va =
```

```
0
```

```
T =
```

```
1
```



```
>> z1('c')
```

Napon na R3 =

$$\frac{R2\ R4\ V1 + R4\ R5\ V2 + Is\ R2\ R4\ R5}{R2\ R4 + R2\ R5 + R4\ R5} - \frac{R5\ (R2\ V1 + R4\ V1 - R4\ V2 - Is\ R2\ R4)}{R2\ R4 + R2\ R5 + R4\ R5}$$

Struja kroz V1 =

$$\frac{R2\ R3\ V1 + R2\ R4\ V1 + R2\ R5\ V1 + R3\ R4\ V1 - R3\ R4\ V2 + R4\ R5\ V1 + Is\ R2\ R3\ R5 + Is\ R3\ R4\ R5}{R3\ (R2\ R4 + R2\ R5 + R4\ R5)}$$

## Zadatak 2

Napisati MATLAB funkciju `z2.m` kod koje je ulaz struktura koja sadrži proizvoljan broj tačaka (minimalno 5 tačaka) opisanih sa oznakom i (x,y) koordinatama u 2D koordinatnom sistemu elemenata. Funkcija ispisuje oznake tačaka koje se nalaze na rubu minimalnog kvadrata koji sadrži sve tačke ulazne strukture (kvadrat sadrži tačku ako se tačka nalazi unutar ili na rubu kvadrata), a vraća obim i površinu kvadrata. Funkcija crta tačke ulazne strukture crnim krugom, a kvadrat crta crvenom linijom.

### Rješenje:

#### **z2.m**

```
function [O, P]=z2(T)
    if nargin ~= 1
        error('Funkcija ima jedan ulazni argument');
    end

    fn = fieldnames(T);

    if numel(fn) < 5
        error('Struktura treba imati minimalno 5 elemenata');
    end

    min_x = T.(fn{1})(1);
    min_y = T.(fn{1})(2);
    max_x = T.(fn{1})(1);
    max_y = T.(fn{1})(2);

    edge_points = [fn{1} fn{1} fn{1} fn{1}];

    for i = 1:numel(fn)
        x = T.(fn{i})(1);
        y = T.(fn{i})(2);

        if x > max_x
            max_x = x;
            edge_points(1) = fn{i};
        end
        if y > max_y
            max_y = y;
            edge_points(2) = fn{i};
        end
        if x < min_x
            min_x = x;
            edge_points(3) = fn{i};
        end
        if y < min_y
            min_y = y;
            edge_points(4) = fn{i};
        end

        plot(x, y, 'ko');
        hold on;
    end

    edge_points = unique(edge_points);
    O = (max_x - min_x + max_y - min_y) * 2;
    P = (max_x - min_x) * (max_y - min_y);

    sq_x = [min_x, max_x, max_x, min_x, min_x];
```

```
sq_y = [min_y, min_y, max_y, max_y, min_y];

grid on;
plot(sq_x, sq_y, 'r');
xlim([min_x-1 max_x+1]);
ylim([min_y-1 max_y+1]);

disp(['Tačke na rubu kvadrata: ' edge_points]);
```

### ***Primjer:***

```
>> [O P]=z2
Error using z2
Funkcija ima jedan ulazni argument

>> T = struct('A', [4 1], 'B', [3 2], 'C', [2 3], 'D', [1 2]);
>> [O P]=z2(T)
Error using z2
Struktura treba imati minimalno 5 elemenata

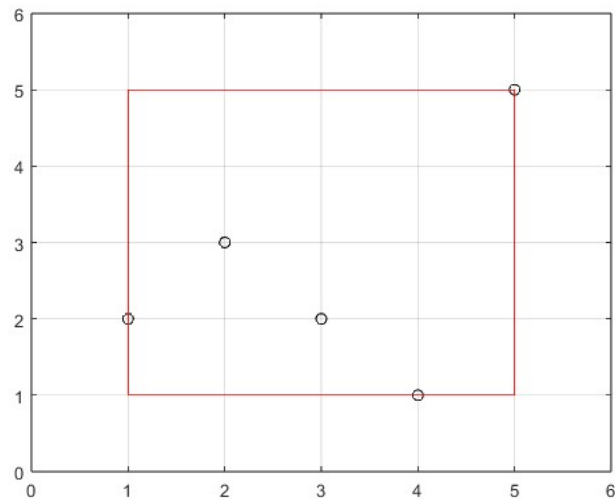
>> T = struct('A', [4 1], 'B', [3 2], 'C', [2 3], 'D', [1 2], 'E', [5 5]);
>> [O P]=z2(T)
Tačke na rubu kvadrata: ADE

O =

    16

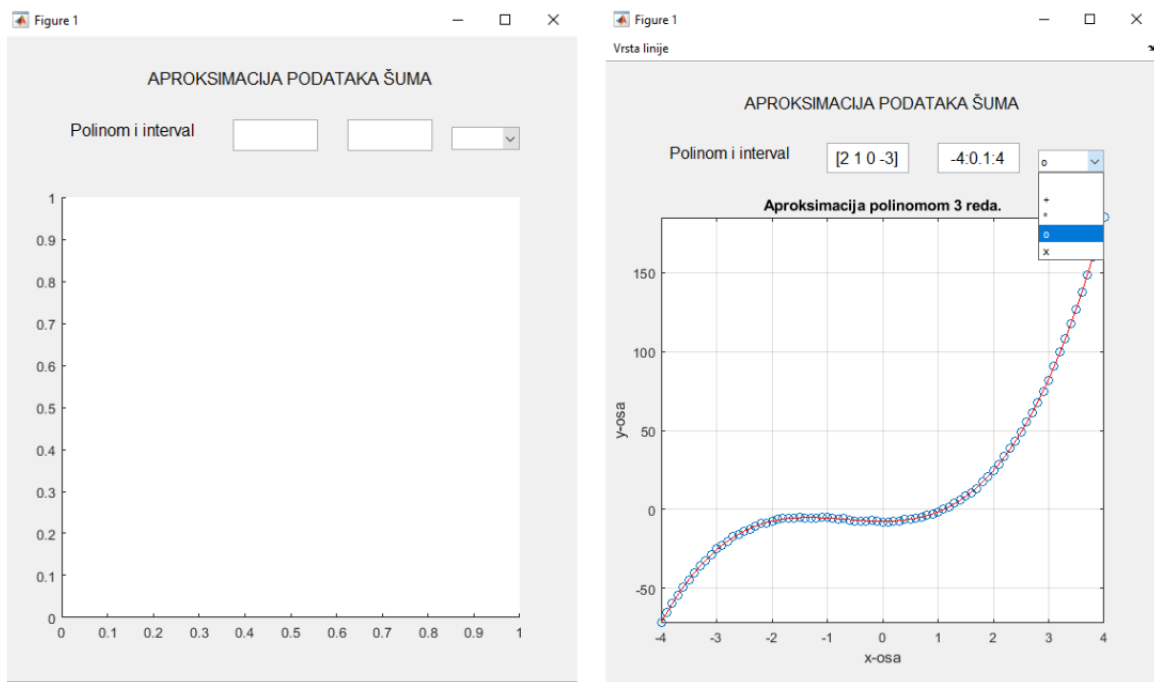
P =

    16
```



### Zadatak 3

Napisati program koji kreira grafički prozor sa svim potrebnim kontrolama sa slike.



Korisnik unosi polinom koji se sabira sa polinomom  $3x^2 - 5$ . Dobijeni zbir se evaluira za uneseni interval, zatim se rezultatima dodaje proizvoljan šum, te iscrtava signal šuma sa markerom (+, \*, o ili x) kojeg bira korisnik iz menija. Program aproksimira podatke o šumu polinomom čiji je red isti kao red izračunatog zbira polinoma, a zatim ga iscrtava na istom grafikonu koristeći isti interval. te ispisuje u naslovu red polinoma kojim se vrši aproksimacija šuma. Program kreira klasični padajući meni sa tri opcije (isprekidana, tačkasta ili puna) za vrstu linije. Ukoliko korisnik bira prvu opciju iz menija, program briše naslov i nacrtani grafik.

#### Rješenje:

**z3.m**

```
fig = figure('Position', [200 200 500 600], 'MenuBar', 'none');

lb_title = uicontrol('Style', 'text');           % title label
lb_desc = uicontrol('Style', 'text');           % description label
tb_poly = uicontrol('Style', 'edit');           % polynome textbox
tb_interval = uicontrol('Style', 'edit');       % interval textbox
cb_marker = uicontrol('Style', 'popup');        % combobox for markers
img_plot = axes;                               % image of plot
menu_line = uimenu;                            % line menu

% grafik
set(img_plot, 'Position', [0.1 0.1 0.85 0.7]);
set(img_plot, 'Tag', 'plot');
xlabel(img_plot, 'x-osa');
ylabel(img_plot, 'y-osa');
title(img_plot, 'Aproksimacija polinomom');

% naslov
set(lb_title, 'String', 'APROKSIMACIJA POKATAKA ŠUMA');
set(lb_title, 'Units', 'normalized');
set(lb_title, 'Position', [0 0.875 1 0.1]);
set(lb_title, 'FontSize', 12);
```

```

% label kontrola
set(lb_desc, 'String', 'Polinom i interval');
set(lb_desc, 'Units', 'normalized');
set(lb_desc, 'Position', [0 0.875 0.3 0.04]);
set(lb_desc, 'FontSize', 10);

% textbox za vektor
set(tb_poly, 'Units', 'normalized');
set(tb_poly, 'Position', [0.3 0.885 0.2 0.04]);

% textbox za interval
set(tb_interval, 'Units', 'normalized');
set(tb_interval, 'Position', [0.525 0.885 0.2 0.04]);

% dropdown meni za oznaku
set(cb_marker, 'Units', 'normalized');
set(cb_marker, 'String', ' |+|*|o|x');
set(cb_marker, 'Position', [0.75 0.885 0.2 0.04]);
set(cb_marker, 'Callback', 'z3_draw');

% meni za liniju
set(menu_line, 'Label', 'Vrsta linije');
func = ['set(findobj(''Tag'', ''line''), ''LineStyle'', '--')'];
uimenu(menu_line, 'Label', 'isprekidana', 'Callback', func);
func = ['set(findobj(''Tag'', ''line''), ''LineStyle'', ':')'];
uimenu(menu_line, 'Label', 'tačkasta', 'Callback', func);
func = ['set(findobj(''Tag'', ''line''), ''LineStyle'', '-')'];
uimenu(menu_line, 'Label', 'puna', 'Callback', func, 'separator', 'on');

z3_draw.m

previous_LineStyle = '-';
current_line = findobj('Tag', 'line');

if ishandle(current_line)
    previous_LineStyle = get(current_line, 'LineStyle');
end

cla;
grid off;

poly = eval(get(tb_poly, 'string'));
interval = eval(get(tb_interval, 'string'));
choice = get(cb_marker, 'value');

% priprema za sabiranje polinoma (u slucaju razlicitih stepena)
n = length(poly);
m = 0;
if n < 3
    poly = [zeros(1, 3 - n) poly];
elseif n > 3
    m = n - 3;
end

poly = poly + [zeros(1, m) 3 0 -5];
x = interval;
y = polyval(poly, x);
noise = y + randn(size(y));

if choice >= 2 && choice <= 5
    current_line = plot(x, y, 'Tag', 'line', 'LineStyle', previous_LineStyle);
    grid on;

```



```

xlabel(img_plot, 'x-osa');
ylabel(img_plot, 'y-osa');
title(img_plot, ['Aproksimacija polinomom ' num2str(length(poly)-1) ' . reda'
]);
end

switch choice
case 2
    hold on;
    plot(x, noise, '+');
case 3
    hold on;
    plot(x, noise, '*');
case 4
    hold on;
    plot(x, noise, 'o');
case 5
    hold on;
    plot(x, noise, 'x');
end

```

