

# Seminář 10

## Test polohy bodu

Test polohy vůči přímce  $F(x,y): ax + by + c = 0$

$F(p_x, p_y) > 0$  -- uvnitř

$F(p_x, p_y) = 0$  -- na přímce

$F(p_x, p_y) < 0$  -- vně

```
A = [-10,10];  
B = [10,-10];  
  
x = [A(1);B(1)];  
y = [A(2);B(2)];  
  
figure,  
primka = line(x,y);  
hold on
```

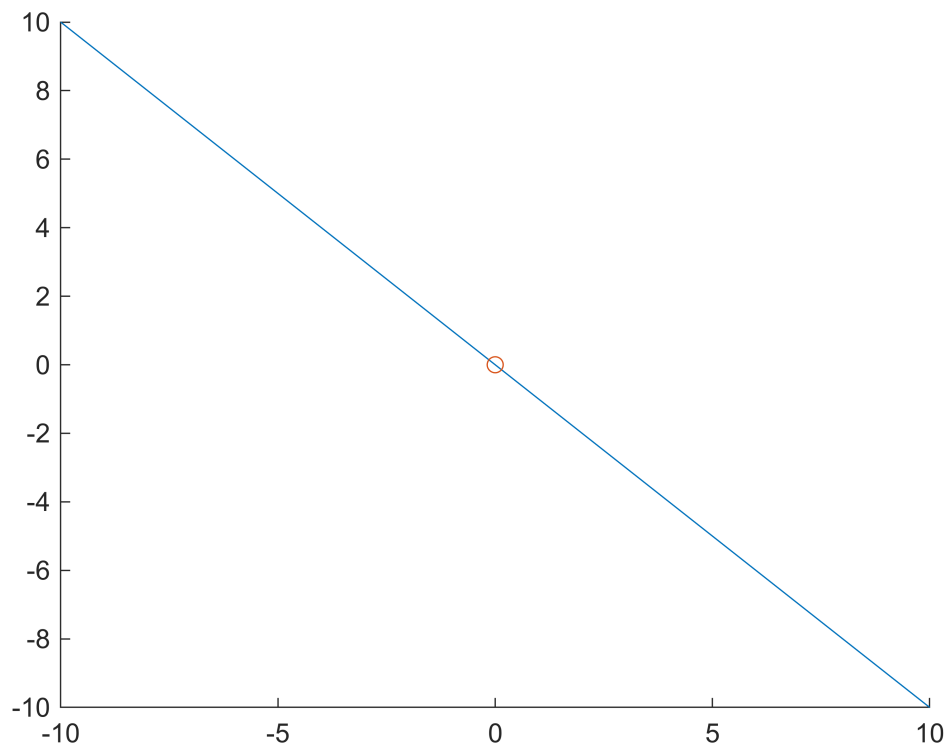
Rovnice přímky

```
% y = ax + b  
[a, b] = rovnice_primky(A, B);  
fprintf("y = %f x + %f \n",a,b);
```

```
y = -1.000000 x + 0.000000
```

Vyhodnocení polohy bodu

```
bod_x = 0;  
bod_y = 0;  
  
plot(bod_x,bod_y,'o');  
hold off
```



```
poloha = -bod_y + a*bod_x + b
```

```
poloha = 0
```

## Ořezání úsečky

Ořezáváme obdélníkovým oknem, které je rovnoběžné s osami

```
X = [0 100];
Y = [0 50];

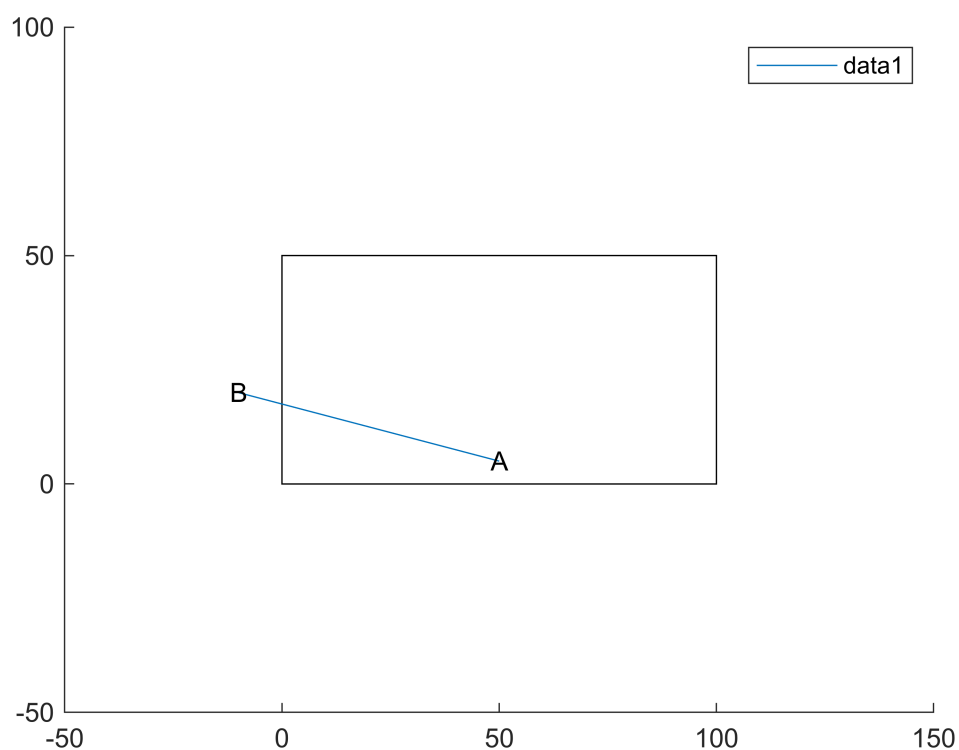
figure,
rectangle('Position',[X(1),Y(1),X(2)-X(1), Y(2)-Y(1)]);
axis([-50 150 -50 100])
hold on

A_x = 50;
A_y = 5;

B_x = -10;
B_y = 20;

line([A_x,B_x], [A_y,B_y]);
text(A_x, A_y, 'A', 'HorizontalAlignment','center')
text(B_x, B_y, 'B', 'HorizontalAlignment','center')
```

```
legend
hold off
```



## Kód bodu

```
kod_A = kod_bodu([A_x A_y], X, Y)
```

```
kod_A = 1×4
      0      0      0      0
```

```
kod_B = kod_bodu([B_x B_y], X, Y)
```

```
kod_B = 1×4
      1      0      0      0
```

## Úkol 1

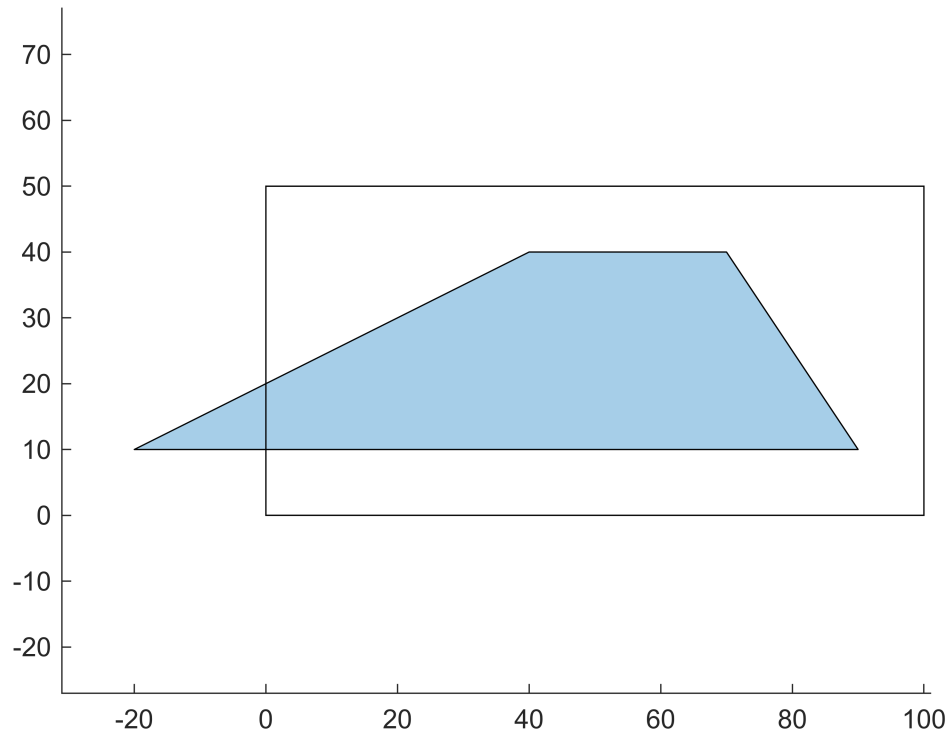
*Jak bychom spočítali průsečík přímky a kružnice?*

## Ořezání polygonu

```
polX = [-20 40 70 90];
polY = [10 40 40 10];
orezX = [X(1) X(2) X(2) X(1)];
orezY = [Y(1) Y(1) Y(2) Y(2)];

pol = polyshape(polX, polY);
```

```
figure,
plot(pol)
hold on
rectangle('Position',[X(1),Y(1),X(2)-X(1), Y(2)-Y(1)]);
axis equal
hold off
```

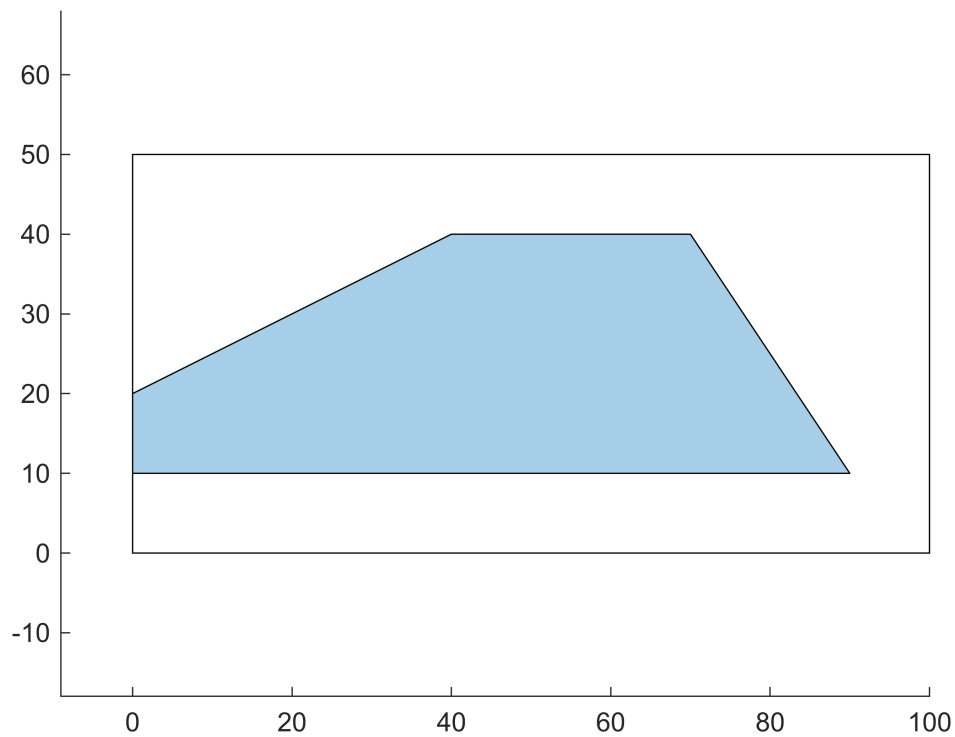


## Sutherland-Hodgmanův algoritmus

```
orezany = sutherlandHodgman([polX' polY'], [orezX' orezY'])
```

```
orezany = 5x2
    0    10.0000
    0    20.0000
  40.0000  40.0000
  70.0000  40.0000
  90.0000  10.0000
```

```
pol = polyshape(orezany(:,1)', orezany(:,2)');
figure,
plot(pol)
hold on
rectangle('Position',[X(1),Y(1),X(2)-X(1), Y(2)-Y(1)]);
axis equal
hold off
```



## Algoritmus Weiler-Atherton

příklad ze slide 12

```

okno = {1 1 'a';
        1 3 'g';
        3 5 'f';
        6 5 'e';
        6 3 'd';
        4 3 'c';
        4 1 'b'};

pol = {5 0 'A';
       2 0 'J';
       2 2 'I';
       0 2 'H';
       0 3 'G';
       2 3 'F';
       2 6 'E';
       7 6 'D';
       7 4 'C';
       5 4 'B'};

pol_okno = polyshape([okno{:,1}]', [okno{:,2}]]', 'SolidBoundaryOrientation', 'cw');
pol_polygon = polyshape([pol{:,1}], [pol{:,2}]);

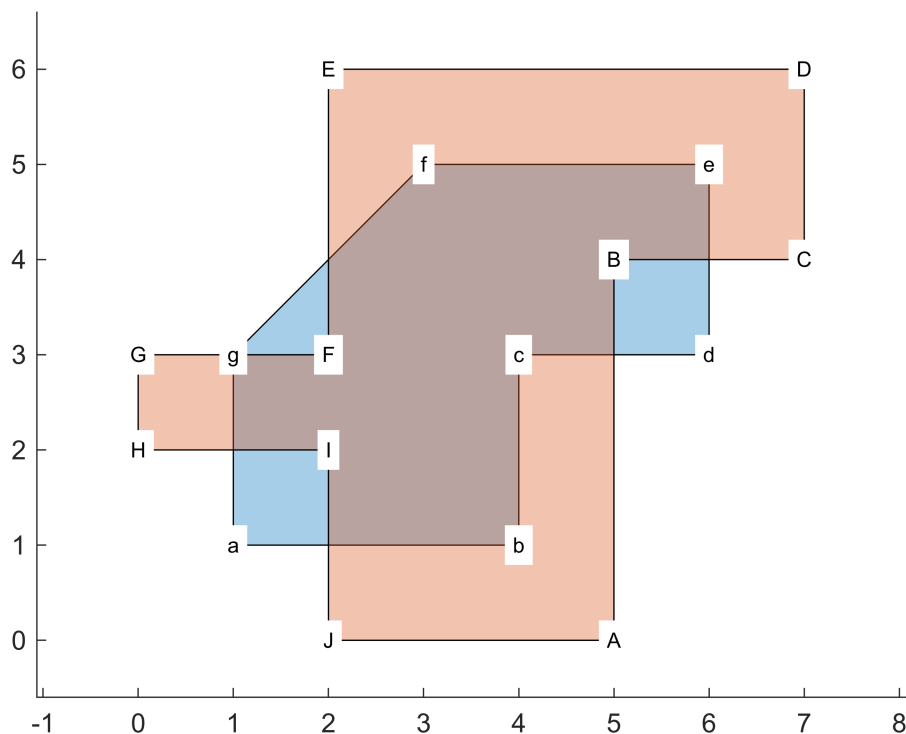
```

```

figure,
plot(pol_okno)
hold on
plot(pol_polygon)
axis equal

for i = 1 : size(okno,1)
    text(okno{i,1}, okno{i,2}, okno{i,3}, 'HorizontalAlignment','center',
'FontSize',8,'BackgroundColor',[1,1,1])
end
for i = 1 : size(pol,1)
    text(pol{i,1}, pol{i,2}, pol{i,3}, 'HorizontalAlignment','center',
'FontSize',8, 'BackgroundColor',[1,1,1])
end
hold off

```



ShapeID představuje index kterému polygonu patří výsledný vrchol (0 znamená, že se jedná o průsečík)

VertexId index vrcholu v polygonech (0 znamená, že se jedná o nový vrchol)

```
[polyout,shapeID,vertexID] = intersect(pol_okno,pol_polygon);
```

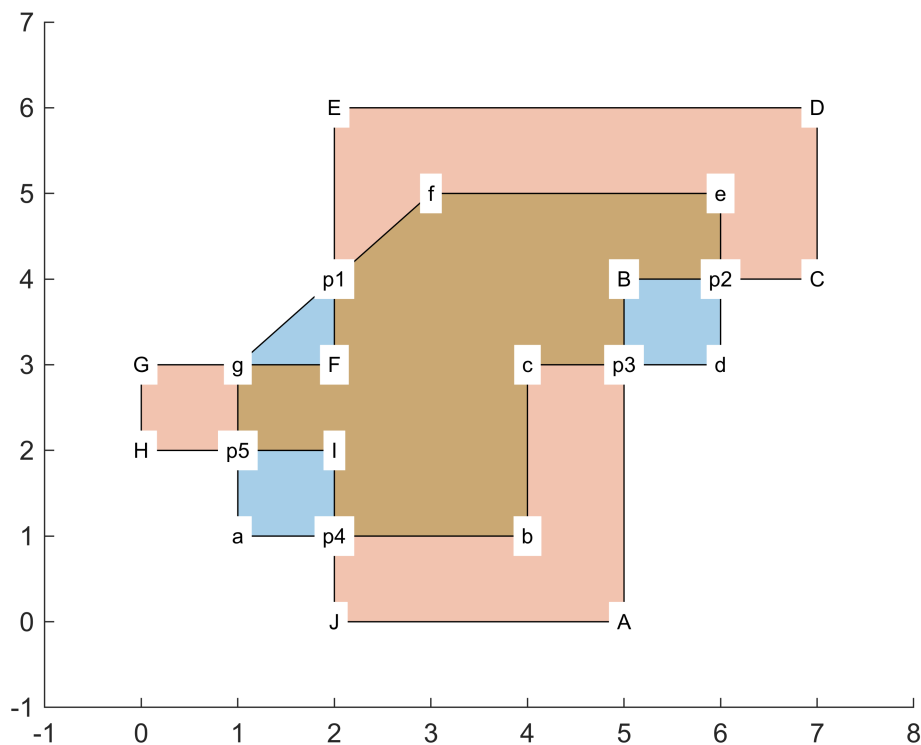
```
pruseciky = polyout.Vertices(shapeID==0,:);
pruseciky_cell = cell(size(pruseciky,1),3);
```

```

for i = 1 : size(pruseciky,1)
    pruseciky_cell{i,1} = pruseciky(i,1);
    pruseciky_cell{i,2} = pruseciky(i,2);
    pruseciky_cell{i,3} = ['p' num2str(i)];
end

figure,
plot(pol_okno)
hold on
plot(pol_polygon)
plot(polyout)
for i = 1 : size(okno,1)
    text(okno{i,1}, okno{i,2}, okno{i,3}, 'HorizontalAlignment','center',
'FontSize',8, 'BackgroundColor',[1,1,1])
end
for i = 1 : size(pol,1)
    text(pol{i,1}, pol{i,2}, pol{i,3}, 'HorizontalAlignment','center',
'FontSize',8, 'BackgroundColor',[1,1,1])
end
for i = 1 : size(pruseciky_cell,1)
    text(pruseciky_cell{i,1}, pruseciky_cell{i,2}, pruseciky_cell{i,3},
'HorizontalAlignment','center', 'FontSize',8, 'BackgroundColor',[1,1,1])
end
hold off

```



```

index_p = 1;

```

```

for i = 1 : size(polyout.Vertices,1)
    index = vertexID(i);
    if shapeID(i) == 1
        fprintf("%c ", okno{index,3});
    elseif shapeID(i) == 2
        fprintf("%c ", pol{index,3});
    else
        fprintf("%s ", pruseciky_cell{index_p,3});
        index_p = index_p + 1;
    end
end
end

```

```

g
F
p1
f e
p2
B
p3
c b
p4
I
p5

```

## Pomocné funkce

### Rovnice přímky

```

function [a, b] = rovnice_primky(A, B)
    % vypočet smernice
    a = (B(2) - A(2))/(B(1)-A(1));

    % dosazení A a vypočet b
    b = A(2) - a * A(1);
end

function kod = kod_bodu(bod,okno_x, okno_y)
% vlevo, vpravo, dole, nahore
    kod = zeros(1,4);
    if bod(1) < okno_x(1)
        kod(1) = 1;
    end
    if bod(1) > okno_x(2)
        kod(2) = 1;
    end
    if bod(2) < okno_y(1)
        kod(3) = 1;
    end
    if bod(2) > okno_y(2)
        kod(4) = 1;
    end
end

```



```
end
```

## Sutherland-Hodgmanův algoritmus

### Průsečík

```
function prusecik = spocitejPrusecik(primka1,primka2)
    prusecik = zeros(1,2);

    detL1 = det(primka1);
    detL2 = det(primka2);

    detL1x = det([primka1(:,1),[1;1]]);
    detL1y = det([primka1(:,2),[1;1]]);

    detL2x = det([primka2(:,1),[1;1]]);
    detL2y = det([primka2(:,2),[1;1]]);

    denominator = det([detL1x detL1y;detL2x detL2y]);

    prusecik(1) = det([detL1 detL1x;detL2 detL2x]) / denominator;
    prusecik(2) = det([detL1 detL1y;detL2 detL2y]) / denominator;

end
```

### Poloha bodu

```
function in = uvnitr(bod,hranice)
    vektorBodu = [diff([bod;hranice(1,:)]) 0];
    vektorHranice = [diff(hranice) 0];
    normala = cross(vektorBodu,vektorHranice);

    if ( normala(3) <= 0 )
        in = true;
    else
        in = false;
    end

end
```

### Ořezávací funkce

```
function orezany = sutherlandHodgman(polygon,okno)
    orezany = polygon;
    pocetVrchlu = size(okno,1);
    predchoziOrezovyVrchol = okno(end,:);

    for P = (1:pocetVrchlu)

        orezovaPrimka = [okno(P,:) ; predchoziOrezovyVrchol];

        kOrezani = orezany;
```

```

orezany = [];
if ~isempty(kOrezani)
    predchoziVrchol = kOrezani(end,:);
end

for aktVrchol = (1:size(kOrezani,1))

    if ( uvnitr(kOrezani(aktVrchol,:),orezovaPrimka) )

        if( not(uvnitr(predchoziVrchol,orezovaPrimka)) )
            primka = [predchoziVrchol;kOrezani(aktVrchol,:)];
            orezany(end+1,1:2) = spocitejPrusecik(orezovaPrimka,primka);
        end

        orezany(end+1,1:2) = kOrezani(aktVrchol,:);

        elseif( uvnitr(predchoziVrchol,orezovaPrimka) )
            primka = [predchoziVrchol;kOrezani(aktVrchol,:)];
            orezany(end+1,1:2) =
spocitejPrusecik(orezovaPrimka,primka);
        end

        predchoziVrchol = kOrezani(aktVrchol,:);
        predchoziOrezovyVrchol = okno(P,:);

    end
end
end

```