

## Metody komputerowe i obliczeniowe

### Metoda Elementów Skończonych

#### Przykład obliczeniowy z elementem czterowęzłowym Q4

##### Płaski Stan Naprężenia

Na poprzedniej lekcji rozwiązywaliśmy zadania w płaskim stanie odkształcenia wyznaczając tylko przemieszczenia układu węzłów. Teraz rozwiązując przykład zadania w płaskim stanie naprężenia zastosujemy już pełny zestaw kroków obliczeniowych procedury MES, tzn. uwzględnimy **Krok 6 – obróbka wyników (postprocessing)**, w którym wyznaczymy wartości składowych stanu naprężenia w punktach Gaussa dla każdego elementu niezależnie, wg formuły:

$$\{\sigma\}_{3 \times 1} = [D]_{3 \times 3} \cdot [B]_{3 \times 8} \cdot \{u\}_{8 \times 1}$$

(dolne indeksy oznaczają odpowiednio: liczbę wierszy x liczbę kolumn wielkości umieszczonej w nawiasie). Obliczenia realizujemy funkcją SigmaQ4PSN:

```
function sQ4 = SigmaQ4PSN(E,ni,x1,y1,x2,y2,x3,y3,x4,y4,u)
%Wyznacza składowe stanu naprezenia dla pojedynczego elementu czworokątne Q4
%na podstawie parametrow materialowych, wspolrzednych wezlow i obliczonych
%wcześniej przemieszczen wezlow w Płaskim Stanie Naprezenia
%wynik 3x4 (3 składowe w 4 punktach Gaussa)

%1. Macierz sprężystości - płaski stan naprezenia
D = Dpsn(E, ni);

%WSPOLRZEDNE PUNKTOW GAUSSA
ksi(1) = -sqrt(1.0 ./ 3.0);
eta(1) = sqrt(1.0 ./ 3.0);

ksi(2) = sqrt(1.0 ./ 3.0);
eta(2) = sqrt(1.0 ./ 3.0);

ksi(3) = sqrt(1.0 ./ 3.0);
eta(3) = -sqrt(1.0 ./ 3.0);

ksi(4) = -sqrt(1.0 ./ 3.0);
eta(4) = -sqrt(1.0 ./ 3.0);

%OBLICZENIA W PUNKTACH GAUSSA
sQ4=zeros(3,4);
for i = 1:4

    %2. Macierz Jacobiego
    Jacobi = Q4J(x1, y1, x2, y2, x3, y3, x4, y4, ksi(i), eta(i));

    %3. Jacobian - wyznacznik z macierzy Jakobiego
    detJacobi = det(Jacobi);

    %4. Odwrocona macierz Jacobiego
    invJacobi = inv(Jacobi);

    %5. Pochodne funkcji kształtu po zmiennych lokalnych
    dNdL = Q4derivNL(ksi(i), eta(i));

    %6. Pochodne funkcji kształtu po zmiennych globalnych
    dNdx = Q4derivNx(invJacobi, dNdL);
```

```

%7. Ukladanie macierzy B
B = Q4makeB(dNdx);

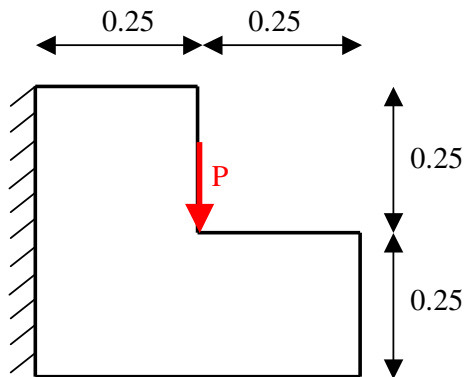
%8. Oblicza składowe napreżenia: iloczyn D*B*u
sQ4[:,i] = D * B * u;
end

```

**PROSZĘ SKOPIOWAĆ Z SERWERA KOMPLET FUNKCJI OBLICZENIOWYCH.**

### **Zadanie 1:**

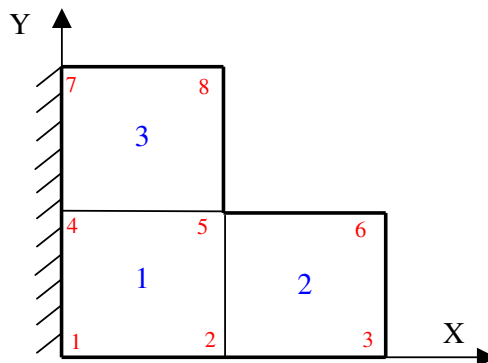
Wyznaczyć przemieszczenia i stan naprężenia w tarczy przedstawionej na rysunku. Przyjąć  $E=210\text{GPa}$ ,  $\nu = 0.3$ , grubość tarczy  $t=0.025\text{m}$  i  $P=12.5\text{kN}$ .



Rozwiązanie:

### **Krok 1 – dyskretyzacja zadania**

Zadanie dzielimy na elementy i węzły : na rysunku poniżej zaznaczono ponumerowane węzły i elementy.



### **Krok 2 – utworzenie macierzy sztywności dla każdego elementu**

Wprowadzamy zmienne globalne, które przechowują dane materiałowe i geometryczne naszego zadania:  $E$ ,  $\nu$  i  $t$  oraz współrzędne węzłów:

```

>>format short
>>E=210e6
>>ni=0.3
>>t=0.025
>>x1=0
>>y1=0

```

```
>>x2=0.25
>>y2=0
>>x3=0.5
>>y3=0
>>x4=0
>>y4=0.25
>>x5=0.25
>>y5=0.25
>>x6=0.5
>>y6=0.25
>>x7=0
>>y7=0.5
>>x8=0.25
>>y8=0.5
```

Mamy trzy elementy, zatem tworzymy trzy macierze sztywności : k1, k2 i k3 komendami:

```
>>k1=SztywnoscQ4PSN(E,ni,t,x1,y1,x4,y4,x5,y5,x2,y2)
>>k2=SztywnoscQ4PSN(E,ni,t,x2,y2,x5,y5,x6,y6,x3,y3)
>>k3=SztywnoscQ4PSN(E,ni,t,x4,y4,x7,y7,x8,y8,x5,y5)
```

Proszę zauważyć, że do wyznaczenia macierzy sztywności elementu w płaskim stanie naprężenia niezbędny jest wymiar elementu w trzecim kierunku (grubość) t. W płaskim stanie odkształcenia obliczenia domyślnie prowadzimy dla t=1m. W obu przypadkach składniki macierzy sztywności elementu są wymnażane przez t.

### **Krok 3 – składanie macierzy sztywności elementów w jedną globalną macierz dla całego układu**

Ponieważ w układzie mamy 8 węzłów, więc globalna macierz sztywności będzie miała wymiar  $2 \times 8 \times 2 \times 8 = 16 \times 16$ . Macierz K należy przed składaniem wyzerować, co wykonujemy komendą:

```
>>K=zeros(16,16)
```

Ponieważ mamy trzy elementy, to funkcję ZlozSztywnosciQ4 trzeba wywołać trzy razy – niezależnie dla każdego elementu, podając jako parametry globalną macierz K (która jest wynikiem), macierz elementu k (k1, k2, a potem k3) i numery węzłów definiujące dany element:

```
>>K=ZlozSztywnosciQ4(K,k1,1,4,5,2)
>>K=ZlozSztywnosciQ4(K,k2,2,5,6,3)
>>K=ZlozSztywnosciQ4(K,k2,4,7,8,5)
```

Na odpowiednich miejscach w macierzy K pojawiają się sumowane sztywności poszczególnych elementów (PROSZĘ SPRAWDZIĆ!).

### **Krok 4 – uwzględnienie warunków brzegowych**

Warunki brzegowe:

- przemieszczeniowe:

$u_{1x}=0; u_{1y}=0; u_{4x}=0; u_{4y}=0; u_{7x}=0; u_{7y}=0;$

- obciążeniowe:

$f_{2x}=0; f_{2y}=0; f_{3x}=0; f_{3y}=0; f_{6x}=0; f_{6y}=0; f_{8x}=0; f_{8y}=0;$

$f_{5x}=0; f_{5y}=-12.5$

Z macierzy sztywności potrzebujemy tylko te wiersze i kolumny, które odpowiadają nieznanym przemieszczeniom, zatem wycinamy następujące wiersze i kolumny: 1, 2, 7, 8, 13, 14. Resztę kopiujemy do roboczej macierzy k:

```
>>k=[K(3:6,3:6) K(3:6,9:12) K(3:6,15:16) ; K(9:12,3:6) K(9:12,9:12)
      K(9:12,15:16) ; K(15:16,3:6) K(15:16,9:12) K(15:16,15:16)]
```

Tworzymy wektor ze znanymi obciążeniami ( $f_{2x}$ ;  $f_{2y}$ ;  $f_{3x}$ ;  $f_{3y}$ ;  $f_{5x}$ ;  $f_{5y}$ ;  $f_{6x}$ ;  $f_{6y}$ ;  $f_{8x}$ ;  $f_{8y}$ );

$f_{5x}=0$ ;  $f_{5y}=-12.5$ ):

```
>>f = [0 ; 0 ; 0 ; 0 ; 0 ; 0 ; -12.5 ; 0 ; 0 ; 0 ; 0 ; 0]
```

wyznaczamy nieznanne przemieszczenia poleceniem:

```
>>u = k\f
```

otrzymując:

```
u =
1.0e-005 *
```

```
-0.1396
-0.3536
-0.1329
-0.5406
 0.0018
-0.4216
 0.0086
-0.5176
 0.1202
-0.3010
```

### Krok 6 – obróbka wyników (postprocessing)

Mając przemieszczenia wszystkich węzłów, możemy obliczyć reakcje w podporach. Najpierw zbierzmy przemieszczenia w jeden wektor:

```
>>U=[0 ; 0 ; u(1:4) ; 0 ; 0 ; u(5:8); 0 ; 0 ; u(9:10)]
```

a potem wyliczymy siły:

```
>>F=K*U
```

otrzymamy:

```
F =
```

```
 6.3994
 4.3354
 0.0000
 0.0000
 0.0000
-0.0000
-0.2988
 3.6296
-0.0000
-12.5000
 0.0000
-0.0000
-6.1006
 4.5350
 0
 0.0000
```

Składowe stanu naprężenia w elementach wyznaczymy dzięki funkcji `SigmaQ4PSN`, której parametrami są: parametry materiałowe, współrzędne węzłów oraz przemieszczenia węzłów definiujących dany element. Najpierw przygotujemy po CZTERY PARY przemieszczeń dla każdego elementu (po dwie wartości dla każdego węzła definiującego dany element):

```
>>u1=[U(1) ; U(2) ; U(7) ; U(8) ; U(9) ; U(10) ; U(3) ; U(4)]
>>u2=[U(3) ; U(4) ; U(9) ; U(10) ; U(11) ; U(12) ; U(5) ; U(6)]
>>u3=[U(7) ; U(8) ; U(13) ; U(14) ; U(15) ; U(16) ; U(9) ; U(10)]
```

a potem wyznaczmy składowe stanu naprężenia  $[\sigma_x \sigma_y \tau_{xy}]^T$ :

```
>>s1=SigmaQ4PSN(E,ni,x1,y1,x4,y4,x5,y5,x2,y2,u1)
```

```
s1 =
```

```
1.0e+003 *
```

```
-0.2990    -0.4077    -1.1614    -1.0527
-0.2104    -0.5728    -0.7989    -0.4365
-1.2192    -0.9554    -0.8286    -1.0924
```

```
>>s2=SigmaQ4PSN(E,ni,x2,y2,x5,y5,x6,y6,x3,y3,u2)
```

```
s2 =
```

```
-72.7476    72.7476    72.7476   -72.7476
-431.4958    53.4882    53.4882  -431.4958
 84.8722    84.8722   -84.8722   -84.8722
```

```
>>s3=SigmaQ4PSN(E,ni,x4,y4,x7,y7,x8,y8,x5,y5,u3)
```

```
s3 =
```

```
1.0e+003 *
```

```
0.9493    1.1421    0.5111    0.3182
0.4989    1.1416    0.9523    0.3096
-0.9740   -0.7532   -0.9782   -1.199
```

Wyniki pokazują wartości składowych stanu naprężenia w kolejnych punktach Gaussa (w każdej kolumnie są składowe dla jednego punktu), zdefiniowanych w procedurze `SigmaQ4PSN` (PROSZĘ SPRAWDZIĆ KOLEJNOŚĆ PUNKTÓW GAUSSA!).

Narysujemy teraz wykresy przemieszczeń węzłów. Najpierw zapiszemy współrzędne węzłów (po kolei, po obrysie tarczy zgodnie z ruchem wskazówek zegara, najpierw  $x$  potem  $y$ ):

```
>>wx = [x1 x4 x7 x8 x5 x6 x3 x2 x1];
```

```
>>wy = [y1 y4 y7 y8 y5 y6 y3 y2 y1];
```

Wprowadzimy do danych wyniki obliczeń z mnożnikiem 5000 dla przemieszczeń (żeby było je widać):

```
>>ux = [U(1) U(7) U(13) U(15) U(9) U(11) U(5) U(3) U(1)];
```

```
>>ux = wx + ux .* 5000
```

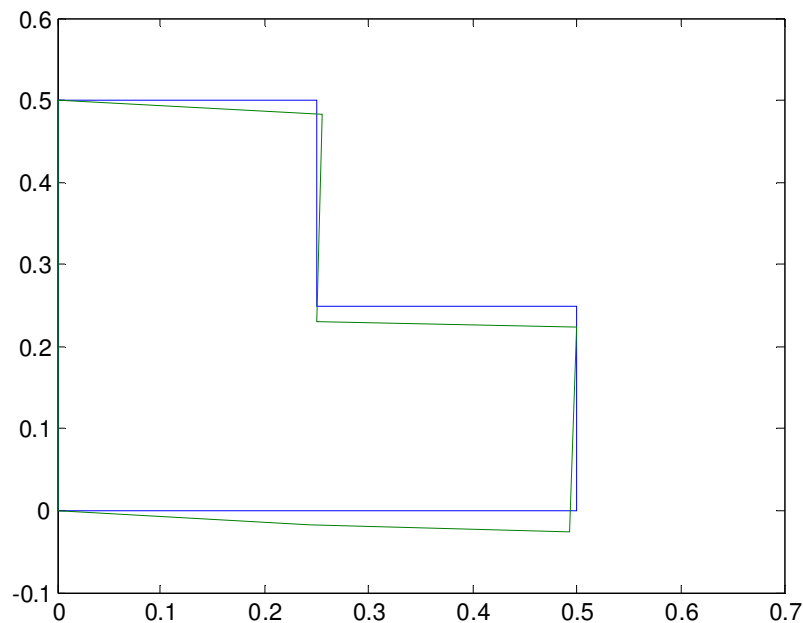
```
>>uy = [U(2) U(8) U(14) U(16) U(10) U(12) U(6) U(4) U(2)];
```

```
>>uy = wy + uy .* 5000
```

i rysujemy:

```
>>plot (wx,wy,ux,uy)
```

powinien pojawić się obrazek:



Aby wyświetlić mapy z rozkładami wartości składowych stanu naprężenia należy wyznaczyć wartości globalne współrzędnych punktów Gaussa (jak w zadaniu z punktami A, B i C, w którym należało znaleźć ich współrzędne globalne, znając wartości ich współrzędnych lokalnych, patrz „Lekcja Q4” i „Q4\_rozwiazania”). Operację wyznaczenia współrzędnych globalnych wykonamy „ręcznie”. Najpierw zadeklarujemy współrzędne lokalne punktów Gaussa (kolejność taka sama, jak użyta w obliczeniach, patrz procedura na str.1):

```
>>ksi1 = -sqrt(1.0 ./ 3.0);  
>>eta1 = sqrt(1.0 ./ 3.0);  
>>ksi2 = sqrt(1.0 ./ 3.0);  
>>eta2 = sqrt(1.0 ./ 3.0);  
>>ksi3 = sqrt(1.0 ./ 3.0);  
>>eta3 = -sqrt(1.0 ./ 3.0);  
>>ksi4 = -sqrt(1.0 ./ 3.0);  
>>eta4 = -sqrt(1.0 ./ 3.0);
```

Następnie obliczymy wartości odpowiadających im funkcji kształtu N (po cztery dla każdego punktu w elemencie):

```
>>N11 = 0.25*(1-ksi1)*(1-eta1);  
>>N12 = 0.25*(1-ksi1)*(1+eta1);  
>>N13 = 0.25*(1+ksi1)*(1+eta1);  
>>N14 = 0.25*(1+ksi1)*(1-eta1);  
  
>>N21 = 0.25*(1-ksi2)*(1-eta2);
```

```
>>N22 = 0.25*(1-ksi2)*(1+eta2);
>>N23 = 0.25*(1+ksi2)*(1+eta2);
>>N24 = 0.25*(1+ksi2)*(1-eta2);
```

```
>>N31 = 0.25*(1-ksi3)*(1-eta3);
>>N32 = 0.25*(1-ksi3)*(1+eta3);
>>N33 = 0.25*(1+ksi3)*(1+eta3);
>>N34 = 0.25*(1+ksi3)*(1-eta3);
```

```
>>N41 = 0.25*(1-ksi4)*(1-eta4);
>>N42 = 0.25*(1-ksi4)*(1+eta4);
>>N43 = 0.25*(1+ksi4)*(1+eta4);
>>N44 = 0.25*(1+ksi4)*(1-eta4);
```

W końcu możemy wyznaczyć wartości współrzędnych x i y (po cztery pary dla każdego elementu, współrzędne węzłów w kolejności definiującej element):  
dla elementu nr 1 (węzły: 1-4-5-2):

```
>>x11 = N11*x1 + N12*x4 + N13*x5 + N14*x2;
>>x12 = N21*x1 + N22*x4 + N23*x5 + N24*x2;
>>x13 = N31*x1 + N32*x4 + N33*x5 + N34*x2;
>>x14 = N41*x1 + N42*x4 + N43*x5 + N44*x2;
```

```
>>y11 = N11*y1 + N12*y4 + N13*y5 + N14*y2;
>>y12 = N21*y1 + N22*y4 + N23*y5 + N24*y2;
>>y13 = N31*y1 + N32*y4 + N33*y5 + N34*y2;
>>y14 = N41*y1 + N42*y4 + N43*y5 + N44*y2;
```

dla elementu nr 2 (węzły: 2-5-6-3):

```
>>x21 = N11*x2 + N12*x5 + N13*x6 + N14*x3;
>>x22 = N21*x2 + N22*x5 + N23*x6 + N24*x3;
>>x23 = N31*x2 + N32*x5 + N33*x6 + N34*x3;
>>x24 = N41*x2 + N42*x5 + N43*x6 + N44*x3;
```

```
>>y21 = N11*y2 + N12*y5 + N13*y6 + N14*y3;
>>y22 = N21*y2 + N22*y5 + N23*y6 + N24*y3;
>>y23 = N31*y2 + N32*y5 + N33*y6 + N34*y3;
>>y24 = N41*y2 + N42*y5 + N43*y6 + N44*y3;
```

dla elementu nr 3 (węzły: 4-7-8-5):

```
>>x31 = N11*x4 + N12*x7 + N13*x8 + N14*x5;
```

```
>>x32 = N21*x4 + N22*x7 + N23*x8 + N24*x5;
>>x33 = N31*x4 + N32*x7 + N33*x8 + N34*x5;
>>x34 = N41*x4 + N42*x7 + N43*x8 + N44*x5;

>>y31 = N11*y4 + N12*y7 + N13*y8 + N14*y5;
>>y32 = N21*y4 + N22*y7 + N23*y8 + N24*y5;
>>y33 = N31*y4 + N32*y7 + N33*y8 + N34*y5;
>>y34 = N41*y4 + N42*y7 + N43*y8 + N44*y5;
```

Uporządkujemy teraz wyliczone współrzędne, układając je w wektory  $gx$  i  $gy$ :

```
>>gx = [x11 x12 x13 x14 x21 x22 x23 x24 x31 x32 x33 x34];
>>gy = [y11 y12 y13 y14 y21 y22 y23 y24 y31 y32 y33 y34];
```

Wprowadzimy do danych wyniki obliczeń:

```
>>sx = [s1(1,1) s1(1,2) s1(1,3) s1(1,4) s2(1,1) s2(1,2) s2(1,3) s2(1,4) s3(1,1)
        s3(1,2) s3(1,3) s3(1,4)];
>>sy = [s1(2,1) s1(2,2) s1(2,3) s1(2,4) s2(2,1) s2(2,2) s2(2,3) s2(2,4) s3(2,1)
        s3(2,2) s3(2,3) s3(2,4)];
>>txy = [s1(3,1) s1(3,2) s1(3,3) s1(3,4) s2(3,1) s2(3,2) s2(3,3) s2(3,4) s3(3,1)
        s3(3,2) s3(3,3) s3(3,4)];
```

Ponieważ Matlab ma swoje „upodobania” dotyczące formatu danych przeznaczonych do graficznej prezentacji, musimy odpowiednio przygotować współrzędne punktów graficznych:  
najpierw siatka X-Y:

```
>>X = 0:0.01:0.5;
>>Y = 0:0.01:0.5;
>>[X Y] = meshgrid(X,Y);
```

potem interpolacja danych – dopasowanie wartości składowej  $\sigma_x$  naprężenia do siatki:

```
>>[XI YI Sx] = griddata(gx,gy,sx,X,Y);
```

i w końcu rysunek:

```
>>contourf(XI,YI,Sx)
```

nie zamykając okna graficznego, narysujmy kontur naszej tarczy:

```
>>line(wx,wy,'color','k','LineWidth',3)
```

zakryjmy obszar wykresu wychodzący poza tarczę (liniowa interpolacja wstawia punkty poza obrysem tarczy):

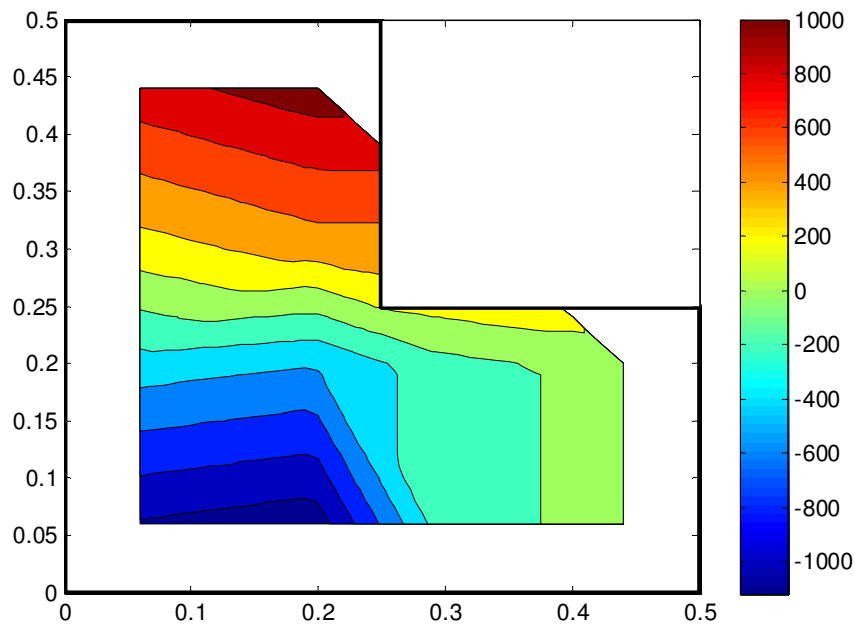
```
>>rectangle('Position',[x5 y5 0.25 0.25],'FaceColor','w')
```

i wyświetlmy skalę wartości naprężenia

```
>>colorbar
```

powinien pojawić się rysunek:



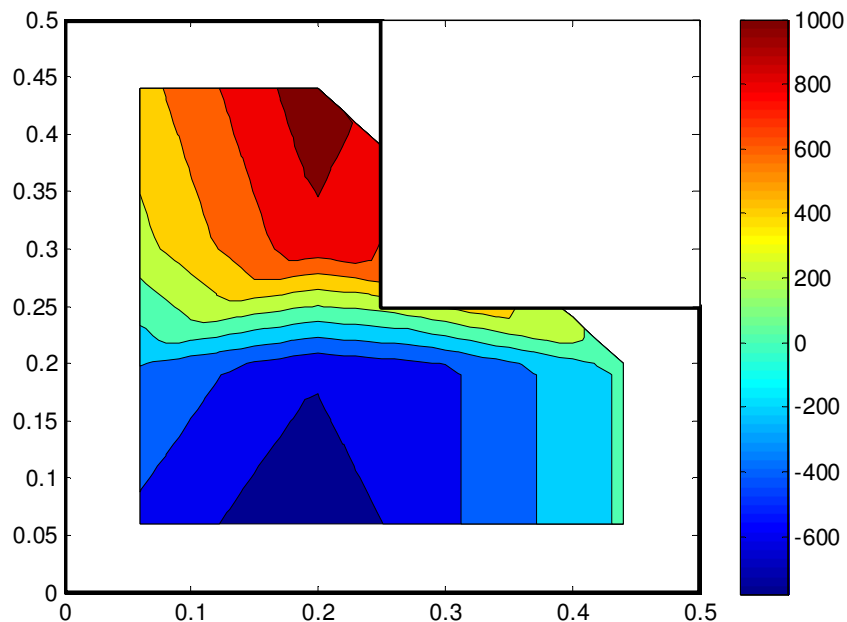


Proszę zauważyć, że zostały wyrysowane wartości składowej  $\sigma_x$  wyinterpolowane liniowo pomiędzy punktami Gaussa.

Zamykamy okno z rysunkiem. Powtarzając interpolację dla składowej  $\sigma_y$  oraz komendy graficzne:

```
>>[XI YI Sy] = griddata(gx,gy,sy,X,Y);
>>contourf(XI,YI,Sy)
>>line(wx,wy,'color','k','LineWidth',3)
>>rectangle('Position',[x5 y5 0.25 0.25],'FaceColor','w')
>>colorbar
```

powinien pojawić się rysunek:

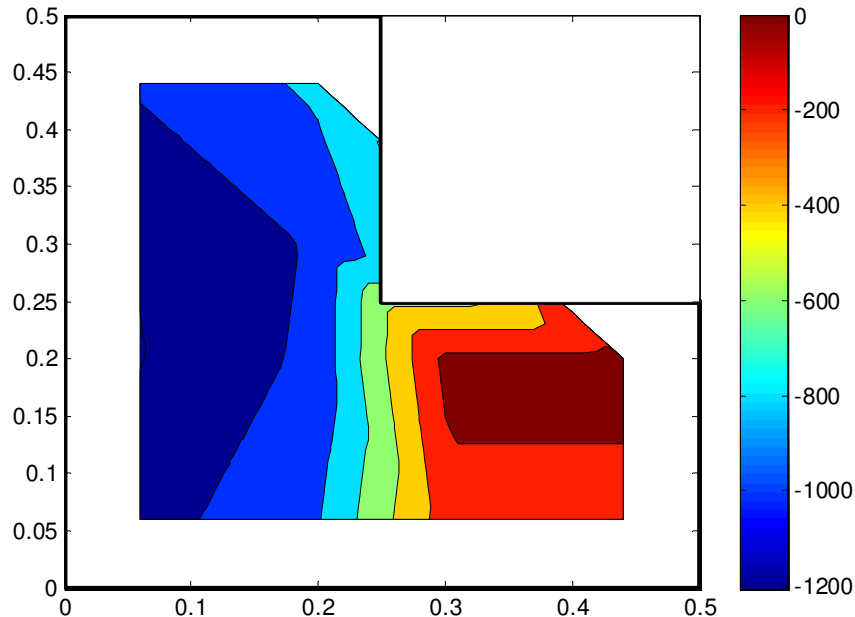


Zamykamy okno z rysunkiem. Powtarzając interpolację dla składowej  $\tau_{xy}$  oraz komendy graficzne:

```
>>[XI YI Txy] = griddata(gx,gy,txy,X,Y);
```

```
>>contourf(XI,YI,txy)
>>line(wx,wy,'color','k','LineWidth',3)
>>rectangle('Position',[x5 y5 0.25 0.25],'FaceColor','w')
>>colorbar
```

powinien pojawić się rysunek:



Proszę zinterpretować wyniki.

### Zadanie 2:

*Proszę sprawdzić, co się stanie jeśli w zadaniu nr 1 wszystkie elementy zdefiniujemy w kierunku przeciwnym do kierunku ruchu wskazówek zegara.*

### Zadanie 3:

*Proszę rozwiązać układ przedstawiony na rysunku poniżej, przyjmując parametry materiałowe oraz grubość tarczy z zadania nr 1.*

