

# METODY NUMERYCZNE

*Kwadratury Gaussa 2D*

**dr hab. inż. Marcin Hojny, prof. AGH**

**AGH Akademia Górniczo-Hutnicza**

Wydział Inżynierii Metali i Informatyki Przemysłowej

*Katedra Informatyki Stosowanej i Modelowania*

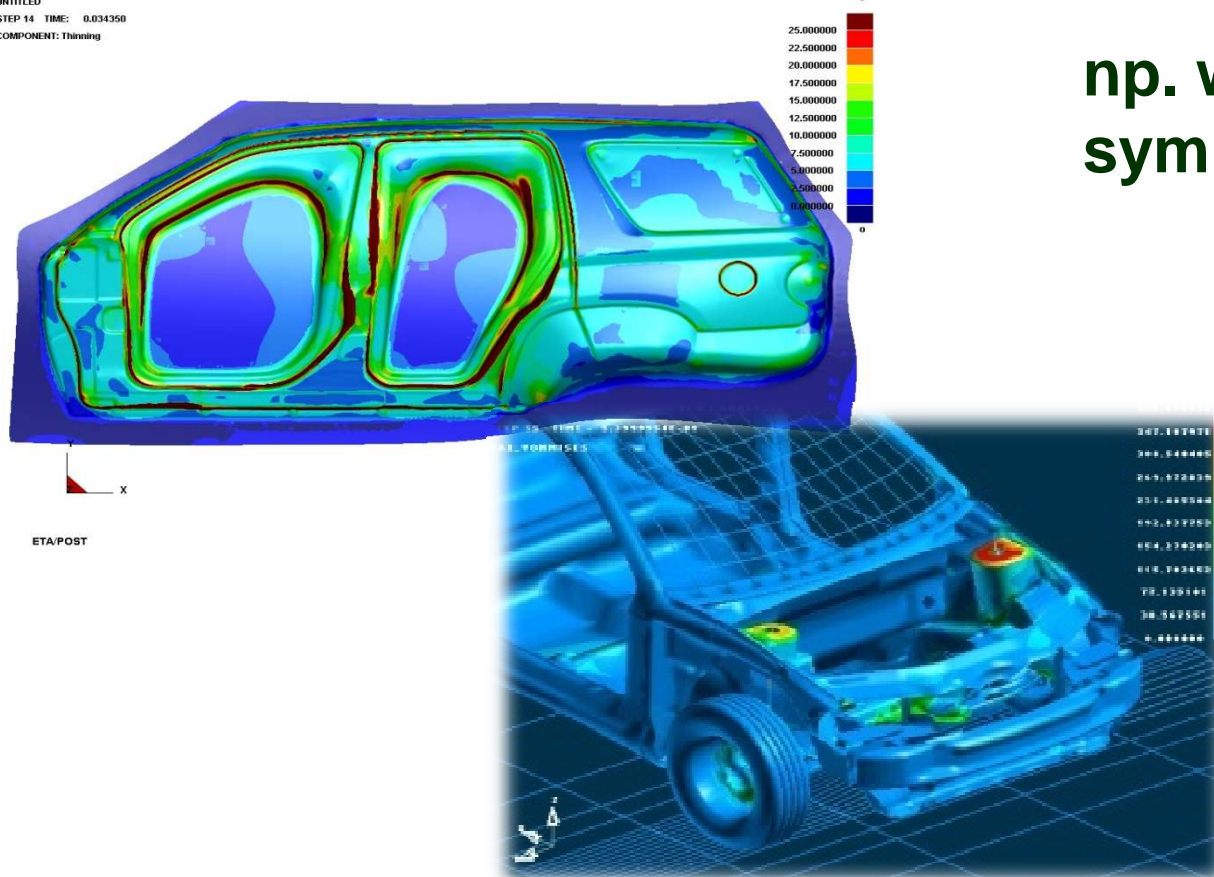
pawilon B5/p.406

tel. (+48)12 617 46 37

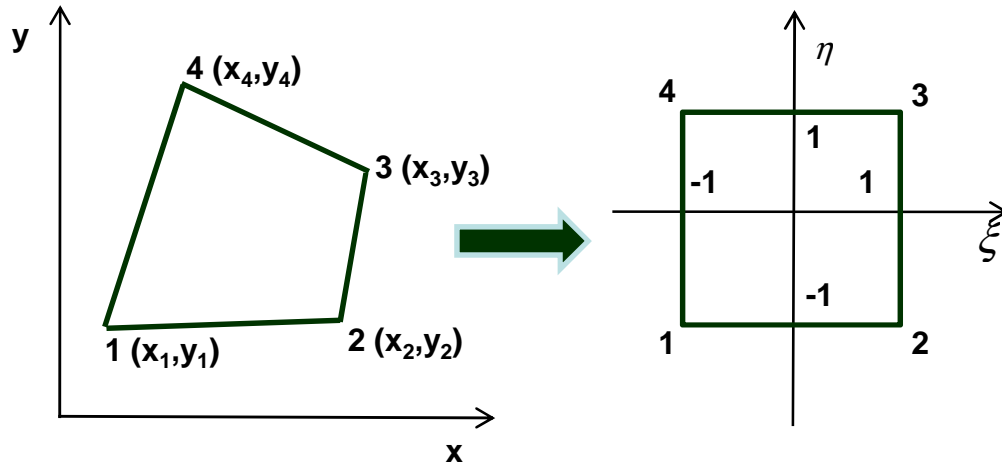
E-mail: [mhojny@metal.agh.edu.pl](mailto:mhojny@metal.agh.edu.pl)



UNTITLED  
STEP 14 TIME: 0.034350  
COMPONENT: Thinning



np. w projektowaniu systemów  
symulacji numerycznych



Element czworokątny raz jego odwzorowanie w lokalnym układzie współrzędnych

Układ współrzędnych przekształca się tak, aby element kwadratowy został odwzorowany przez kwadrat o wymiarach 2x2. Transformacja układu współrzędnych określona jest równaniem:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} \{N\}^T & 0 \\ 0 & \{N\}^T \end{bmatrix} \begin{Bmatrix} x_n \\ z_n \end{Bmatrix}$$

$$\{x_n\} = \{x_1, x_2, x_3, x_4\}^T$$

$$\{y_n\} = \{y_1, y_2, y_3, y_4\}^T$$

$$\{N\} = \frac{1}{4} \{(1-\xi)(1-\eta), (1+\xi)(1-\eta), (1+\xi)(1+\eta), (1-\xi)(1+\eta)\}^T$$

Pochodne względem  $\xi$  i  $\eta$  dane są równaniem:

$$\begin{Bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix}$$

gdzie:  $[J]$  - macierz Jacobiego, z której wyznacznik  $[J_0]$  jest Jakobianem transformacji układu współrzędnych.

Całkowanie funkcji w układzie  $\xi \eta$ , prowadzone jest metodą Gaussa, którą ilustruje wzór:

$$\iint f(x, y) dx dy = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) J_0 d\eta d\xi = \sum_{i=1}^n \sum_{j=1}^n w_i w_j f(\xi_i, \eta_j) J_0$$

$$w_0 = w_1 = 1$$

$$\xi_0 = \eta_0 = 0.5773502692$$

$$\xi_1 = \eta_1 = -0.5773502692$$

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad \det |J| = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta}$$

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i \quad \frac{\partial x}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i$$

$$\frac{\partial y}{\partial \xi} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i \quad \frac{\partial y}{\partial \eta} = \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i$$

$$N_1(\xi, \eta) = 0.25(1 - \xi)(1 - \eta)$$

$$N_2(\xi, \eta) = 0.25(1 + \xi)(1 - \eta)$$

$$N_3(\xi, \eta) = 0.25(1 + \xi)(1 + \eta)$$

$$N_4(\xi, \eta) = 0.25(1 - \xi)(1 + \eta)$$

```
Program KwadraturaGaussa2D

implicit REAL*8 (A-H,O-Z)

real*8 WSP_X(4), WSP_Y(4)
real*8 WAGA(2), PUNKT(2)
real*8 FKSZTALT(2,2,4)
real*8 POCH_KSI(2,4), POCH_NI(2,4), FUN_DETJ(2,2)

open (11, file= 'input.k', status='unknown')

read (11,*) WSP_X(1), WSP_Y(1)
read (11,*) WSP_X(2), WSP_Y(2)
read (11,*) WSP_X(3), WSP_Y(3)
read (11,*) WSP_X(4), WSP_Y(4)

WAGA(1)= 1.0
WAGA(2)= 1.0

PUNKT(1)= -0.5773502692
PUNKT(2)= 0.5773502692

do jpn= 1, 2
do ipn= 1, 2

FKSZTALT(ipn,jpn,1)= 0.25*(1.0-PUNKT(ipn))*(1.0-PUNKT(jpn))
FKSZTALT(ipn,jpn,2)= 0.25*(1.0+PUNKT(ipn))*(1.0-PUNKT(jpn))
FKSZTALT(ipn,jpn,3)= 0.25*(1.0+PUNKT(ipn))*(1.0+PUNKT(jpn))
FKSZTALT(ipn,jpn,4)= 0.25*(1.0-PUNKT(ipn))*(1.0+PUNKT(jpn))

POCH_KSI(jpn,1)= -0.25*(1.0-PUNKT(jpn))
POCH_KSI(jpn,2)= 0.25*(1.0-PUNKT(jpn))
POCH_KSI(jpn,3)= 0.25*(1.0+PUNKT(jpn))
POCH_KSI(jpn,4)= -0.25*(1.0+PUNKT(jpn))
POCH_NI(ipn,1)= -0.25*(1.0-PUNKT(ipn))
POCH_NI(ipn,2)= -0.25*(1.0+PUNKT(ipn))
POCH_NI(ipn,3)= 0.25*(1.0+PUNKT(ipn))
POCH_NI(ipn,4)= 0.25*(1.0-PUNKT(ipn))
enddo
enddo
```

```
do jpn= 1, 2
do ipn= 1, 2

dxdkSI = POCH_KSI(jpn,1)*WSP_X(1) + POCH_KSI(jpn,2)*WSP_X(2) +
& POCH_KSI(jpn,3)*WSP_X(3) + POCH_KSI(jpn,4)*WSP_X(4)

dydkSI = POCH_KSI(jpn,1)*WSP_Y(1) + POCH_KSI(jpn,2)*WSP_Y(2) +
& POCH_KSI(jpn,3)*WSP_Y(3) + POCH_KSI(jpn,4)*WSP_Y(4)

dxkNI = POCH_NI(ipn,1)*WSP_X(1) + POCH_NI(ipn,2)*WSP_X(2) +
& POCH_NI(ipn,3)*WSP_X(3) + POCH_NI(ipn,4)*WSP_X(4)

dykNI = POCH_NI(ipn,1)*WSP_Y(1) + POCH_NI(ipn,2)*WSP_Y(2) +
& POCH_NI(ipn,3)*WSP_Y(3) + POCH_NI(ipn,4)*WSP_Y(4)

FUN_DETJ(ipn,jpn)= dxdkSI*dykNI - dxkNI*dydkSI

enddo
enddo

POWIERZCHNIA= 0.0

do jpn= 1, 2
do ipn= 1, 2

POWIERZCHNIA=POWIERZCHNIA+dabs(FUN_DETJ(ipn,jpn))*WAGA(ipn)*WAGA(jpn)

enddo
enddo

write (*,*) 'POWIERZCHNIA WYNOSI=', POWIERZCHNIA
write (*,*) ''
write (*,*) ''
write (*,*) 'Dowolny znak i ENTER - wyjscie z programu'
read (*,*) znak
write (*,*) ''
write (*,*) ''

end
```