پروژه درس اجزا محدود

تحلیل اجزا محدود با استفاده از المان هایQ8 & Q4 & LST & CST

سارا حاجی زاده
دانشجوی کارشناسی ارشد عمران-زلزله دانشگاه تربیت مدرس

Sarah_hajizadeh@modares.ac.ir

مقدمه

در واقعیت تمام اجسام صلب سه بعدی هستند ولی خوشبختانه برای بسیاری از مسائل که در عمل مورد توجه هستند می توان برای توزیع تنش و کرنش ساده سازی هایی انجام داد. در صورتی که نحوه بارگذاری و هندسه المان اجازه دهدمی توان جسم صلب را به عنوان مسئله تنش صفحه ای یا کرنش صفحه ای تحلیل کرد.همچنین برای اجسام صلبی که هم در بارگذاری و هم در هندسه تقارن دارند می توان ساده سازی هایی را انجام داد.برخلاف سازه های اسکلتی که گسسته سازی آن ها ساده است، اتصالات و مفاصل متصل به هم طبیعتا دارای گره هایی هستند که در نتیجه وجود آن ها کار گسسته سازی را نمی توان برای زنجیر های متصل دوبعدی یا سه بعدی انجام داد.مفاصلی وجود ندارند که بعنوان گره در نظر گرفته می شود یا خطوط حاشیه ای وجود ندارند که بعنوان حاشیه المان بکار بروند. بنابراین گسسته سازی به فرآیندی تبدیل می شود که مستلزم شناخت و درک فیزیکی مسئله است. بعلاوه باید افزود که هرچه سعی کنیم جزئیات فیزیکی بیشتری را درک کنیم مدل مورد نظر پیچیده تر می شود و کاربر باید درمورد انتخاب نوع و اندازه المان تصمیم گیری کند. تمام اینها به ساختار فیزیکی جسم نحوه بارگذاری و دقت مورد نیاز بستگی دارد.

فرمولبندی المان محدود برای مسائل صفحه ای

روابط تنش کرنش برای یک مسئله تنش صفحه ای:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xz} \end{Bmatrix}$$

برای کرنش صفحه ای:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & -\nu & 0 \\ -\nu & 1-\nu & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}$$

چه در حالت تنش صفحه ای و چه کرنش صفحه ای یک نقطه از جسم فقط می تواند در دو جهت Xو Y حرکت کندبنابراین دو متغیر جابجایی که در این مبحث نقش دارند عبارتست از Uو V.جابجایی های کرنشی بی نهایت کوچک برای هر دو تئوری یکسان است.

$$\epsilon_{xx} = \frac{\partial u}{\partial x}$$
$$\epsilon_{yy} = \frac{\partial v}{\partial y}$$
$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

$$\begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix}$$

$$\{\epsilon\} = [L]U$$

که در آن L عامل دیفرانسیل یا مشتق گیری خطی ست. و تنها مجهولات فقط Vو U هستند که اگر این ها محاسبه شوند می توان تنش ها و کرنشهارا نیز بدست آورد(به شرط برقراری سازگاری).

برای المانی که n گره دارد. جابجایی های مجهول با استفاده از تقریب درونیابی می شوند. بطوری که:

$$u = N_1 u_1 + N_2 u_2 + \cdots + N_n u_n$$
$$v = N_1 v_1 + N_2 v_2 + \cdots + N_n v_n$$

$$\{U\} = [N]a$$

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \left[\begin{array}{cc|cc|c|cc} N_1 & 0 & N_2 & 0 & \cdots & N_n & 0 \\ 0 & N_1 & 0 & N_2 & \cdots & 0 & N_n \end{array}\right] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix}$$

که در آن $\{a\} = \{u_1, v_1, u_2, v_2, \ldots, u_n, v_n\}$ بردار جابجایی گرهیست. با جایگذاری:

$$\{\epsilon\} = [B]\{a\}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & | & \frac{\partial N_2}{\partial x} & 0 & | & \dots & | & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & | & 0 & \frac{\partial N_2}{\partial y} & | & \dots & | & 0 & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & | & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & | & \dots & | & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{bmatrix}$$

با استفاده از اصل کارمجازی:

$$\int_{V_e} \delta\{\epsilon\}^T\{\sigma\}\, dV = \int_{V_e} \delta\{U\}^T\{b\}\, dV + \int_{\Gamma_e} \delta\{U\}^T\{t\}\, d\Gamma + \sum_i \delta\{U\}^T_{(\{x\}=\{\overline{x}\})}\{P\}_i$$

$\{\epsilon\}$ represents the strain vector
$\{\sigma\}$ is the stress vector
$\{U\}$ is the displacements vector
$\{b\}$ is the body forces vector
$\{t\}$ is the traction forces vector
$\{P\}_i$ is the vector of concentrated forces applied at $\{x\} = \{\overline{x}\}$
dv is an element of volume
$d\Gamma$ is an element of the boundary of the element on which the traction forces $\{t\}$ are applied

می توان نوشت:

$$\{\delta\epsilon\} = \delta([B]\{a\}) = [B]\{\delta a\}$$
$$\{\delta U\} = \delta([N]\{a\}) = [N]\{\delta a\}$$

با جایگذاری در روابط تنش کرنش:

$$\{\sigma\} = [D]\{\epsilon\} = [D][B]\{a\}$$

با جایگذاری در رابطه کارمجازی:

$$\int_{V_e} \delta\{a\}^T[B]^T[D][B]\{a\}\,dV = \int_{V_e} \delta\{a\}^T[N]^T\{b\}\,dV + \int_{\Gamma_e} \delta\{a\}^T[N]^T\{t\}\,d\Gamma + \sum_i \delta\{a\}^T[N_{(\{x\}=\{\overline{x}\})}]^T\{P\}_i$$

داریم:

$$\left[\int_{A_e}[B]^T[D][B]tdA\right]\{a\} = \int_{A_e}[N]^T\{b\}tdA + \int_{L_e}[N]^T\{t\}tdl + \sum_i[N_{(\{x\}=\{\overline{x}\})}]^T\{P\}_i$$

$$[K_e]\{a\} = f_e$$

$$[K_e] = \left[\int_{A_e}[B]^T[D][B]tdA\right]$$

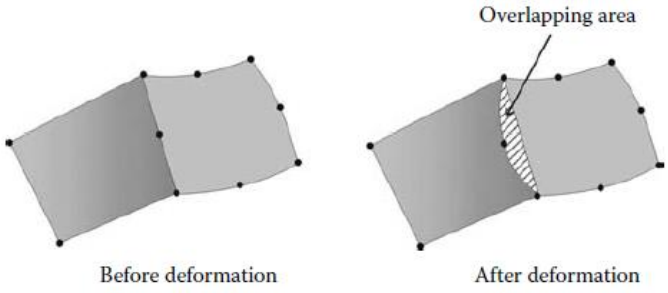
بردار نیویی المان:

$$\{f_e\} = \int_{A_e}[N]^T\{b\}tdA + \int_{L_e}[N]^T\{t\}tdl + \sum_i[N_{(\{x\}=\{\overline{x}\})}]^T\{P\}_i$$

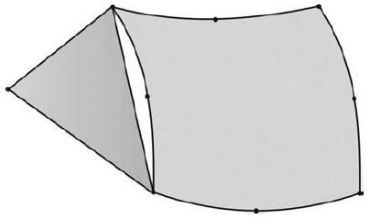
گسسته سازی فضایی

اولین مرحله در تحلیل المان محدود جدا کردن دامنه و تبدیل آن به شبکه مناسبی از المان هاست.در بدست آوردن یک شبکه هیچ راه منحصر بفردی وجود ندارد. اما باید شرایط زیر را در نظر گرفت:

- دو المان مجزا اگر مرز مشترکی داشته باشند فقط می توانند در گره هایی که روی مرز مشترک وجود دارد مشترک باشند.این شرط برهم پوشانندگی بین دو یا چند المان را شامل نمی شود. شکل زیر یکی از متداول ترین خطاهای گسسته سازی یعنی برهم پوشانندگی بین المان هارا نشان می دهد.

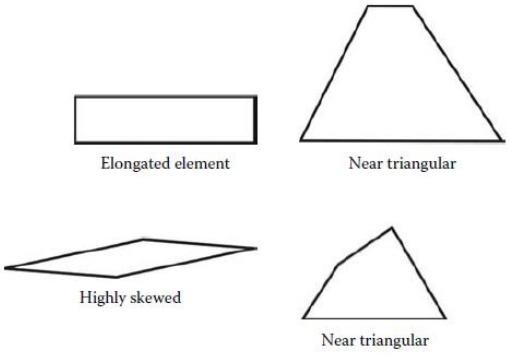


- دامنه شبکه بندی شده باید در حد امکان شبیه دامنه اصلی باشد. وجود حفره ها بین المان ها صحیح نیست مگر اینکه بطور فیزیکی در دامنه اصلی وجود داشته باشند.



Discretization error involving holes between elements.

- باید از بکار بردن المان های طویل یا اریب جلوگیری کردچون میزان دقت را کاهش می دهد.



Plane elements with shape distortions.

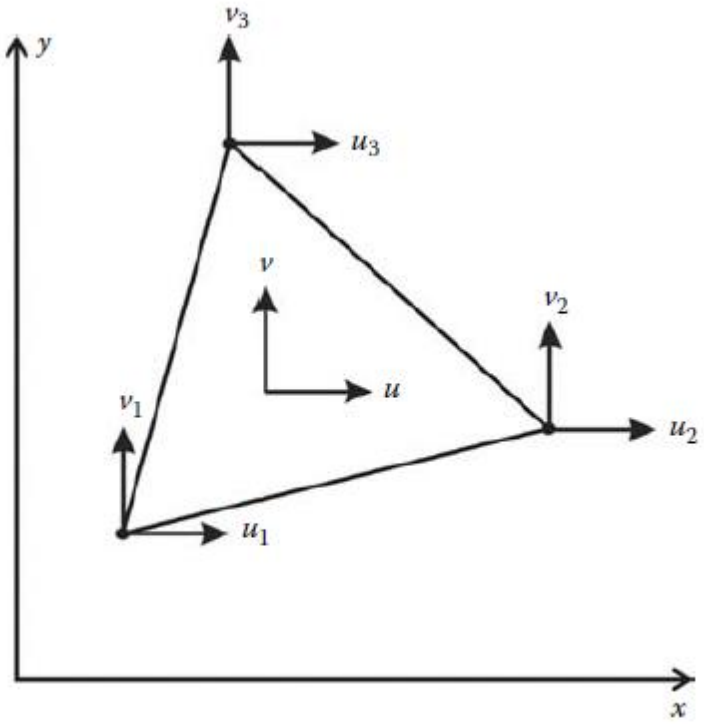
- وقتی حوزه هایی با مرز های کج و خمیده را مش بندی می کنیم خطاهایی در گسسته سازی رخ می دهد که گریز ناپذیر می باشد. اما با مش بندی ریز با استفاده از المان های درجه بالاتر می توان این خطاهارا به حداقل رساند.

Geometrical discretization error.

مثلث کرنش ثابت

المان مثلثی شکل زیر سه گره دارد و هر گره دو درجه آزادی. توابع شکل بدین شکل نوشته می شود:

$$N_1(x,y) = m_{11} + m_{12}x + m_{13}y$$
$$N_2(x,y) = m_{21} + m_{22}x + m_{23}y$$
$$N_3(x,y) = m_{31} + m_{32}x + m_{33}y$$



Linear triangular element.

$$m_{11}=\frac{x_2y_3-x_3y_2}{2A} \qquad m_{12}=\frac{y_2-y_3}{2A} \qquad m_{13}=\frac{x_3-x_2}{2A}$$
$$m_{21}=\frac{x_3y_1-x_1y_3}{2A} \qquad m_{22}=\frac{y_3-y_1}{2A} \qquad m_{23}=\frac{x_1-x_3}{2A}$$
$$m_{31}=\frac{x_1y_2-x_2y_1}{2A} \qquad m_{32}=\frac{y_1-y_2}{2A} \qquad m_{33}=\frac{x_2-x_1}{2A}$$

$$A=\frac{1}{2}\det\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{bmatrix}$$

حوزه جابجایی

حوزه جابجایی اینگونه تقریب زده می شود:

$$u=N_1u_1+N_2u_2+N_3u_3$$
$$v=N_1v_1+N_2v_2+N_3v_3$$

$$\begin{Bmatrix}u\\ v\end{Bmatrix}=\left[\begin{array}{cc|cc|cc}N_1 & 0 & N_2 & 0 & N_3 & 0\\ 0 & N_1 & 0 & N_2 & 0 & N_3\end{array}\right]\begin{Bmatrix}u_1\\ v_1\\ u_2\\ v_2\\ u_3\\ v_3\end{Bmatrix}$$

$$\{U\}=[N]\{a\}$$

ماتریس کرنش

$$\{\epsilon\}=[B]\{a\}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix}$$

$$[B] = \begin{bmatrix} m_{12} & 0 & m_{22} & 0 & m_{32} & 0 \\ 0 & m_{13} & 0 & m_{23} & 0 & m_{33} \\ m_{13} & m_{12} & m_{23} & m_{22} & m_{33} & m_{32} \end{bmatrix}$$

توجه شود که ماتریس $[B]$ از مختصات دکارتی XوY مستقل است. این ماتریس فقط تابعی از مختصات گرهی و در سراسر المان ثابت است. در نتیجه بردار کرنشی در سراسر المان ثابت است.

ماتریس سختی

$$[K_e] = [B]^T [D][B] t A_e$$

نیروهای ناشی از جاذبه جسم

$$\int_{A_e} [N]^T \{b\} t \, dA = t \int_{A_e} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \begin{Bmatrix} 0 \\ -\rho g \end{Bmatrix} dA = t \begin{bmatrix} 0 \\ -\int_{A_e} N_1 \rho g \, dA \\ 0 \\ -\int_{A_e} N_2 \rho g \, dA \\ 0 \\ -\int_{A_e} N_3 \rho g \, dA \end{bmatrix}$$

$$\int_{A_e} N_1 \rho g \, dA = \rho g \int_{A_e} N_1^1 N_2^0 N_3^0 \, dA = \rho g \frac{1!0!0!}{(1+0+0+2)!} 2A_e = \rho g \frac{A_e}{3}$$

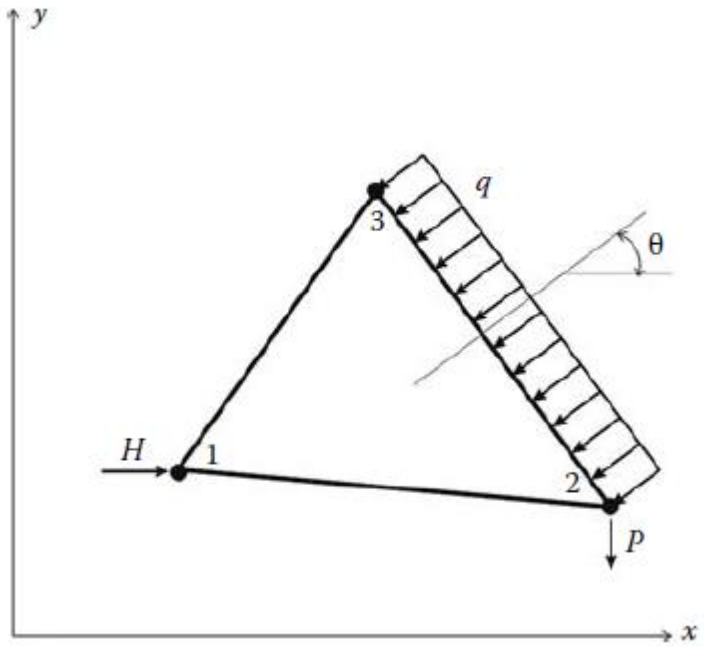
$$\int_{A_e} N_2 \rho g \, dA = \rho g \int_{A_e} N_1^0 N_2^1 N_3^0 \, dA = \rho g \frac{0!1!0!}{(0+1+0+2)!} 2A_e = \rho g \frac{A_e}{3}$$

$$\int_{A_e} N_3 \rho g \, dA = \rho g \int_{A_e} N_1^0 N_2^0 N_3^1 \, dA = \rho g \frac{0!0!1!}{(0+0+1+2)!} 2A_e = \rho g \frac{A_e}{3}$$

$$\int_{A_e} [N]^T \{b\} t \, dA = -\frac{t}{3} \begin{Bmatrix} 0 \\ \rho g A_e \\ 0 \\ \rho g A_e \\ 0 \\ \rho g A_e \end{Bmatrix}$$

نیروهای انقباضی

المان شکل زیر را در نظر بگیرید که در معرض باری به بزرگی q قرار دارد که بطور یکنواخت و نرمال بر وجه ۲-۳ وارد می شود و با محور اصلی زاویه می سازد. بنابراین می توان نوشت:



$$\int_{L_e}[N]^T\{t\}t\,dl=\int_{L_{2-3}}\begin{bmatrix}0&0\\0&0\\N_2&0\\0&N_2\\N_3&0\\0&N_3\end{bmatrix}\begin{Bmatrix}-q\cos\theta\\-q\sin\theta\end{Bmatrix}t\,dl=t\int_{L_{2-3}}\begin{Bmatrix}0\\0\\-N_2q\cos\theta\\-N_2q\sin\theta\\-N_3q\cos\theta\\-N_3q\sin\theta\end{Bmatrix}dl$$

$\{t\}=\{-q\cos\theta,-q\sin\theta\}^T.$ بر روی وجه ۲-۳ داریم: $N_1=0$

$$\int_{L_e}[N]^T\{t\}t\,dl=t\begin{Bmatrix}0\\0\\-q\cos\theta L_{2-3}/2\\-q\sin\theta L_{2-3}/2\\-q\cos\theta L_{2-3}/2\\-q\sin\theta L_{2-3}/2\end{Bmatrix}$$

در اینجا گره های ۲و ۳ بار را بطور یکسان بین خودشان تقسیم کرده اند.

نیروهای متمرکز

مطابق شکل قبلی المان تحت نیروی افقی و عمودی HوP قرار گرفته است.

$$\sum_i [N_{([x]=[\bar{x}])}]^T \{P\}_i = \begin{bmatrix} N_1 = 1 & 0 \\ 0 & N_1 = 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} H \\ 0 \end{Bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ N_2 = 1 & 0 \\ 0 & N_2 = 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ -P \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ H \\ 0 \\ 0 \\ -P \end{Bmatrix}$$

برنامه کامپیوتری با استفاده از مثلث کرنش ثابت

تیر طره ای را در نظر میگیریم که راه حل تحلیلی دقیق دارد. و جابجایی عمودی اش اینگونه بدست می آید:

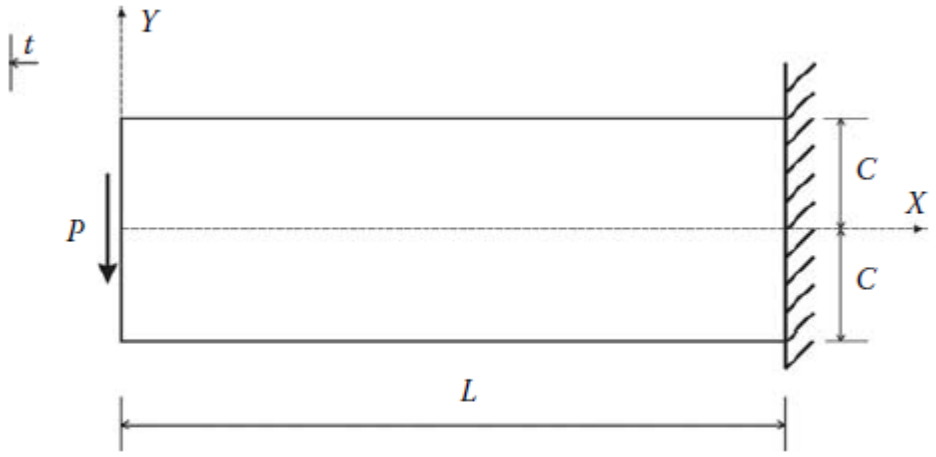
$$v = \frac{\nu P x y^2}{2EI} + \frac{P x^3}{6EI} - \frac{P L^2 x}{2EI} + \frac{P L^3}{3EI}$$

برخی مقادیر عددی برای بعدها و ثابت های الاستیک و بار:

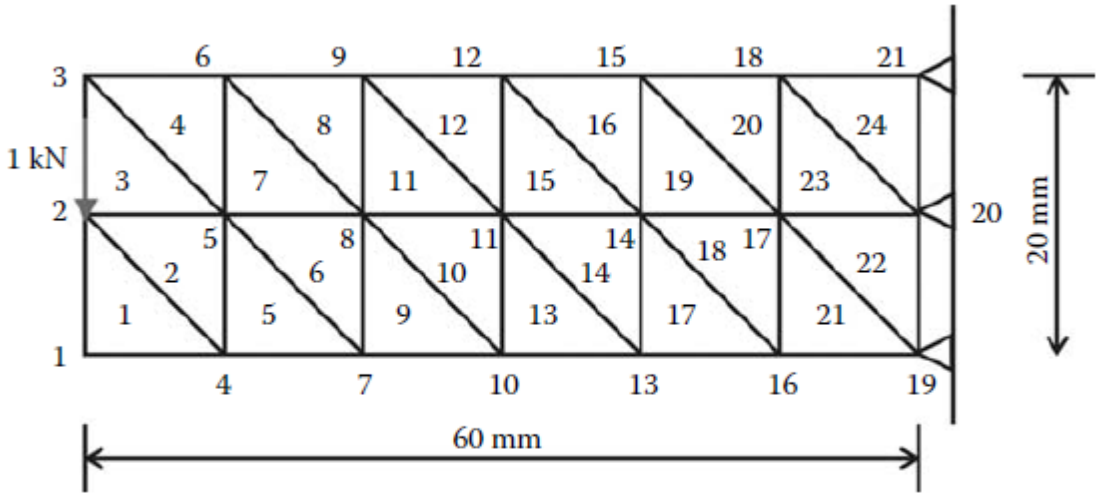
$C = 10\,\text{mm}, L = 60\ \ \text{mm}, t = 5$
Young's modulus of 200000 MPa
ratio of 0.3
Poisson's

P of 1000 N

برای گسسته سازی از ۲۴ المان استفاده می کنیم. گره های شماره ۱۹،۲۰ و ۲۱ ثابت هستند.



Analysis of a cantilever beam in plane stress.



Finite element discretization with linear triangular elements.

FILE: CST_COARSE_MESH_DATA.m

```
% File: CST_COARSE_MESH_DATA.m
%
% The following variables are declared as global in order
% to be used by all the functions (M-files) constituting
% the program
%
%
global nnd nel nne nodof eldof n
```

```
global geom connec dee nf Nodal_loads
%
format short e
%
nnd = 21 ; % Number of nodes:
nel = 24 ; % Number of elements:
nne = 3 ; % Number of nodes per element:
nodof =2; % Number of degrees of freedom per node
eldof = nne*nodof; % Number of degrees of freedom per element
%
% Nodes coordinates x and y
%
geom = zeros(nnd,2);
%
geom = [ 0, -10; ... % Node 1
0, 0; ... % Node 2
0, 10; ... % Node 3
10, -10; ... % Node 4
10, 0; ... % Node 5
10, 10; ... % Node 6
20, -10; ... % Node 7
20, 0; ... % Node 8
20, 10; ... % Node 9
30, -10; ... % Node 10
30, 0; ... % Node 11
30, 10; ... % Node 12
40, -10; ... % Node 13
40, 0; ... % Node 14
40, 10; ... % Node 15
50, -10; ... % Node 16
50, 0; ... % Node 17
50, 10; ... % Node 18
60, -10; ... % Node 19

60, 0; ... % Node 20

60, 10]; % Node 21
%
% Element connectivity
%
connec=zeros(nel,3);
connec = [ 1, 4, 2; ...% Element 1
4, 5, 2; ...% Element 2
2, 5, 3; ...% Element 3
5, 6, 3; ...% Element 4
4, 7, 5; ...% Element 5
7, 8, 5; ...% Element 6
5, 8, 6; ...% Element 7
```

```
8, 9, 6; ...% Element 8
7, 10, 8; ...% Element 9
10, 11, 8; ...% Element 10
8, 11, 9; ...% Element 11
11, 12, 9; ...% Element 12
10, 13, 11; ...% Element 13
13, 14, 11; ...% Element 14
11, 14, 12; ...% Element 15
14, 15, 12; ...% Element 16
13, 16, 14; ...% Element 17
16, 17, 14; ...% Element 18
14, 17, 15; ...% Element 19
17, 18, 15; ...% Element 20
16, 19, 17; ...% Element 21
19, 20, 17; ...% Element 22
17, 20, 18; ...% Element 23
20, 21, 18]; % Element 24
%
% Material
%
E = 200000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 5.; % Beam thickness in mm
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
nf(19,1) = 0; nf(19,2) = 0; % Prescribed nodal freedom of
node 19
nf(20,1) = 0; nf(20,2) = 0; % Prescribed nodal freedom of
node 20
nf(21,1) = 0; nf(21,2) = 0; % Prescribed nodal freedom of
node 21
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
```

```
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2);
%
Nodal_loads(2,1) = 0.; Nodal_loads(2,2) = -1000.; % Node 2
%
%%%%%%%%%%%% End of input %%%%%%%%%%%%
```

داده های ورودی این تیر عبارتند از:

nnd=21; تعداد گره ها

nel=24;تعداد المانها

nne=3;تعداد گره های هر المان

nodof=2;تعداد درجات آزادی به ازای هر گره

مختصات گره ای

مختصات XوY گره ها به شکل ماتریس **geom**(**nnd**, **2**) نوشته می شود.

پیوستگی المان

پیوستگی در ماتریس **connec(nel**, **3)** نوشته می شود. عددگذاری با شماره گذاری داخلی گره ها برخلاف عقربه های ساعت است.

خواص ماده

خواص یعنی مدول الاستسیسته و نسبت پواسون به صورت متغیرهای **E = 200000** and **vu = 0.3**. نوشته می شود. با این خواص ماتریس الاستیسیته تنش صفحه ای را با استفاده از تابع **formdsig.m** می نویسیم که ماتریس **dee** را برگرداند.

```
function[dee] = formdsig(E,vu)
%
% This function forms the elasticity matrix for a plane
stress problem
%
c=E/(1.-vu*vu);
%
dee=c*[1 vu 0. ;...
vu 1 0. ;...
0. 0. .5*(1.-vu)];
%
% end function formdsig
```

شرایط مرزی

درجات آزادی محدود شده با صفر و درجه آزادی آزاد با یک نشان داده می شود. گره های ۱۹،۲۰،۲۱ انتهای ثابت تیر طره هستندو درجات آزادی مربوط به اینها با ۰ نشان داده می شوند و تمام درجات آزادی گزه های دیگر با ۱. اطلاعات شرایط مرزی در ماتریس **nf(nnd, nodof)**. نوشته می شوند.

بارگذاری

بارمتمرکط ۱۰۰۰ نیوتنی بر گره ۲ وارد می شود. در برنامه اصلی این بار در بردار نیروی کلی **fg** اسمبل می شود.

برنامه اصلی

```
% THIS PROGRAM USES AN 3-NODE LINEAR TRIANGULAR ELEMENT FOR
THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
%
clear all
clc
%
% Make these variables global so they can be shared by other
functions
%
global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
%
format long g
%
```

```
% ALTER NEXT LINES TO CHOOSE THE NAME OF THE OUTPUT FILE
%
fid =fopen('CST_COARSE_MESH_RESULTS.txt','w');

%
% To change the size of the problem or change elastic
properties
% supply another input file
%
CST_COARSE_MESH_DATA;
%
%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0;
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Assembly of the global stiffness matrix
%
% initialize the global stiffness matrix to zero
%
kk = zeros(n, n);
%
for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering
vector
ke=thick*A*bee'*dee*bee; % Compute stiffness matrix
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
node_disp=zeros(nnd,2);
```

```
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
node_disp(i,:) =[x_disp y_disp];
end
%
% Retrieve the x_coord and y_disp of the nodes located on the
neutral axis
%
k = 0;
for i=1:nnd;
if geom(i,2)== 0.
k=k+1;
x_coord(k) = geom(i,1);
vertical_disp(k)=node_disp(i,2);
end
end
%
%
for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof
if g(m)==0 eld(m)=0.;
else %
eld(m)=delta(g(m)); % Retrieve element displacement
end
end
%
eps=bee*eld; % Compute strains
EPS(i,:)=eps ; % Store strains for all elements
sigma=dee*eps; % Compute stresses
SIGMA(i,:)=sigma ; % Store strains for all elements
end
%
% Print results to file
```

```
%
print_CST_results;
%
% Plot the stresses in the x_direction
%
x_stress = SIGMA(:,1);
cmin = min(x_stress);
cmax = max(x_stress);
caxis([cmin cmax])
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',x_stress, ...
'Facecolor','flat','Marker','o')
colorbar
%
Plottools
```

پس از مشخص کردن متغیر های کلی که توسط توابع زیر بکار می روند و نام گذاری فایل نتایج خروجی **CST_COARSE_MESH_RESULTS.txt** با آپلود فایل داده ها و اسمبل کردن بردار نیروی کلی **fg** برنامه آغاز می شود. ماتریس سختی کلی، راه حل معادلات کلی،و محاسبه تنش کرنش بصورت زیر بدست می آید:

ماتریس سختی المان

برای هر المان از ۱ تا nel ماتریس تنشی آن bee و بردار راهنمای آن g را می نویسیم و A را محاسبه میکنیم. این ها از طریق تابع **elem_T3.m** به دست می آیند.

```
function[bee,g,A] = elem_T3(i)
%
% This function returns the coordinates of the nodes of
element i
% and its steering vector
%
global nnd nel nne nodof eldof n
global geom connec dee nf load
%
x1 = geom(connec(i,1),1); y1 = geom(connec(i,1),2);
x2 = geom(connec(i,2),1); y2 = geom(connec(i,2),2);
x3 = geom(connec(i,3),1); y3 = geom(connec(i,3),2);
%
A = (0.5)*det([1 x1 y1; ...
1 x2 y2; ...
```

```
1 x3 y3]);
%
m11 = (x2*y3 - x3*y2)/(2*A);
m21 = (x3*y1 - x1*y3)/(2*A);
m31 = (x1*y2 - y1*x2)/(2*A);
m12 = (y2 - y3)/(2*A);
m22 = (y3 - y1)/(2*A);
m32 = (y1 - y2)/(2*A);
m13 = (x3 - x2)/(2*A);
m23 = (x1 - x3)/(2*A);
m33 = (x2 -x1)/(2*A);
%
bee = [ m12 0 m22 0 m32 0; ...
0 m13 0 m23 0 m33; ...
m13 m12 m23 m22 m33 m32] ;
%
l=0;
for k=1:nne
for j=1:nodof
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_T3
```

برای هر المان **i** مختصات XوY آن را می نویسیم.

x1 = geom(connec(i, 1), 1); y1 = geom(connec(i, 1), 2)
x2 = geom(connec(i, 2), 1); y2 = geom(connec(i, 2), 2)
x3 = geom(connec(i, 3), 1); y3 = geom(connec(i, 3), 2)

ضرایب **j**و**k**=۱و۲و۳ و **mjk** را با استفاده از روابط زیر محاسبه می کنیم.

$$m_{11} = \frac{x_2y_3 - x_3y_2}{2A} \qquad m_{12} = \frac{y_2 - y_3}{2A} \qquad m_{13} = \frac{x_3 - x_2}{2A}$$

$$m_{21} = \frac{x_3y_1 - x_1y_3}{2A} \qquad m_{22} = \frac{y_3 - y_1}{2A} \qquad m_{23} = \frac{x_1 - x_3}{2A}$$

$$m_{31} = \frac{x_1y_2 - x_2y_1}{2A} \qquad m_{32} = \frac{y_1 - y_2}{2A} \qquad m_{33} = \frac{x_2 - x_1}{2A}$$

با استفاده از ضرایب **mjk** ماتریس bee را با استفاده از رابطه زیر اسمبل می کنیم.

$$[B] = \begin{bmatrix} m_{12} & 0 & | & m_{22} & 0 & | & m_{32} & 0 \\ 0 & m_{13} & | & 0 & m_{23} & | & 0 & m_{33} \\ m_{13} & m_{12} & | & m_{23} & m_{22} & | & m_{33} & m_{32} \end{bmatrix}$$

با استفاده از ماتریس درجه آزادی گرهی nf در ترکیب با ماتریس پیوستگی بردار راهنما g را برای المان می نویسیم.

$$\mathbf{g} = \begin{Bmatrix} \mathbf{nf(connec(1,1),1)} \\ \mathbf{nf(connec(1,1),2)} \\ \mathbf{nf(connec(2,1),1)} \\ \mathbf{nf(connec(2,1),2)} \\ \mathbf{nf(connec(3,1),1)} \\ \mathbf{nf(connec(3,1),2)} \end{Bmatrix}$$

وقتی ماتریس bee را نوشتیم، ماتریس سختی المان ke را اینگونه می توان بدست آورد که

ke = thick × A × beeT × dee × bee

اسمبل ماتریس سختی کل

همانطور که در شکل دیدیم المان مثلثی خطی ۶ درجه آزادی دارد. ماتریس سختی کل KK با استفاده از حلقه دوگانه بر اجزا بردار g اسمبل می شود.

```
function[KK]=form_KK(KK, kg, g)
%
% This function assembles the global stiffness matrix
%
global eldof
%
% This function assembles the global stiffness matrix
%
```

```
for i=1:eldof
if g(i) ~= 0
for j=1: eldof
if g(j) ~= 0
KK(g(i),g(j))= KK(g(i),g(j)) + kg(i,j);
end
end
end
end
%
%%%%%%%%%%%%% end function form_KK %%%%%%%%%%%%%%%%%
```

راه حل سیستم کلی معادلات

پاسخ سیستم کلی معادلات با یک رابطه بدست می آید.

delta = KK\fg

جابجایی گره ها

وقتی بردار جابجایی کل یعنی دلتا بدست آمد می توان هریک از جابجایی گره ای را بدست آورد. بر روی همه گره ها یک حلقه تشکیل می شود. اگر درجه آزادی **j** در گره **i** آزاد باشد یعنی **nf(i, j) ≠ 0**,

آنگاه می تواند جابجایی داشته باشد که صفر نیست. مقدار جابجایی از بردار جابجایی های کلی دلتا استخراج می شود:

node_disp(i, j) = delta(nf(i, j))

تنش ها و کرنش های المان

برای بدست آوردن تنش ها و کرنش های المان در تمام المان ها یک حلقه ایجاد می شود:

۱)ماتریس تنشی المان **bee** و بردار راهنما **g** را می نویسیم.

الف)برای بدست آوردن بردار جابجایی های المان **edg** روی درجات آزادی المان حلقه تشکیل می دهیم.

ب)اگر **g(j) = 0** باشد آنگاه درجه آزادی محدود است **edg (j) = 0**

ج)در غیر این صورت **edg (j) = delta (g (j))**

۲)بردار کرنشی المان را بدست می آوریم **eps = bee × edg**

۳)بردار تنشی المان را بدست می آوریم **sigma = dee × bee × edg**

۴)کرنش های المان هارا ذخیره می کنیم **EPS (i, :) = eps** تا در فایل نتیجه پرینت شود.

نتایج و بحث

پس از اجرای برنامه **CST_PLANE_STRESS.m** نتایج در فایل نوشتاری **CST_COARSE_MESH_RESULTS.txt** نوشته می شود:

CST_COARSE_MESH_RESULTS.txt

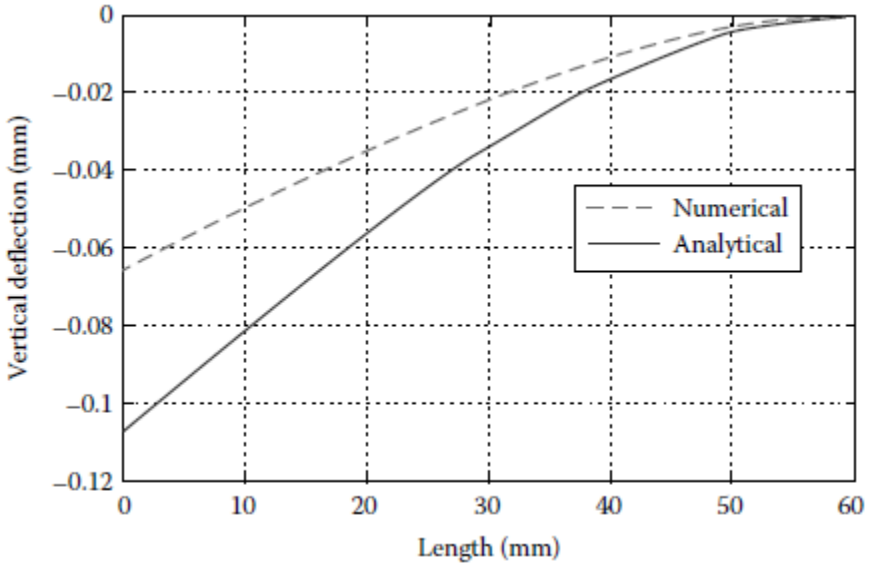
```
-----------------------------------------------------
******* PRINTING ANALYSIS RESULTS ************
-----------------------------------------------------
Nodal displacements
Node disp_x disp_y
1, 1.45081e-002, -6.49329e-002
2, 3.28049e-004, -6.52078e-002
3, -1.42385e-002, -6.47141e-002
4, 1.42332e-002, -4.97317e-002
5, 1.82950e-004, -4.94530e-002
6, -1.38358e-002, -4.94091e-002
7, 1.29745e-002, -3.50495e-002
8, 1.37982e-004, -3.46630e-002
9, -1.26721e-002, -3.47556e-002
10, 1.09224e-002, -2.19922e-002
11, 8.95233e-005, -2.14870e-002
12, -1.07002e-002, -2.16958e-002
13, 8.08085e-003, -1.13485e-002
14, 2.56420e-005, -1.07261e-002
15, -7.90991e-003, -1.10480e-002
16, 4.46383e-003, -3.88383e-003
17, -6.63586e-005, -3.19069e-003
18, -4.26507e-003, -3.66370e-003
19, 0.00000e+000, 0.00000e+000
20, 0.00000e+000, 0.00000e+000
21, 0.00000e+000, 0.00000e+000
-----------------------------------------------------
```

Element stresses
element sigma_(xx) sigma_(yy) tau_(xy)
1, -7.8546e+000, -7.8546e+000, 7.8546e+000
2, -1.3515e+000, 5.1683e+000, 1.3112e+001
3, 6.6118e-002, 9.8937e+000, 9.1400e+000
4, 9.1400e+000, 3.6192e+000, 9.8937e+000
5, -2.5827e+001, -2.1744e+000, 4.8607e+000
6, 1.5601e+000, 8.1980e+000, 1.5027e+001
7, -6.9913e-001, 6.6741e-001, 5.9323e+000
8, 2.4966e+001, 5.6374e+000, 1.4180e+001
9, -4.2552e+001, -5.0356e+000, 1.6983e+000
10, 2.2662e+000, 1.0785e+001, 1.8024e+001
11, -1.6757e+000, -2.3552e+000, 2.8152e+000
12, 4.1961e+001, 8.4119e+000, 1.7462e+001
13, -5.9121e+001, -7.6315e+000, -1.4550e+000
14, 2.6997e+000, 1.3258e+001, 2.0813e+001
15, -2.7809e+000, -5.0108e+000, -2.2163e-001
16, 5.9202e+001, 1.1322e+001, 2.0864e+001
17, -7.5391e+001, -1.0170e+001, -4.5429e+000
18, 2.5481e+000, 1.4627e+001, 2.3117e+001
19, -4.1445e+000, -7.6816e+000, -3.0783e+000
20, 7.6988e+001, 1.3636e+001, 2.4504e+001
21, -9.3536e+001, -1.4198e+001, -4.9720e+000
22, 1.4584e+000, 4.3753e-001, 2.4544e+001
23, -1.6603e+000, -9.9582e+000, -7.7540e+000
24, 9.3738e+001, 2.8121e+001, 2.8182e+001

Element strains
element epsilon_(xx) epsilon_(yy) gamma_(xy)
1, -2.7491e-005, -2.7491e-005, 1.0211e-004
2, -1.4510e-005, 2.7869e-005, 1.7045e-004
3, -1.4510e-005, 4.9369e-005, 1.1882e-004
4, 4.0271e-005, 4.3858e-006, 1.2862e-004
5, -1.2587e-004, 2.7869e-005, 6.3189e-005
6, -4.4967e-006, 3.8650e-005, 1.9535e-004
7, -4.4967e-006, 4.3858e-006, 7.7120e-005
8, 1.1637e-004, -9.2623e-006, 1.8434e-004
9, -2.0521e-004, 3.8650e-005, 2.2078e-005
10, -4.8459e-006, 5.0524e-005, 2.3431e-004
11, -4.8459e-006, -9.2623e-006, 3.6597e-005
12, 1.9719e-004, -2.0883e-005, 2.2701e-004
13, -2.8416e-004, 5.0524e-005, -1.8915e-005
14, -6.3881e-006, 6.2239e-005, 2.7057e-004
15, -6.3881e-006, -2.0883e-005, -2.8812e-006
16, 2.7903e-004, -3.2191e-005, 2.7123e-004
17, -3.6170e-004, 6.2239e-005, -5.9058e-005
18, -9.2001e-006, 6.9314e-005, 3.0052e-004

```
19, -9.2001e-006, -3.2191e-005, -4.0018e-005
20, 3.6448e-004, -4.7301e-005, 3.1856e-004
21, -4.4638e-004, 6.9314e-005, -6.4636e-005
22, 6.6359e-006, 0.0000e+000, 3.1907e-004
23, 6.6359e-006, -4.7301e-005, -1.0080e-004
24, 4.2651e-004, 0.0000e+000, 3.6637e-004
```

پس از آنکه محاسبات انجام شد اولین چیزی که باید بررسی شود این است که نتایج منطقی هستند؟اولین چیزی که می توانیم بررسی کنیم این است که آیا شکل شکسته شده صحیح است یا نه. برای این کار جابجایی های عمودی گره هارا رسم می کنیم. همانطور که بنظر می آید این شکل قابل قبول است.



برنامه ایجاد مش بندی خودکار

برای مدلسازی بهتر لازم است مش را ریزتز یا refine کنیم.

CST_PLANE_STRESS_MESH.m

```
% THIS PROGRAM USES AN 3-NODE LINEAR TRIANGULAR ELEMENT FOR
THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
% IT INCLUDES AN AUTOMATIC MESH GENERATION
%
% Make these variables global so they can be shared by other
functions
%
```

```
clear all
clc
global nnd nel nne nodof eldof n
global geom dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin
%
format long g
%
%
% To change the size of the problem or change elastic
properties
% supply another input file
%
Length = 60.; % Length of the model
Width =20.; % Width
NXE = 24; % Number of rows in the x direction
NYE = 10; % Number of rows in the y direction
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = Width/2. ; % Y origin of the global coordinate
system
%
nne = 3;
nodof = 2;
eldof = nne*nodof;
%
T3_mesh ; % Generate the mesh
%
% Material
% E = 200000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 5.; % Beam thickness in mm
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
% Restrain in all directions the nodes situated @
% (x = Length)
%
for i=1:nnd
```

```
if geom(i,1) == Length;
nf(i,:) = [0 0];
end
end
%
% Counting of the free degrees of freedom
%
n=0; for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply the load as a concentrated load on the node having
coordinate X = Y =0.
%
Force = 1000.; % N
%
for i=1:nnd
if geom(i,1) == 0. && geom(i,2) == 0.
Nodal_loads(i,:) = [0. -Force];
end
end
%
%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
```

```
% Assembly of the global stiffness matrix
%
% initialize the global stiffness matrix to zero
%
kk = zeros(n, n);
%

for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering
vector
ke=thick*A*bee'*dee*bee; % Compute stiffness matrix
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
node_disp(i,:) =[x_disp y_disp];
end
%
%
% Retrieve the x_coord and y_disp of the nodes located on the
neutral axis
%
k = 0;
vertical_disp=zeros(1,NXE+1);
for i=1:nnd;
if geom(i,2)== 0.
k=k+1;
x_coord(k) = geom(i,1);
vertical_disp(k)=node_disp(i,2);
```

```
end
end
%
for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof
if g(m)==0
eld(m)=0.;
else %
eld(m)=delta(g(m)); % Retrieve element displacement
end
end
%
eps=bee*eld; % Compute strains
EPS(i,:)=eps ; % Store strains for all elements
sigma=dee*eps; % Compute stresses
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
%
% Plot stresses in the x_direction
%
x_stress = SIGMA(:,1);
cmin = min(x_stress);
cmax = max(x_stress);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',x_stress, ...
'Facecolor','flat','Marker','o');

colorbar;
%
plottools;
```

T3_mesh.m

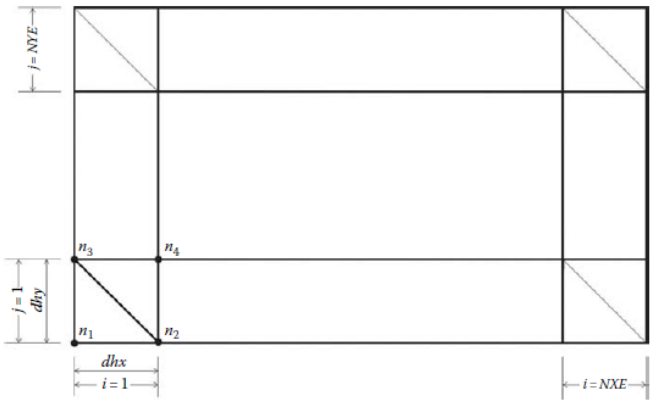
```
% This function generates a mesh of triangular elements
%
global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin dhx dhy
%
nnd = 0;
k = 0;
for i = 1:NXE
for j=1:NYE
k = k + 1;
```

```
n1 = j + (i-1)*(NYE + 1);
geom(n1,:) = [(i-1)*dhx - X_origin (j-1)*dhy - Y_origin ];
n2 = j + i*(NYE+1);
geom(n2,:) = [i*dhx - X_origin (j-1)*dhy - Y_origin ];
n3 = n1 + 1;
geom(n3,:) = [(i-1)*dhx - X_origin j*dhy - Y_origin ];
n4 = n2 + 1;
geom(n4,:) = [i*dhx- X_origin j*dhy - Y_origin ];
nel = 2*k;
m = nel -1;
connec(m,:) = [n1 n2 n3];
connec(nel,:) = [n2 n4 n3];
nnd = n4;
end
end
%
```

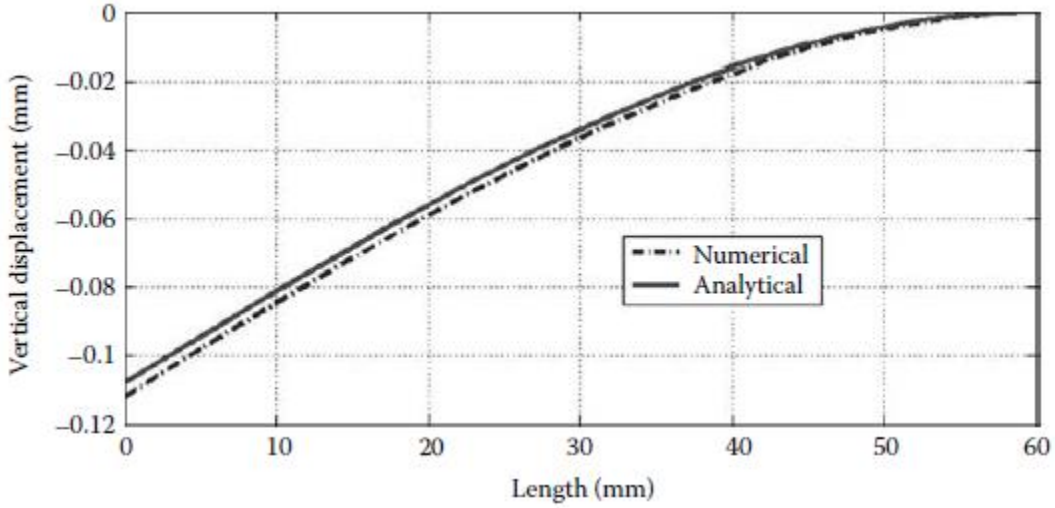
متغیر های NYE,NXE به ترتیب تعداد فواصل در طول جهت های y,x را نشان می دهند که در شکل زیر می توان مشاهده نمود. برای هر فاصله i,j ۴ گره $n1, n2, n3\ n4$ و دو المان ایجاد می شود. المان اول گره های $n1, n2, n3$ را دارد. درحالیکه المان دوم گره های $n2, n4, n3$ را دارد. در کل تعداد گره ها و المانهای ایجاد شده برابر است با:

$$nel = 2 \times NXE \times NYE$$

$$nnd = (NXE + 1) \times (NYE + 1)$$



همچنین این مدول ماتریس های $geom(nnd, 2)$ و $connec(nel, nne)$ را بدست می دهد.نتایج بدست آمده از مش ریز با راه حل تحلیلی هم خوابی بیشتری دارد.

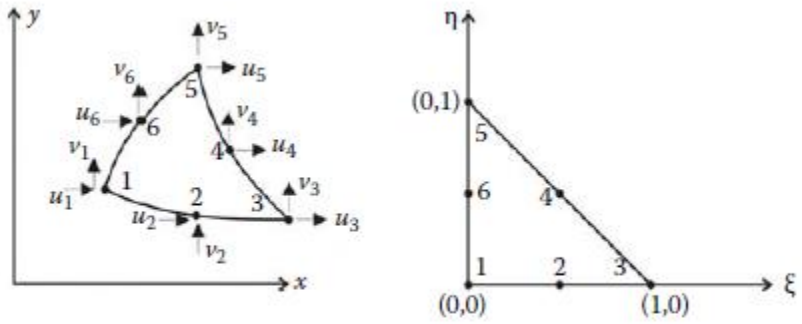


مثلث کرنش خطی

تطبیق پذیرترین المان در خانواده مثلث ها مثلث کرنش خطی ست. این مثلث ۶ گره دارد و وجه های آن می تواند ممکن است صاف یا خمیده باشند. ازین مثلث برای شبکه بندی دومین هایی با مرز های منحنی شکل استفاده می شود. توابع شکل بدین صورت تعریف می شود:

$$\begin{Bmatrix} N_1(\xi,\eta) \\ N_2(\xi,\eta) \\ N_3(\xi,\eta) \\ N_4(\xi,\eta) \\ N_5(\xi,\eta) \\ N_6(\xi,\eta) \end{Bmatrix} = \begin{Bmatrix} -\lambda(1-2\lambda) \\ 4\xi\lambda \\ -\xi(1-2\xi) \\ 4\xi\eta \\ -\eta(1-2\eta) \\ 4\eta\lambda \end{Bmatrix}$$

که در آن $\lambda = 1 - \xi - \eta$



Linear strain triangular element.

حوزه جابجایی

حوزه جابجایی به این شکل تقریب زده می شود.

$$u = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4 + N_5u_5 + N_6u_6$$
$$v = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4 + N_5v_5 + N_6v_6$$

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \\ u_5 \\ v_5 \\ u_6 \\ v_6 \end{Bmatrix}$$

$$\{U\} = [N]\{a\}$$

این المان ایزوپارامتریک است و مختصات xو y هر نقطه از المان مادر بدین شکل نوشته می شود:

$$x = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6$$
$$y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 + N_5y_5 + N_6y_6$$

که جفت مختصات (x_i, y_i) نشان دهنده مختصات گره هاست. ماتریس $[J]$ ژاکوبین تغییر شکل هندسی ست که به شکل زیر نوشته می شود.

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{6} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{6} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{6} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{6} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}$$

$$[J] = \frac{1}{4} \begin{bmatrix} 1-4\lambda & 4(\lambda-\xi) & -1+4\xi & 4\eta & 0 & -4\eta \\ 1-4\lambda & -4\xi & 0 & 4\xi & -1+4\eta & 4(\lambda-\eta) \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ x_5 & y_5 \\ x_6 & y_6 \end{bmatrix}$$

که در آن $\lambda = 1 - \xi - \eta.$

ماتریس کرنشی

$$\{\epsilon\} = [B]\{a\}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 & \frac{\partial N_5}{\partial x} & 0 & \frac{\partial N_6}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} & 0 & \frac{\partial N_5}{\partial y} & 0 & \frac{\partial N_6}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} & \frac{\partial N_5}{\partial y} & \frac{\partial N_5}{\partial x} & \frac{\partial N_6}{\partial y} & \frac{\partial N_6}{\partial x} \end{bmatrix}$$

برای محاسبه ماتریس B لازم است رابطه مشتق های جزئی در مختصات در مختصات (x,y) با مختصات مکانی(ε,η) را بنویسیم.

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi}$$
$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta}$$

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

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix}$$

ماتریس سختی

$$[K_e] = \left[\int_{A_e} [B]^T [D][B] t \, dA \right]$$

انتگرال گیری از حجم بدین صورت از فرمول هامرمحاسبه می شود.

$$[K_e] = t \int_0^{+1} \int_0^{1-\xi} [B(\xi,\eta]^T [D][B(\xi,\eta)] \det[J(\xi,\eta)] d\eta \, d\xi$$
$$= t \sum_{i=1}^{nhp} W_i [B(\xi_i,\eta_i]^T [D][B(\xi_i,\eta_i)] \det[J(\xi_i,\eta_i)]$$

که در آن nnp تعداد نقاط هامر است.

برنامه کامپیوتری LST_PLANE_STRESS_MESH.m

برای ارزیابی عملکرد تیر طره را بررسی می کنیم. این برنامه شامل ایجاد مش خودکار تابع *T6_mesh.m* و نیز یک تابع دیگر *prepare_contour_data.m* است که داده های تنشی برای ترسیم طرح المان با استفاده از Contour در برنامه matlab را آماده می سازد.

```
% THIS PROGRAM USES A 6-NODE LINEAR TRIANGULAR ELEMENT FOR
THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
% IT INCLUDES AN AUTOMATIC MESH GENERATION
%
% Make these variables global so they can be shared by other
functions
%
clear all
clc
global nnd nel nne nodof eldof n
global connec geom dee nf Nodal_loads XIG YIG
global Length Width NXE NYE X_origin Y_origin
%
format long g

%
%
% To change the size of the problem or change elastic
properties
% supply another input file
%
Length = 60.; % Length of the model
Width =20.; % Width
NXE = 12; % Number of rows in the x direction
NYE = 5; % Number of rows in the y direction
XIG = zeros(2*NXE+1,1); YIG=zeros(2*NYE+1,1); % Vectors
holding grid coordinates
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = Width/2. ; % Y origin of the global coordinate
system
%
nne = 6;
nodof = 2;
eldof = nne*nodof;
%
```

```
T6_mesh ; % Generate the mesh
%
% Material
%
E = 200000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 5.; % Beam thickness in mm
nhp = 3; % Number of sampling points
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
% Restrain in all directions the nodes situated @
% (x = Length)
%
for i=1:nnd
if geom(i,1) == Length;
nf(i,:) = [0 0];
end
end
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply an equivalent nodal load of (Pressure*thick*dhx) to
the central
% node located at x=0 and y = 0.
```

```
%
Force = 1000.; % N

for i=1:nnd
if geom(i,1) == 0. && geom(i,2) == 0.
Nodal_loads(i,:) = [0. -Force];
end
end
%
%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Assembly of the global stiffness matrix
%
%
% Form the matrix containing the abscissas and the weights of
Hammer points
%
samp=hammer(nhp);
%
% initialize the global stiffness matrix to zero
%
kk = zeros(n, n);
%
for i=1:nel
[coord,g] = elem_T6(i); % Form strain matrix, and steering
vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
matrix to zero
for ig = 1:nhp
wi = samp(ig,3);
[der,fun] = fmT6_quad(samp, ig);
jac = der*coord;
d = det(jac);
jac1=inv(jac); % Compute inverse of the Jacobian
```

```
deriv=jac1*der; % Derivative of shape functions in global
coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*bee'*dee*bee; % Integrate stiffness matrix
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
node_disp(i,:) =[x_disp y_disp];

end
%
%
% Retrieve the x_coord and y_disp of the nodes located on the
neutral axis
%
k = 0;
for i=1:nnd;
if geom(i,2)== 0.
k=k+1;
x_coord(k) = geom(i,1);
vertical_disp(k)=node_disp(i,2);
end
end
%
nhp = 1; % Calculate stresses at the centroid of the element
samp=hammer(nhp);
%
```

```
for i=1:nel
[coord,g] = elem_T6(i); % Retrieve coordinates and steering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
eld(m)=delta(g(m)); % Retrieve element displacement from the
% global displacement vector
end
end
%
for ig=1: nhp
[der,fun] = fmT6_quad(samp, ig); % Derivative of shape
functions in
% local coordinates
jac=der*coord; % Compute Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions
% in global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
eps=bee*eld; % Compute strains
sigma=dee*eps ; % Compute stresses
end % Compute stresses
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
% Prepare stresses for plotting
%
[ZX, ZY, ZT, Z1, Z2]=prepare_contour_data(SIGMA);
%
% Plot mesh using patches
%
% patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',hsv(nel), ...
'Facecolor','none','Marker','o');
%
% Plot stresses in the x_direction
%
[C,h]= contourf(XIG,YIG,ZX,40);
%clabel(C,h);
colorbar plottools;
```

T6_mesh.m

```
% This function generates a mesh of the linear strain
triangular element
%
global nnd nel geom connec XIG YIG
global Length Width NXE NYE X_origin Y_origin dhx dhy

%
nnd = 0;
k = 0;
for i = 1:NXE
for j=1:NYE
k = k + 1;
n1 = (2*j-1) + (2*i-2)*(2*NYE+1) ;
n2 = (2*j-1) + (2*i-1)*(2*NYE+1);
n3 = (2*j-1) + (2*i)*(2*NYE+1);
n4 = n1 + 1;
n5 = n2 + 1;
n6 = n3 + 1 ;
n7 = n1 + 2;
n8 = n2 + 2;
n9 = n3 + 2;
%
geom(n1,:) = [(i-1)*dhx - X_origin (j-1)*dhy - Y_origin];
geom(n2,:) = [((2*i-1)/2)*dhx - X_origin (j-1)*dhy - Y_origin
];
geom(n3,:) = [i*dhx - X_origin (j-1)*dhy - Y_origin ];
geom(n4,:) = [(i-1)*dhx - X_origin ((2*j-1)/2)*dhy - Y_origin
];
geom(n5,:) = [((2*i-1)/2)*dhx - X_origin ((2*j-1)/2)*dhy -
Y_origin ];
geom(n6,:) = [i*dhx - X_origin ((2*j-1)/2)*dhy - Y_origin ];
geom(n7,:) = [(i-1)*dhx - X_origin j*dhy - Y_origin];
geom(n8,:) = [((2*i-1)/2)*dhx - X_origin j*dhy - Y_origin];
geom(n9,:) = [i*dhx - X_origin j*dhy - Y_origin];
%
nel = 2*k;
m = nel -1;
connec(m,:) = [n1 n2 n3 n5 n7 n4];
connec(nel,:) = [n3 n6 n9 n8 n7 n5];
max_n = max([n1 n2 n3 n4 n5 n6 n7 n8 n9]);
if(nnd <= max_n); nnd = max_n; end;
%
% XIN and YIN are two vectors that holds the coordinates X
and Y
% of the grid necessary for the function contourf (XIN,YIN,
stress)
```

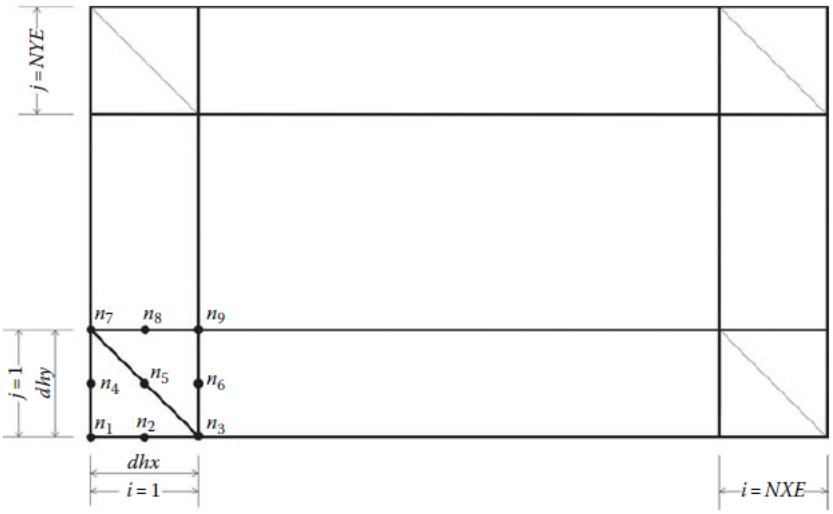
```
%
XIG(2*i-1) = geom(n1,1); XIG(2*i) = geom(n2,1); XIG(2*i+1) =
geom(n3,1);
YIG(2*j-1) = geom(n1,2); YIG(2*j) = geom(n4,2); YIG(2*j+1) =
geom(n7,2);
end
end
%
```

متغیرهای NXE و NYE به ترتیب نشان دهنده تعداد فواصل در طول جهت های x,y هستند که در شکل نشان داده شده اند. برای هر فاصله i و j نه گره $n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8 \quad n_9$ و دو المان ایجاد می شود. اولین المان دارای گره های $n_1, n_2, n_3, n_5, n_7, n_4,$ درحالیکه المان دوم گره های $n_3, n_6, n_9, n_8, n_7, n_5$ را دارد. درکل تعداد المان ها و گره های ایجاد شده به ترتیب برابر است با:

$$NEL = 2 \times NXE \times NYE,$$

$$nnd = (2 \times NXE + 1) \times (2 \times NYE + 1)$$

همچنین ایم مدول ماتریس های $geom\,(nnd, 2)$ و $connec\,(nel, nne)$ و نیز دو بردار $XIG\,(2 \times NXE + 1)$و $YIG\,(2 \times NYE + 1)$ را بدست می دهد که مختصات مش دارند.از آنها برای ترسیم کانتور با استفاده از تابع contourf در MATLAB استفاده می شود.



انتگرال گیری عددی از ماتریس سختی

ماتریس سختی اینگونه محاسبه می شود

$$[K_e] = t\sum_{i=1}^{nhp} W_i[B(\xi_i,\eta_i]^T[D][B(\xi_i,\eta_i)]\det[J(\xi_i,\eta_i)]$$

۱)برای هر دو المان داریم nel تا i=1

۲)مختصات گره های آن **coord** (**nne**, **2**) و بردار راهنمای آن **g** (**eldof**) با استفاده از تابع *elem_t6.m* نوشته می شود.

elem_T6.m

```
function[coord,g] = elem_T6(i)
%
% This function returns the coordinates of the nodes of
element i
% and its steering vector
%
global nnd nel nne nodof eldof n
global geom connec dee nf load
%
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_T6
```

۳)مقدار آغازین ماتریس سختی صفر است.

الف) روی نقطاط هامر nhpتا ig=1 حلقه ایجاد می شود.

ب)وزن wi به صورت **samp** (**ig**, **3**) بدست می آید.

ج)برای محاسبه توابع شکلی بردار fun و مشتق های محلی آن der در مختصات محلی **ξ =** **samp (ig, 1)** و **η = samp (ig, 2)** از تابع *fmT6_quad.m* استفاده می شود.

fmT6_quad.m

```
function[der,fun] = fmT6_quad(samp, ig)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta at the gauss points
for
% an 8-nodded quadrilateral
%
xi=samp(ig,1);
eta=samp(ig,2);
lambda = 1. - xi - eta;
%
fun(1) = -lambda*(1.-2*lambda);
fun(2) = 4.*xi*lambda;
fun(3) = -xi*(1.-2*xi);
fun(4) = 4.*xi*eta;
fun(5) = -eta*(1.-2*eta);
fun(6) = 4.*eta*lambda;
%
der(1,1)=1.-4*lambda; der(1,2)=4.*(lambda-xi);
der(1,3)=-1.+4*xi; der(1,4)=4.*eta;
der(1,5)=0.; der(1,6)=-4.*eta;
%
der(2,1)=1.-4*lambda; der(2,2)=-4.*xi;
der(2,3)=0.; der(2,4)=4.*xi;
der(2,5)=-1.+4.*eta; der(2,6)=4.*(lambda-eta);
%
% end function fmT6_quad
```

د)ژاکوبین محاسبه می گردد **jac = der * coord**

ه)دترمینان ژاکوبین استفاده می گردد **d = det (jac)**

ر)معکوس ژاکوبین محاسبه می گردد. **jac1 = inv (jac)**

و)مشتقات توابع شکلی با توجه به مختصات کلی x,y محاسبه می شود **deriv = jac1 * der**

ن)برای بدست آوردن ماتریس کرنشی bee از تابع formbee.m استفاده می شود.

formbee.m

```
function[bee] = formbee(deriv,nne,eldof)
%
% This function assembles the matrix [bee] from the
% derivatives of the shape functions in global coordinates
%
bee=zeros(3,eldof);
for m=1:nne
k=2*m;
l=k-1;
x=deriv(1,m);
bee(1,l)=x;
bee(3,k)=x;
y=deriv(2,m);

bee(2,k)=y;
bee(3,l)=y;
end
%
% End function formbee
```

ی)ماتریس سختی را بصورت **ke = ke + d * thick * wi * bee_ * dee * bee** محاسبه می شود.

۴) ماتریس سختی ke را به صورت kk اسمبل می کنیم.

ابعاد افقی و وزن های فرمول هامر در جدول آمده و از فرمول hammer.m بدست می آیند.

hammer.m

```
function[samp]=hammer(npt)
%
% This function returns the abscissae and weights of the
% integration points for npt equal up to 7
%
%
samp=zeros(npt,3);
%
if npt==1
samp=[1/3. 1/3. 1/2.];
elseif (npt==2 | npt==3)
npt=3;
```

```
samp=[1/6. 1/6. 1/6.; ...
2/3 1./6 1/6.; ...

1/6. 2./3. 1/6];
elseif (npt==4 | npt==5)
npt=4;
samp= [1/3 1/3 -27/96; ...
1/5 1/5. 25/96;...
3/5 1/5. 25/96;...
1/5 3/5 25/96];
elseif npt==6
a = 0.445948490915965; b = 0.091576213509771;
samp= [ a a 0.111690794839005; ...
1-2*a a 0.111690794839005; ...
a 1-2*a 0.111690794839005; ...
b b 0.054975871827661; ...
1-2*b b 0.054975871827661; ...
b 1-2*b 0.054975871827661];
elseif npt==7
a = (6+sqrt(15))/21 ; b = 4/7 -a;
A = (155+sqrt(15))/2400; B = (31/240 -A);
samp= [ 1/3 1/3 9/80; ...
a a A ; ...
1-2*a a A ; ...
a 1-2*a A ; ...
b b B ; ...
1-2*b b B ; ...
b 1-2*b B];
end
%
```

```
% End function hammer
```

TABLE 8.2
Abscissae and Weights for a Triangle

Order m	Number of Points r	ξ	η	W
1	1	0.333333333333	0.333333333333	0.5
2	3	0.5	0.5	0.166666666666
		0	0.5	0.166666666666
		0.5	0	0.166666666666
2	3	0.166666666666	0.166666666666	0.166666666666
		0.666666666666	0.166666666666	0.166666666666
		0.166666666666	0.666666666666	0.166666666666
3	4	0.333333333333	0.333333333333	−0.28125
		0.2	0.2	0.260416666666
		0.6	0.2	0.260416666666
		0.2	0.6	0.260416666666
4	6	0.44594849092	0.44594849092	0.111690794839
		0.10810301817	0.44594849092	0.111690794839
		0.44594849092	0.10810301817	0.111690794839
		0.09157621351	0.09157621351	0.054975871827
		0.81684757289	0.09157621351	0.054975871827
		0.09157621351	0.81684757289	0.054975871827
5	7	0.33333333333	0.33333333333	0.1125
		0.470142064105	0.470142064105	0.066197076394
		0.05971587179	0.470142064105	0.066197076394
		0.470142064105	0.05971587179	0.066197076394
		0.101286507324	0.101286507324	0.708802923606
		0.898713492676	0.101286507324	0.708802923606
		0.101286507324	0.898713492676	0.708802923606

محاسبه تنش ها و کرنش ها

وقتی سیستم کلی معادلات حل شد تنش هارا در مرکز المان ها محاسبه خواهیم کرد. برای این منظور این تساوی را درنظر میگیریم nhp=1 برای هر المان:

۱)مختصات گره های المان **coord** *(***nne**, **2** *)* و بردار راهنمای آن *(***eldof***)* **g** را با استفاده از تابع *elem_t*6.*m* بدست می آوریم.

elem_T6.m

```
function[coord,g] = elem_T6(i)
%
% This function returns the coordinates of the nodes of
element i
% and its steering vector
%
global nnd nel nne nodof eldof n
global geom connec dee nf load
%
```

```
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_T6
```

۲)با استفاده از بردار کلی جابجایی ها **delta** (**n**) جابجایی های گرهی المان (**eldof**) **eld** را بدست می اوریم.

الف)بر روی نقاط هامر nhp تا ig=1 حلقه می سازیم.

ب)برای محاسبه توابع شکلی بردار fun و مشتقات محلی آنها der در مختصات مکانی ξ = **samp** (**ig**, **1**) و η = **samp** (**ig**, **2**) از تابع *fmT6_quad.m* استفاده می کنیم.

fmT6_quad.m

```
function[der,fun] = fmT6_quad(samp, ig)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta at the gauss points
for
% an 8-nodded quadrilateral
%
xi=samp(ig,1);
eta=samp(ig,2);
lambda = 1. - xi - eta;
%
fun(1) = -lambda*(1.-2*lambda);
fun(2) = 4.*xi*lambda;
fun(3) = -xi*(1.-2*xi);
fun(4) = 4.*xi*eta;
fun(5) = -eta*(1.-2*eta);
fun(6) = 4.*eta*lambda;
%
der(1,1)=1.-4*lambda; der(1,2)=4.*(lambda-xi);
der(1,3)=-1.+4*xi; der(1,4)=4.*eta;
der(1,5)=0.; der(1,6)=-4.*eta;
```

```
%
der(2,1)=1.-4*lambda; der(2,2)=-4.*xi;
der(2,3)=0.; der(2,4)=4.*xi;
der(2,5)=-1.+4.*eta; der(2,6)=4.*(lambda-eta);
%
% end function fmT6_quad
```

ج) ژاکوبین را محاسبه می کنیم **jac = der * coord**

د)دترمینان ژاکوبین را محاسبه می کنیم **d = det (jac)**

ر)معکوس ژاکوبین محاسبه می گردد. **jac1 = inv (jac)**

و)مشتقات توابع شکلی با توجه به مختصات کلی x,y محاسبه می شود **deriv = jac1 * der**

ن)برای بدست آوردن ماتریس کرنشی bee از تابع *formbee.m* استفاده می شود.

formbee.m

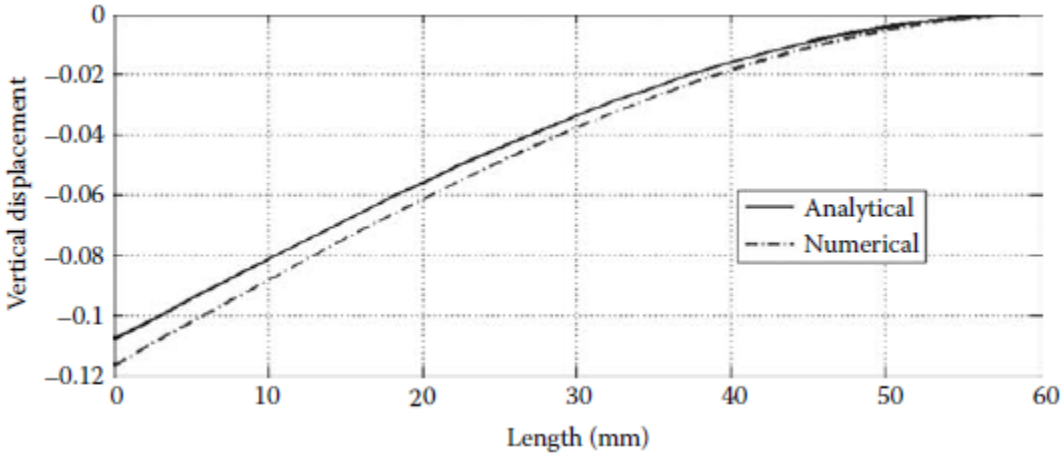
```
function[bee] = formbee(deriv,nne,eldof)
%
% This function assembles the matrix [bee] from the
% derivatives of the shape functions in global coordinates
%
bee=zeros(3,eldof);
for m=1:nne
k=2*m;
l=k-1;
x=deriv(1,m);
bee(1,l)=x;
bee(3,k)=x;
y=deriv(2,m);

bee(2,k)=y;
bee(3,l)=y;
end
%
% End function formbee
```

و) کرنش را بصورت **eps = bee * eld** بدست می آوریم.

ی)تنش هارا بصورت **sigma = dee * eps** محاسبه می کنیم.

۳)تنش هارا در ماتریس **SIGMA** (**nel, 3**) دخیره می کنیم.

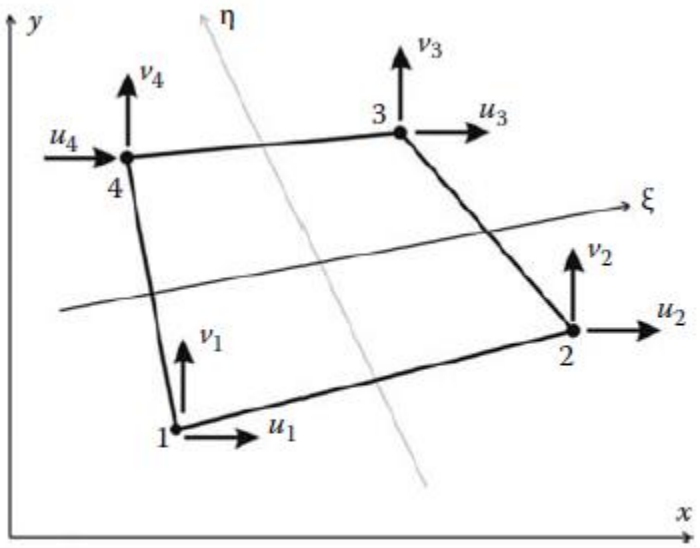
با توجه به نمودار می توان مشاهده کرد که نتایج به نتایج تحلیلی بسیار نزدیک است.



المان چهارضلعی Q4

چهارگرهی کرنش خطی ۴ گره و حاشیه مستقیم دارد. توابع شکل آن بدین صورت می باشد.

$$N_1(\xi,\eta) = 0.25(1-\xi-\eta+\xi\eta)$$
$$N_2(\xi,\eta) = 0.25(1+\xi-\eta-\xi\eta)$$
$$N_3(\xi,\eta) = 0.25(1+\xi+\eta+\xi\eta)$$
$$N_4(\xi,\eta) = 0.25(1-\xi+\eta-\xi\eta)$$



Linear quadrilateral element.

حوزه جابجایی

تقریب حوزه جابجایی در این المان اینگونه بدست می آید

$$u = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4$$
$$v = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4$$

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \left[\begin{array}{cc|cc|cc|cc} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{array}\right] \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$

$$\{U\} = [N]\{a\}$$

این المان ایزوپارامتریک است. بنابراین توابع شکل آن $Ni(\xi, \eta)$ تبدیل هندسی بین المان مرجع و المان مادر را نیز تعریف می کند. مختصات x,y هر نقطه از المان مادر اینگونه نوشته می شود

$$x = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4$$
$$y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4$$

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}$$

$$[J] = \frac{1}{4} \begin{bmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\xi) & -(1+\xi) & (1+\xi) & (1-\xi) \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}$$

ماتریس کرنشی

از جایگذاری جابجایی های u,v بردار کرنشی اینگونه بدست می آید

$$\{\epsilon\} = [B]\{a\}$$

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}$$

برای محاسبه ماتریس B لازم است رابطه مشتقات جزئی در مختصات (x,y) با مختصات محلی (ξ, η) مشخص می شود. مشتقات توابع شکلی با استفاده از قانون زنجیره ای را می توان به شکل زیر نوشت

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi}$$
$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta}$$

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

مشتقات توابع شکلی در سیستم (x,y) با معکوس کردن معادله قبلی بدست می آید.

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

ماتریس سختی

$$[K_e] = \left[\int_{A_e} [B]^T [D][B] t \, dA \right]$$

انتگرال گیری از حجم با استفاده از روش مربع سازی گاوس محاسبه می شود.

$$[K_e] = t \int_{-1}^{+1} \int_{-1}^{+1} [B(\xi,\eta]^T [D][B(\xi,\eta)] det[J(\xi,\eta)] d\eta \, d\xi$$

$$= t \sum_{i=1}^{ngp} \sum_{j=1}^{ngp} W_i W_j [B(\xi_i,\eta_j]^T [D][B(\xi_i,\eta_j)] det[J(\xi_i,\eta_j)]$$

که در آن t ضخامت و ngp تعداد نقاط گاوس می باشد. برای انتگرال گیری دقیق المان در هر جهت ۲ نقطه گاوس لازم است.

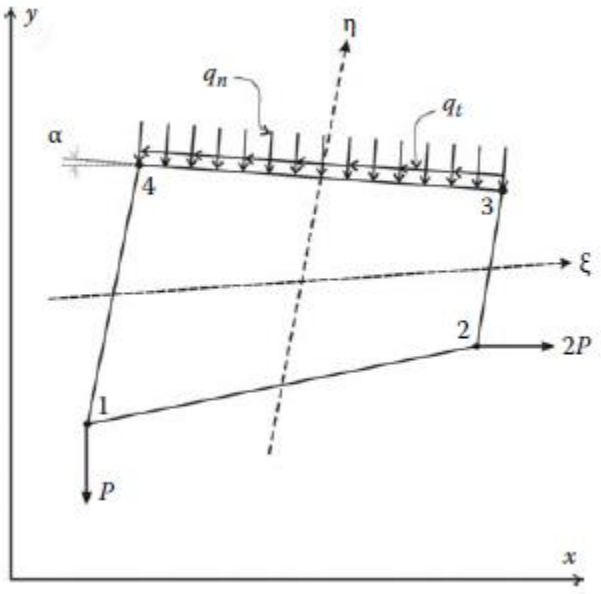
بردار نیروی المان

$$\{f_e\} = \int_{A_e} [N]^T \{b\} t \, dA + \int_{L_e} [N]^T \{t\} t \, dl + \sum_i [N_{([x]=[\bar{x}])}]^T \{P\}_i$$

با توجه به اینکه نیروهای جسم b بعلت جاذبه هستند قسمت اول معادله بالا با روش گاوس محاسبه می شود.

$$\int_{A_e} [N]^T \{b\} t \, dA = t \sum_{i=1}^{ngp} \sum_{j=1}^{ngp} W_i W_j [N(\xi_i,\eta_j]^T \begin{Bmatrix} 0 \\ -\rho g \end{Bmatrix} det[J(\xi_i,\eta_j)]$$

برای محاسبه عبارتهای دوم و سوم معادله بالا بهتر است با نمونه ای مثل آنچه در شکل نشان داده شده کار را ادامه دهیم. المان در وجه ۳-۴ در معرض تنش سطحی q قرار دارد که دارای جز نرمال *qn* و جز تانژانتی *qt* و نیز دو نیروی متمرکز به بزرگی pو2p می باشد که بترتیب بر گره های ۱و۲ وارد می شوند.



برای آنکه بتوان عبارت دوم سمت راست معادله را محاسبه کرد که به تنش سطحی مربوط می شود لازم است برای علامت ها قرارداد تعریف کنیم. وقتی گره های یک المان خلاف جهت عقربه های ساعت شماره گذاری شوند نیروی تانژانتی مثبت و اگر در خلاف جهت عقربه های ساعت وارد شود یک نیروی نرمال مثبت است اگر به سمت داخل المان وارد شود.

بارها به شکب زیر نوشته می شوند.

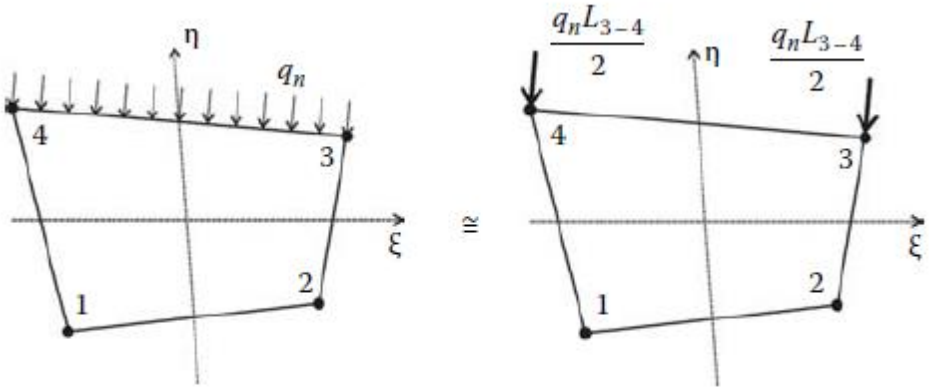
$$q_x = q_t dL\cos\alpha - q_n dL\sin\alpha = q_t dx - q_n dy$$
$$q_y = q_n dL\cos\alpha + q_t dL\sin\alpha = q_n dx + q_t dy$$

چون انتگرال گیری در طول وجه $(-1,+1)$ انجام می شود پس تغییر متغیر های زیر $dx = \frac{\partial x}{\partial \xi} d\xi$ and $dy = \frac{\partial y}{\partial \eta} d\eta$ صحیح است.

$$q_x = \left(q_t \frac{\partial x}{\partial \xi} - q_n \frac{\partial y}{\partial \xi}\right) d\xi$$
$$q_y = \left(q_n \frac{\partial x}{\partial \xi} + q_t \frac{\partial y}{\partial \xi}\right) d\xi$$

$$\int_{A_e} [N]^T \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} dA = t \int_{L_{3-4}} [N(\xi, +1)]^T \begin{Bmatrix} q_x \\ q_y \end{Bmatrix} dl$$

$$= t \sum_{i=1}^{ngp} W_i [N(\xi_i, +1)]^T \begin{Bmatrix} \left(q_t \dfrac{\partial x(\xi_i, +1)}{\partial \xi} - q_n \dfrac{\partial y(\xi_i, +1)}{\partial \xi} \right) \\ \left(q_n \dfrac{\partial x(\xi_i, +1)}{\partial \xi} + q_t \dfrac{\partial y(\xi_i, +1)}{\partial \xi} \right) \end{Bmatrix}$$



Equivalent nodal loading.

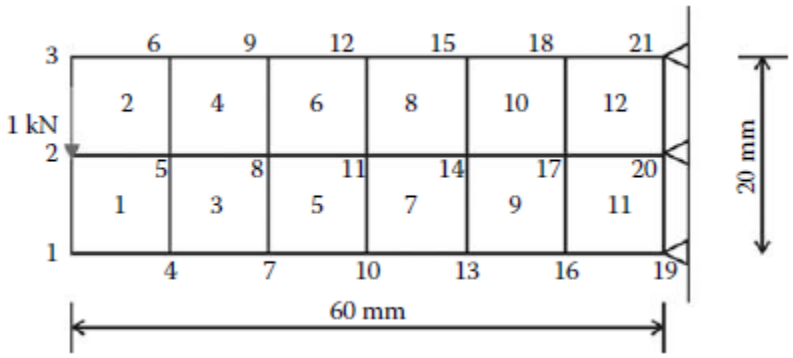
در گره ۱ داریم $N4 = 0,\ N3 = 0,\ N2 = 0,\ N1 = 1,$ و در گره ۲ داریم $N1 = 0,\ N2 = 1,$

$N3 = 0,\ N4 = 0.$.

$$\sum_{k=1} [N]_{x=x_k} \{P_k\} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} 0 \\ -P \end{Bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} 2P \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -P \\ 2P \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

برنامه کامپیوتری Q4_PLANE_STRESS.m

این برنامه مشابه برنامه قبلی خود یعنی CST_PLANE_STRESS.m می باشد به استثنا اینکه ماتریس سختی با استفاده از انتگرال گیری عددی به روش مربعات گاوس انجام می شود. بخاطر درجات آزادی بیشتر اندازه برخی خطوط افزایش یافته است. به منظور ارزیابی عملکرد المان مجددا تیر طره را تحلیل می کنیم. برای گسسته سازی از ۱۲ المان استفاده خواهیم کرد. گره های ۱۹و۲۰و۲۱ انتهای ثابت را نشان می دهند.



آماده سازی داده ها

برای خواندن داده ها از m-file بنام **Q4_COARSE_MESH_DATA.m** استفاده می کنیم.

Q4_COARSE_MESH_DATA.m

```
% File: Q4_COARSE_MESH_DATA
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
%
% To change the size of the mesh, alter the next statements
%
nnd = 21 ; % Number of nodes:
nel = 12; % Number of elements:
nne = 4 ; % Number of nodes per element:
nodof =2; % Number of degrees of freedom per node
ngp = 2 % number of Gauss points
eldof = nne*nodof; % Number of degrees of freedom per element
```

```
%
%
% Nodes coordinates x and y
geom = [0, -10.0; ... % x and y coordinates of node 1
0.0 0.0; ... % x and y coordinates of node 2
0.0 10.0; ... % x and y coordinates of node 3
10.0 -10.0; ... % x and y coordinates of node 4
10.0 0.0; ... % x and y coordinates of node 5
10.0 10.0; ... % x and y coordinates of node 6
20.0 -10.0; ... % x and y coordinates of node 7
20.0 0.0; ... % x and y coordinates of node 8
20.0 10.0; ... % x and y coordinates of node 9
30.0 -10.0; ... % x and y coordinates of node 10
30.0 0.0; ... % x and y coordinates of node 11
30.0 10.0; ... % x and y coordinates of node 12
40.0 -10.0; ... % x and y coordinates of node 13
40.0 0.0; ... % x and y coordinates of node 14
40.0 10.0; ... % x and y coordinates of node 15
50.0 -10.0; ... % x and y coordinates of node 16
50.0 0.0; ... % x and y coordinates of node 17
50.0 10.0; ... % x and y coordinates of node 18
60.0 -10.0; ... % x and y coordinates of node 19
60.0 0.0; ... % x and y coordinates of node 20
60.0 10.0]; % x and y coordinates of node 21
%
%
%
disp ('Nodes X-Y coordinates')
geom
%
% Element connectivity
connec= [ 1 4 5 2 ;... % Element 1
2 5 6 3 ;... % Element 2
4 7 8 5 ;... % Element 3
5 8 9 6 ;... % Element 4
7 10 11 8 ;... % Element 5
8 11 12 9 ;... % Element 6
10 13 14 11 ;... % Element 7
11 14 15 12 ;... % Element 8

13 16 17 14 ;... % Element 9
14 17 18 15 ;... % Element 10
16 19 20 17 ;... % Element 11
17 20 21 18]; % Element 12
%
%
disp ('Elements connectivity')
```

```
connec
%
E = 200000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 5.; % Beam thickness in mm
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
nf(19,:) = [0 0]; % Node 19 is restrained in the x and y
directions
nf(20,:) = [0 0]; % Node 20 is restrained in the x and y
directions
nf(21,:) = [0 0]; % Node 21 is restrained in the x and y
directions
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply a concentrated at the node having x = 0, and y = 0.
%
Force = 1000.; % N
%
Nodal_loads(1,:) = [0. -Force];
```

داده های ورودی برای این تیر عبارتند از

nnd=۲۱ تعداد گره ها

nel =۱۲تعداد المانها

nne =۴تعداد گره ها در واحد المان

nodof=۲ تعداد درجات آزادی در واحد گره

مختصات x,y به شکل ماتریس . **geom** (**nnd**, **2**) نوشته می شود.اتصال المان در ماتریس **connec** (**nel**, **4**). نوشته می شود. توجه داشته باشید که شماره گذاری داخلی گره ها برخلاف جهت عقربه های ساعت است. درجات آزادی برای گره های ثابت با ۰ نشان داده می شود. به درجات آزادی سایر گره ها که آزاد هستند عدد ۱ اختصاص می یابد. اطلاعات شرایط مرزی در ماتریس **nf** (**nnd**, **nodof**) نوشته می شوند. نیروی متمرکز ۱۰۰۰ نیوتنی بر گره ۲ وارد می شود. این نیرو در برنامه اصلی در بردار نیروی کلی fg اسمبل می شود.

برنامه اصلی Q4_PLANE_STRESS.m

```
% THIS PROGRAM USES AN 4-NODDED QUADRILATERAL ELEMENT FOR THE
LINEAR ELASTIC
% STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
%
% Make these variables global so they can be shared by other
functions
%
clc
clear all
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
%
format long g
%
% To change the size of the problem or change the elastic
properties
% supply another input file
%
Q4_COARSE_MESH_DATA
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Form the matrix containing the abscissas and the weights of
Gauss points
%
ngp = 2;
samp=gauss(ngp);
%
% Numerical integration and assembly of the global stiffness
matrix
%
% initialize the global stiffness matrix to zero
kk = zeros(n, n);
%
for i=1:nel
[coord,g] = elem_q4(i) ; % coordinates of the nodes of
element i,
% and its steering vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
matrix
% to zero
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmlin(samp, ig,jg); % Derivative of shape
functions
%in local coordinates
jac=der*coord; % Compute Jacobian matrix
d=det(jac); % Compute determinant of Jacobian
% matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions
```

```
% in global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*wj*bee'*dee*bee; % Integrate stiffness
matrix
end
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
disp('node x_disp y_disp ') %
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
disp([i x_disp y_disp]) % Display displacements of each node
DISP(i,:) = [x_disp y_disp]
end
%
%
ngp=1; % Calculate stresses and strains at
%the center of each element
samp=gauss(ngp);
%
for i=1:nel
[coord,g] = elem_q4(i); % coordinates of the nodes of element
i,
% and its steering vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
```

```
eld(m)=delta(g(m)); % Retrieve element displacement from the
% global displacement vector
end
end
%
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmlin(samp, ig,jg); % Derivative of shape
functions
% in local coordinates
jac=der*coord; % Compute Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions
% in global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
eps=bee*eld % Compute strains
sigma=dee*eps % Compute stresses
end
end
SIGMA(i,:)=sigma ; % Store stresses for all elements

%
% Average stresses at nodes
%
[ZX, ZY, ZT, Z1, Z2]=stresses_at_nodes_Q4(SIGMA);
%
%
% Plot stresses in the x_direction
%
U2 = DISP(:,2);
cmin = min(U2);
cmax = max(U2);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',U2,...
'Facecolor','interp','Marker','.');
colorbar;
```

انتگرال گیری ماتریس سختی

۱)برای هر المان nel تا i=1

۲)مختصات گره های ان بصورت **coord (nne, 2)** نوشته می شود و بردار راهنمای ان **g (eldof)** با استفاده از تابع elem_Q4.m بدست می آید.

elem_q4.m

```
function[coord,g] = elem_q4(i)
%
% This function returns the coordinates of the nodes of
% element i and its steering vector g
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf load
%
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_q4
```

۳)در ابتدا ماتریس سختی را معادل صفر در نظر میگیریم.

الف) روی نقطه گاوس jg=1-ngp حلقه می سازیم.

ب) وزن wi را بصورت **samp (ig, 2)** می نویسیم.

I روی نقطه گاوس jg=1-ngp حلقه می سازیم.

II وزن wj را بصورت **samp (jg, 2)** می نویسیم.

III برای محاسبه توابع شکلی بردار fun و مشتقات آنها ماتریس der در مختصات مکانی

samp (ig, 1) = ξ و

samp (ig, 1) =η از تابع *fmlin.m* استفاده می کنیم.

fmlin.m

```
function[der,fun] = fmlin(samp, ig,jg)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta
%
xi=samp(ig,1);
eta=samp(jg,1);
%
fun = 0.25*[(1.- xi - eta + xi*eta);...
(1.+ xi - eta - xi*eta);...
(1.+ xi + eta + xi*eta);...
(1.- xi + eta - xi*eta)];
%
der = 0.25*[-(1-eta) (1-eta) (1+eta) -(1+eta);...
-(1-xi) -(1+xi) (1+xi) (1-xi)];
% end function fmlin
```

IV ژاکوبین **jac = der * coord** را محاسبه می کنیم.

V دترمینان ژاکوبین را به صورت **d = det (jac)** محاسبه می کنیم.

VI معکوس ژاکوبین را به صورت **jac1 = inv (jac)** بدست می آوریم.

VII مشتق توابع شکلی را با توجه به مختصات کلی x,y به صورت **deriv = jac1 * der** بدست می آوریم.

VIII برای بدست آوردن ماتریس کرنشی bee از تابع *formbee.m* استفاده می کنیم.

formbee.m

```
function[bee] = formbee(deriv,nne,eldof)
%
% This function assembles the matrix [bee] from the
% derivatives of the shape functions in global coordinates
%
```

```
bee=zeros(3,eldof);
for m=1:nne
k=2*m;
l=k-1;
x=deriv(1,m);
bee(1,l)=x;
bee(3,k)=x;
y=deriv(2,m);

bee(2,k)=y;
bee(3,l)=y;
end
%
% End function formbee
```

IX ماتریس سختی را اینگونه محاسبه می کنیم

ke = ke + d * thick * wi * wj * bee_ * dee * bee

۴)ماتریس سختی ke را در ماتریس کلی kk اسمبل می کنیم.

محاسبه ماتریس سختی نیازمند استفاده از روش مربع سازی گاوس است. برای انجام این کار ابعاد افقی و وزن متناظر با نقاط گاوس باید در اختیار برنامه قرار گیرد. این اطلاعات در خط **samp (ngp, 2)** وجود دارند که به صورت زیر نوشته می شوند.

$\xi i = samp(i, 1)$ and $Wi = samp(i, 2)$

تابع **gauss.m** در پایین امده و تا ngp=4 قابل استفاده است. تابع **elem_q4.m** نیز مختصات گره های هر المان و نیز تابع راهنما آن .textbfg را برمی گرداند. تابع **fmlin.m** هم تابع شکلی بردار fun و مشتقات آن ها ماتریس der در مختصات کانونی را بدست می دهد.

gauss.m

```
function[samp]=gauss(ngp)
%
% This function returns the abscissas and weights of the
Gauss
% points for ngp equal up to 4
%
%
```

```
samp=zeros(ngp,2);
%
if ngp==1
samp=[0. 2];
elseif ngp==2
samp=[-1./sqrt(3) 1.;...
1./sqrt(3) 1.];
elseif ngp==3
samp= [-.2*sqrt(15.) 5./9; ...
0. 8./9.;...
.2*sqrt(15.) 5./9];
elseif ngp==4
samp= [-0.861136311594053 0.347854845137454; ...
-0.339981043584856 0.652145154862546; ...
0.339981043584856 0.652145154862546; ...
0.861136311594053 0.347854845137454];
end
%
% End function Gauss
```

elem_q4.m

```
function[coord,g] = elem_q4(i)
%
% This function returns the coordinates of the nodes of
% element i and its steering vector g
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf load
%
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_q4
```

fmlin.m

```
function[der,fun] = fmlin(samp, ig,jg)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta
%
```

```
xi=samp(ig,1);
eta=samp(jg,1);
%
fun = 0.25*[(1.- xi - eta + xi*eta);...
(1.+ xi - eta - xi*eta);...
(1.+ xi + eta + xi*eta);...
(1.- xi + eta - xi*eta)];
%
der = 0.25*[-(1-eta) (1-eta) (1+eta) -(1+eta);...
-(1-xi) -(1+xi) (1+xi) (1-xi)];
% end function fmlin
```

محاسبه تنش ها و کرنش ها

وقتی سیستم کلی معادلات حل شد کرنش ها در مرکز المان را محاسبه می کنیم. برای این منظور تساوی ngp=1 را برقرار می دانیم.

۱)برای هر المان

۲)مختصات گره های ان ***(nne, 2)*** **coord** و بردار راهنمای آن ***(eldof)*** **g** را با استفاده از تابع *elem_Q4.m* می نویسیم.

elem_q4.m

```
function[coord,g] = elem_q4(i)
%
% This function returns the coordinates of the nodes of
% element i and its steering vector g
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf load
%
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_q4
```

۳)جابجایی گرهی ان **eldof** (**eld** را از بردار کلی جابجایی ها **delta** (**n** بدست می آوریم.

I روی نقاط گاوسی jg=ngp-1 حلقه می سازیم.

II روی نقاط گاوسی jp=1-ngp حلقه می سازیم.

III برای محاسبه توابع شکلی بردار fun و مشتقات محلی ان ها der در مختصات محلی

ξ و **samp** (**ig**, **1**) **=**

η از تابع *fmlin.m* استفاده می کنیم. **samp** (**jg**, **1**) **=**

fmlin.m

```
function[der,fun] = fmlin(samp, ig,jg)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta
%
xi=samp(ig,1);
eta=samp(jg,1);
%
fun = 0.25*[(1.- xi - eta + xi*eta);...
(1.+ xi - eta - xi*eta);...
(1.+ xi + eta + xi*eta);...
(1.- xi + eta - xi*eta)];
%
der = 0.25*[-(1-eta) (1-eta) (1+eta) -(1+eta);...
-(1-xi) -(1+xi) (1+xi) (1-xi)];
% end function fmlin
```

IV ژاکوبین **jac = der * coord** را محاسبه می کنیم.

V دترمینان ژاکوبین را به صورت **d = det** (**jac**) بدست می آوریم.

VI معکوس ژاکوبین را بصورت **jac1 = inv** (**jac**) بدست می آوریم.

VII مشتقات اشکال تابعی را با توجه به مختصات کلی x,y به صورت **deriv = jac1 * der** بدست می آوریم.

VIII برای نوشتن ماتریس کرنشی bee از تابع *formbee.m* استفاده می کنیم.

formbee.m

```
function[bee] = formbee(deriv,nne,eldof)
%
% This function assembles the matrix [bee] from the
% derivatives of the shape functions in global coordinates
%
bee=zeros(3,eldof);
for m=1:nne
k=2*m;
l=k-1;
x=deriv(1,m);
bee(1,l)=x;
bee(3,k)=x;
y=deriv(2,m);

bee(2,k)=y;
bee(3,l)=y;
end
%
% End function formbee
```

IX کرنش هارا به صورت **eps = bee * eld** محاسبه می کنیم.

X تنش هارا بصورت **sigma = dee * eps** محاسبه می کنیم.

۴)تنش هارا در ماتریس **SIGMA** *(**nel**, **3**)* دخیره می کنیم.

تنش های محاسبه شده در مراکزالمان ها در گره ها با استفاده از تابع **Stresses_at_nodes_Q4.m** محاسبه می شود.

Stresses_at_nodes_Q4.m

```
function[ZX, ZY, ZT, Z1, Z2]=Stresses_at_nodes_Q4(SIGMA)
%
% This function averages the stresses at the nodes
%
```

```
%
global nnd nel nne geom connec XIG YIG NXE NYE
%
for k = 1:nnd
sigx = 0. ;sigy = 0.; tau = 0.;
ne = 0;
for iel = 1:nel;
for jel=1:nne;
if connec(iel,jel) == k;
ne=ne+1;
sigx = sigx+SIGMA(iel,1);
sigy = sigy + SIGMA(iel,2);
tau = tau + SIGMA(iel,3);
end
end
end
ZX(k,1) = sigx/ne;
ZY(k,1) = sigy/ne;

ZT(k,1)=tau/ne;
Z1(k,1)= ((sigx+sigy)/2 + sqrt(((sigx+sigy)/2)^2 +tau^2))/ne;
Z2(k,1)= ((sigx+sigy)/2 - sqrt(((sigx+sigy)/2)^2 +tau^2))/ne;
End
```

برنامه ایجاد شبکه خودکار

برای مدلسازی بهتر به ایجاد شبکه و اصلاح ان نیاز داریم. در برنامه زیر شبکه بطور خودکار با استفاده از تابع **Q4_mesh.m** ایجاد می گردد. این تابع اتصالات المان ها و ماتریس های هندسی گره ای را ایجاد می کند که بعد از برنامه اصلی می آید.

```
% THIS PROGRAM USES AN 4-NODDED QUADRILATERAL ELEMENT FOR THE
LINEAR ELASTIC
% STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
%
% Make these variables global so they can be shared by other
functions
%
clc
clear all
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin dhx dhy
%
format long g
```

```
%
% To change the size of the mesh, alter the next statements
%
Length = 60.; % Length of the model
Width =20.; % Width
NXE = 24; % Number of rows in the x direction
NYE = 8; % Number of rows in the y direction
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = Width/2. ; % Y origin of the global coordinate
system
%
nne = 4;
nodof = 2;
eldof = nne*nodof;
%
Q4_mesh % Generate the mesh
%
E = 200000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 5.; % Beam thickness in mm
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
%
% Boundary conditions

%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
% Restrain in all directions the nodes situated @
% (x = Length)
%
for i=1:nnd
if geom(i,1) == Length;
nf(i,:) = [0 0];
end
end
%
% Counting of the free degrees of freedom
%
n=0; for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
```

```
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply a concentrated at the node having x = 0, and y = 0.
%
Force = 1000.; % N
%
for i=1:nnd
if geom(i,1) == 0. && geom(i,2) == 0.
Nodal_loads(i,:) = [0. -Force];
end
end
%
%%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Form the matrix containing the abscissas and the weights of
Gauss points
%
ngp = 2;
samp=gauss(ngp);
%
% Numerical integration and assembly of the global stiffness
matrix
%
% initialize the global stiffness matrix to zero
kk = zeros(n, n);
```

```
%
for i=1:nel
[coord,g] = elem_q4(i) ; % coordinates of the nodes of
element i,
% and its steering vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
% matrix to zero

wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmlin(samp, ig,jg); % Derivative of shape
functions
% in local coordinates
jac=der*coord; % Compute Jacobian matrix
d=det(jac); % Compute determinant of Jacobian
% matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in
% global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*wj*bee'*dee*bee; % Integrate stiffness
matrix
end
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
disp('node x_disp y_disp ') %
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
```

```
end
disp([i x_disp y_disp]) % Display displacements of each node
DISP(i,:) = [x_disp y_disp]
end
%
%
ngp=1; % Calculate stresses and strains at
%the center of each element
samp=gauss(ngp);
%
for i=1:nel
[coord,g] = elem_q4(i); % coordinates of the nodes of element
i,
% and its steering vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
eld(m)=delta(g(m)); % Retrieve element displacement from the
% global displacement vector
end
end
%
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmlin(samp, ig,jg); % Derivative of shape
functions in
% local coordinates
jac=der*coord; % Compute Jacobian matrix

jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in
% global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
eps=bee*eld % Compute strains
sigma=dee*eps % Compute stresses
end
end
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
% Average stresses at nodes
%
[ZX, ZY, ZT, Z1, Z2]=stresses_at_nodes_Q4(SIGMA);
```

```
%
%
% Plot stresses in the x_direction
%
U2 = DISP(:,2);
cmin = min(U2);
cmax = max(U2);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',U2, ...
'Facecolor','interp','Marker','.');
colorbar;
```

Q4_mesh.m

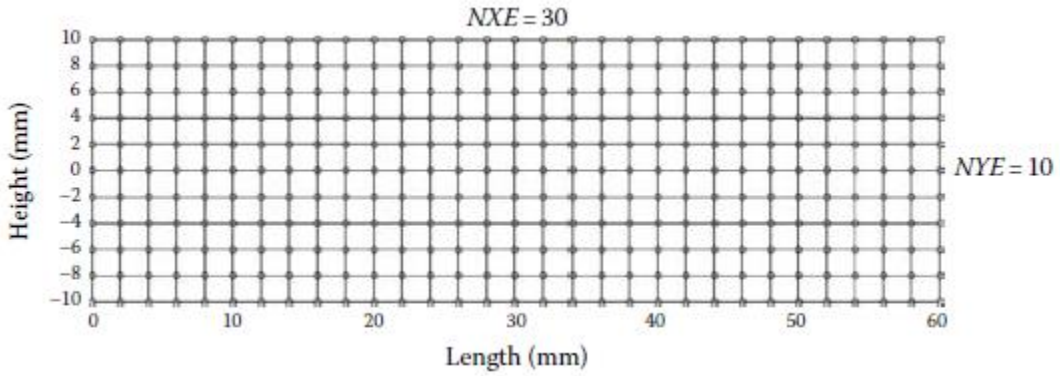
```
% This module generates a mesh of linear quadrilateral
elements
%
global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin dhx dhy
%
%
nnd = 0;
k = 0;
for i = 1:NXE
for j=1:NYE
k = k + 1;
n1 = j + (i-1)*(NYE + 1);
geom(n1,:) = [(i-1)*dhx - X_origin (j-1)*dhy - Y_origin ];
n2 = j + i*(NYE+1);
geom(n2,:) = [i*dhx - X_origin (j-1)*dhy - Y_origin ];
n3 = n1 + 1;
geom(n3,:) = [(i-1)*dhx - X_origin j*dhy - Y_origin ];
n4 = n2 + 1;
geom(n4,:) = [i*dhx- X_origin j*dhy - Y_origin ];
nel = k;
connec(nel,:) = [n1 n2 n4 n3];
nnd = n4;
end
end
```

متغیر های NXE و NYE بترتیب تعداد فواصل در زول جهت های X و Y را نشان می دهند. برای هر فاصله i و j ۴ گره $n1, n2, n3, ,n4$ و ۱ المان ایجاد می شوند. این المان گره های $n1, n2, n3, ,n4$ را داراست. در کل تعداد المان ها و گره های ایجاد شده بترتیب برابرند با:

$$nel = NXE \times NYE$$

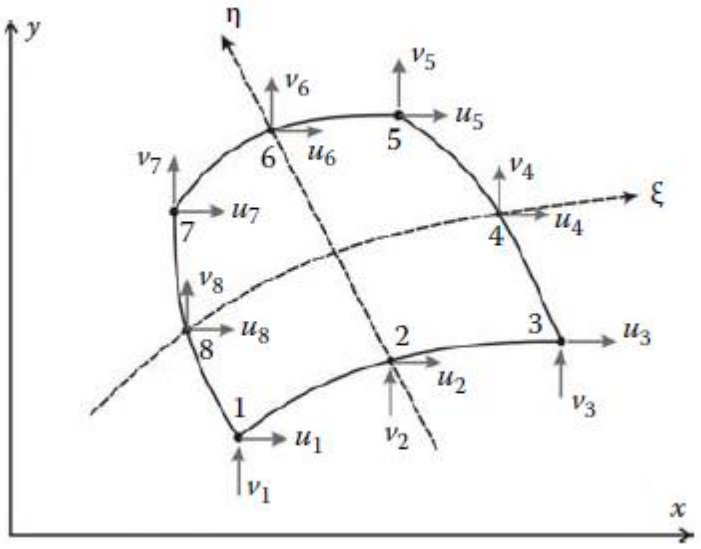
$$nnd = (NXE+1) \times (NYE+1)$$

همچنین این مدول ماتریس های $connec(nel, nne)$, $geom(nnd, 2)$ را برمی گرداند.نتایج بدست آمده از مش ریز باطبع دقیق تر هستند و بسیار شبیه به نتایج بدست امده از مثلث کرنش خطی ست.



چهار ضلعی ۸ گرهیQ8

المان چهرضلعی ۸ گرهی وجه های خمیده ای دارد که استفاده از ان برای مدلسازی سازه هایی با حاشیه های خم را سودمند می سازد. توابع شکل این المان بدین صورت نوشته می شود.



$$\begin{Bmatrix} N_1(\xi,\eta) \\ N_2(\xi,\eta) \\ N_3(\xi,\eta) \\ N_4(\xi,\eta) \\ N_5(\xi,\eta) \\ N_6(\xi,\eta) \\ N_7(\xi,\eta) \\ N_8(\xi,\eta) \end{Bmatrix} = \begin{Bmatrix} -0.25(1-\xi)(1-\eta)(1+\xi+\eta) \\ 0.50(1-\xi^2)(1-\eta) \\ -0.25(1+\xi)(1-\eta)(1-\xi+\eta) \\ 0.50(1+\xi)(1-\eta^2) \\ -0.25(1+\xi)(1+\eta)(1-\xi-\eta) \\ 0.50(1-\xi^2)(1+\eta) \\ -0.25(1-\xi)(1+\eta)(1+\xi-\eta) \\ 0.50(1-\xi)(1-\eta^2) \end{Bmatrix}$$

حوزه جابجایی برای این المان به صورت زیر تقریب زده می شود.

$$u = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4 + N_5u_5 + N_6u_6 + N_7u_7 + N_8u_8$$

$$v = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4 + N_5v_5 + N_6v_6 + N_7v_7 + N_8v_8$$

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & | & N_2 & 0 & | & \dots & \dots & | & N_8 & 0 \\ 0 & N_1 & | & 0 & N_2 & | & \dots & \dots & | & 0 & N_8 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ u_8 \\ v_8 \end{Bmatrix}$$

$$\{U\} = [N]\{a\}$$

این المان ایزوپارامتریک است. بنابراین توابع شکل $N_i(\xi,\eta)$ نیز تبدیل هندسی بین المان مرجع و مادر را تعریف می کنند. مختصات x,y هر نقطه از المان مادر اینگونه نوشته می شود که:

$$x = N_1x_1 + N_2x_2 + \cdots + N_8x_8$$
$$y = N_1y_1 + N_2y_2 + \cdots + N_8y_8$$

ژاکوبین تبدیل بدین صورت بدست می آید:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{8} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{8} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{8} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}$$

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \cdots & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \cdots & \frac{\partial N_8}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_8 & y_8 \end{bmatrix}$$

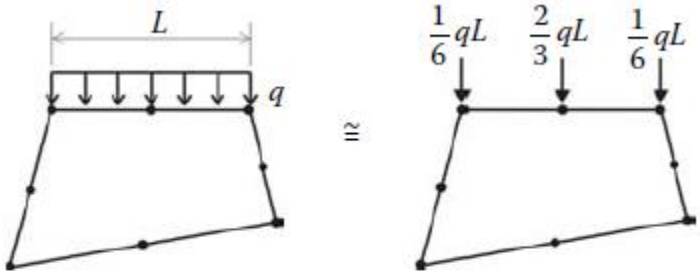
ماتریس کرنشی B به صورت زیر بدست می آید:

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \cdots & \cdots & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \cdots & \cdots & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \cdots & \cdots & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}$$

ماتریس سختی هم به همان روش المان چهار ضلعی خطی بدست می آید با این ابعاد که ابعاد آن ۱۶*۱۶ هستند چون به ازای هر المان ۱۶ درجه آزادی وجود دارد.

نیروهای گرهی معادل

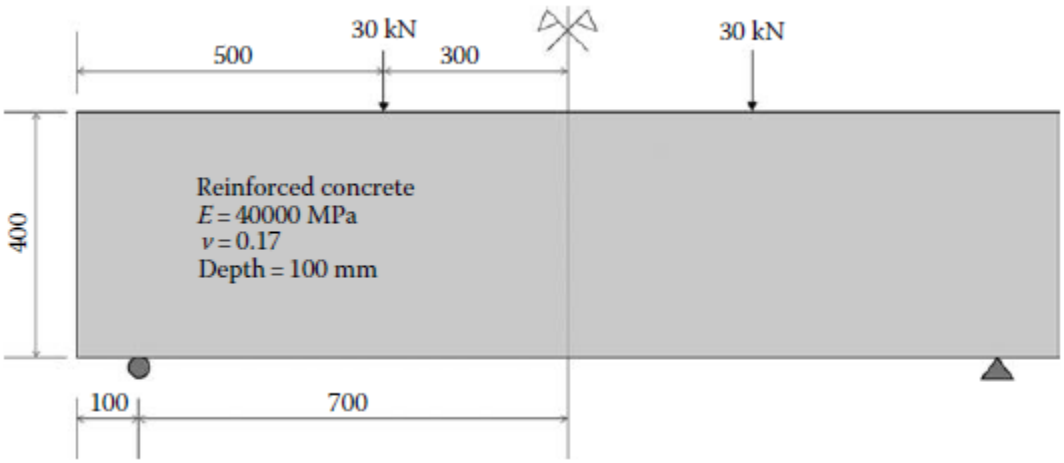
اگر شکل بارگذاری روی لبه المان پیچیده باشد باید از فرآیند انتگرال گیری که جزئیات آن قبلا شرح داده شد مورد استفاده قرار گیرد. اما اگر توزیع بار یکنواخت باشدمی توان از بار معادل گرهی مطابق شکل زیر استفاده کرد.

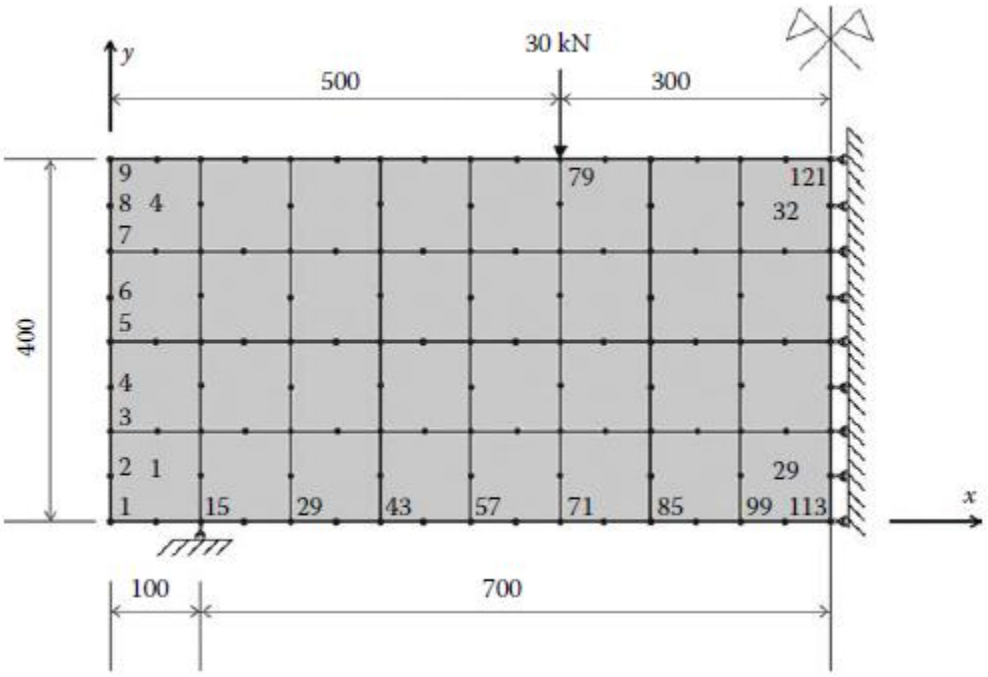


Equivalent nodal loads.

برنامه Q8_PLANE_STRESS.m

این برنامه در واقع مشابه نمونه قبل خود یعنی Q4_PLANE_STRESS.m است با این تفاوت که برخی خطوط ابعاد کمی بزرگی دارند که بخاطر افزایش درجه آزادی در هر المان است. برای ارزیابی عملکرد المان تیر عمیق با تکیه گاه ساده ای را تحلیل می کنیم که در معرض خمش ۴ نقطه ای است. با بهره گیری از تقارن فقط نیمی ازین مدل تحلیل می شود. برای گسسته سازی از ۳۲ المان استفاده می کنیم. گره های ۱۲۱-۱۱۳ نشان دهنده حوزه میانی هستند. این گره ها می توانند فقط به صورت عمودی جابجا شوند.





Q8_COARSE_MESH_DATA.m

```
%
%%%%%%%%%%%%%%%%%%%%%%%% Beginning of data input
%%%%%%%%%%%%%%%%%%%%%%%%%%
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
%
nnd = 121 ; % Number of nodes:

nel = 32; % Number of elements:
nne = 8 ; % Number of nodes per element:
```

```
nodof =2; % Number of degrees of freedom per node
ngp = 2 % number of Gauss points
eldof = nne*nodof; % Number of degrees of freedom per element
%
% Thickness of the domain
thick = 100.
%
% Nodes coordinates x and y
%
geom = [ 0 0 ; ... % x and y coordinates of node 1
0 50 ; ...
0 100 ; ...
0 150 ; ...
0 200 ; ...
0 250 ; ...
0 300 ; ...
0 350 ; ...
0 400 ; ...
50 0 ; ...
50 100 ; ...
50 200 ; ...
50 300 ; ...
50 400 ; ...
100 0 ; ...
100 50 ; ...
100 100 ; ...
100 150 ; ...
100 200 ; ...
100 250 ; ...
100 300 ; ...
100 350 ; ...
100 400 ; ...
150 0 ; ...
150 100 ; ...
150 200 ; ...
150 300 ; ...
150 400 ; ...
200 0 ; ...
200 50 ; ...

200 100 ; ...
200 150 ; ...
200 200 ; ...
200 250 ; ...
200 300 ; ...
200 350 ; ...
200 400 ; ...
```

```
250 0 ; ...
250 100 ; ...
250 200 ; ...
250 300 ; ...
250 400 ; ...
300 0 ; ...
300 50 ; ...
300 100 ; ...
300 150 ; ...
300 200 ; ...
300 250 ; ...
300 300 ; ...
300 350 ; ...
300 400 ; ...
350 0 ; ...
350 100 ; ...
350 200 ; ...
350 300 ; ...
350 400 ; ...
400 0 ; ...
400 50 ; ...
400 100 ; ...
400 150 ; ...
400 200 ; ...
400 250 ; ...
400 300 ; ...
400 350 ; ...
400 400 ; ...
450 0 ; ...
450 100 ; ...
450 200 ; ...
450 300 ; ...
450 400 ; ...
500 0 ; ...
500 50 ; ...
500 100 ; ...
500 150 ; ...
500 200 ; ...
500 250 ; ...
500 300 ; ...
500 350 ; ...
500 400 ; ...
550 0 ; ...
550 100 ; ...
550 200 ; ...
550 300 ; ...
550 400 ; ...
```

```
600 0 ; ...
600 50 ; ...
600 100 ; ...
600 150 ; ...
600 200 ; ...
600 250 ; ...
600 300 ; ...
600 350 ; ...
600 400 ; ...
650 0 ; ...
650 100 ; ...
650 200 ; ...
650 300 ; ...

650 400 ; ...
700 0 ; ...
700 50 ; ...
700 100 ; ...
700 150 ; ...
700 200 ; ...
700 250 ; ...
700 300 ; ...
700 350 ; ...
700 400 ; ...
750 0 ; ...
750 100 ; ...
750 200 ; ...
750 300 ; ...
750 400 ; ...
800 0 ; ...
800 50 ; ...
800 100 ; ...
800 150 ; ...
800 200 ; ...
800 250 ; ...
800 300 ; ...
800 350 ; ...
800 400] ; % x and y coordinates of node 121
%
% Element connectivity
%
connec = [1 10 15 16 17 11 3 2 ; ... % Element 1
3 11 17 18 19 12 5 4 ; ...
5 12 19 20 21 13 7 6 ; ...
7 13 21 22 23 14 9 8 ; ...
15 24 29 30 31 25 17 16 ; ...
17 25 31 32 33 26 19 18 ; ...
```

```
19 26 33 34 35 27 21 20 ; ...
21 27 35 36 37 28 23 22 ; ...
29 38 43 44 45 39 31 30 ; ...
31 39 45 46 47 40 33 32 ; ...
33 40 47 48 49 41 35 34 ; ...
35 41 49 50 51 42 37 36 ; ...
43 52 57 58 59 53 45 44 ; ...
45 53 59 60 61 54 47 46 ; ...
47 54 61 62 63 55 49 48 ; ...
49 55 63 64 65 56 51 50 ; ...
57 66 71 72 73 67 59 58 ; ...
59 67 73 74 75 68 61 60 ; ...
61 68 75 76 77 69 63 62 ; ...
63 69 77 78 79 70 65 64 ; ...
71 80 85 86 87 81 73 72 ; ...
73 81 87 88 89 82 75 74 ; ...
75 82 89 90 91 83 77 76 ; ...
77 83 91 92 93 84 79 78 ; ...
85 94 99 100 101 95 87 86 ; ...
87 95 101 102 103 96 89 88 ; ...
89 96 103 104 105 97 91 90 ; ...
91 97 105 106 107 98 93 92 ; ...
99 108 113 114 115 109 101 100 ; ...
101 109 115 116 117 110 103 102 ; ...
103 110 117 118 119 111 105 104 ; ...
105 111 119 120 121 112 107 106 ]; % Element 32
%
% Material properties
%
E=40000; vu=0.17; % Young's modulus and Poisson's ratio
%
% Form the matrix of elastic properties

dee=formdsig(E,vu); % Matrix of elastic properties for plane
stress
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
for i=1:nnd
if geom(i,1) == 800.;
nf(i,:) = [0 1];
end
if geom(i,1) == 100. && geom(i,2) == 0.;
nf(i,:) = [1 0];
end
```

```
end
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
disp ('Nodal freedom')
nf
disp ('Total number of active degrees of freedom')
n
%
% loading
%
Nodal_loads = zeros(nnd, 2);
Nodal_loads(79,2)=-30000.; % Vertical load on node 79
%
% End input
```

داده های ورودی برای این تیر عبارتند از:

Nnd=121 تعداد گره ها

Nel=32 تعداد المان ها

Nne=8 تعداد گره ها در واحد المان

Nodof=2 تعداد درجات آزادی در واحد گره

مختصات x,y گره ها به شکل ماتریسی . **geom**(**nnd**, **2**) نوشته می شوند. اتصال المان در ماتریس **connec**(**nel**, **8**) . نوشته می شود. باید توجه داشت که شماره گذاری ذاخلی گره ها در خلاف جهت عقربه های ساعت است.

همانطور که در شکل مشاهده می کنیم گره های ۱۲۱-۱۱۳ فقط در جهت x ثابت هستند. گره ۱۵ نسان دهنده تکیه گاه ساده است و فقط در جهت y ثابت است. اطلاعات شرایط مرزی در ماتریس **nf**(**nnd**, **nodof**) . نوشته می شوند. نیروی متمرکز ۳۰۰۰۰ نیوتنی فقط بر گره ۷۹ وارد می شود. در برنامه اصلی این نیرو در بردار نیروی کلی اسمبل می شود.

برنامه اصلی

Q8_PLANE_STRESS.m

```
% THIS PROGRAM USES AN 8-NODDED QUADRILATERAL ELEMENT FOR THE
LINEAR ELASTIC
% STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM

%
% Make these variables global so they can be shared by other
functions
%
clc
clear all
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
%
format long g
%
% This is where the to input the data in the form of a file
with
% an extension .m
%
Q8_coarse_mesh_data
%
%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
```

```
%
% Form the matrix containing the abscissas and the weights of
Gauss points
%
samp=gauss(ngp);
%
% Numerical integration and assembly of the global stiffness
matrix
%
% initialize the global stiffness matrix to zero
kk = zeros(n, n);
%
for i=1:nel
[coord,g] = elem_q8(i) ; % coordinates of the nodes of
element i,
% and its steering vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
% matrix to zero
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmquad(samp, ig,jg); % Derivative of shape
functions
% in local coordinates
jac=der*coord; % Compute Jacobian matrix
d=det(jac); % Compute determinant of Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in
% global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*wj*bee'*dee*bee; % Integrate stiffness
matrix
end
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%

delta = kk\fg ; % solve for unknown displacements
disp('node x_disp y_disp ') %
for i=1: nnd %
```

```
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
disp([i x_disp y_disp]) ; % Display displacements of each
node
DISP(i,:) = [x_disp y_disp];
end
%
%
ngp=1; % Calculate stresses and strains at
% the center of each element
samp=gauss(ngp);
%
for i=1:nel
[coord,g] = elem_q8(i); % coordinates of the nodes of element
i, and its steering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
eld(m)=delta(g(m)); % Retrieve element displacement from the
global displacement
vector
end
end
%
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmquad(samp, ig,jg); % Derivative of shape
functions in local coordinates
jac=der*coord; % Compute Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in global
coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
```

```
eps=bee*eld % Compute strains
sigma=dee*eps % Compute stresses
end
end
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
%
[ZX, ZY, ZT, Z1, Z2]=stresses_at_nodes_Q8(SIGMA);
U2 = DISP(:,2);
%
%
% Choose one the quantities ( U2, ZX, ZY, ZT, Z1, Z2) to plot
%
cmin = min(ZT);
cmax = max(ZT);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',ZT,...
'Facecolor','interp','Marker','.');
colorbar;
```

انتگرال گیری ماتریس سختی

محاسبه ماتریس سختی این المان نیز مانند چهارضلعی خطی انجام می شود بجز اینکه تابع elem_Q8.m جایگزین elem_Q4.m و تابع fmquad.m جایگزین fmlin.m می شوند.

انتگرال گیری دقیق ماتریس سختی نیازمند ۳ نقطه گاوس در هر جهت است.

Elem_q8.m

```
function[coord,g] = elem_q8(i)
%
% This function returns the coordinates of the nodes of
element i
% and its steering vector
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf load
%
l=0;
coord=zeros(nne,nodof);
for k=1: nne
for j=1:nodof
coord(k,j)=geom(connec(i,k),j);
```

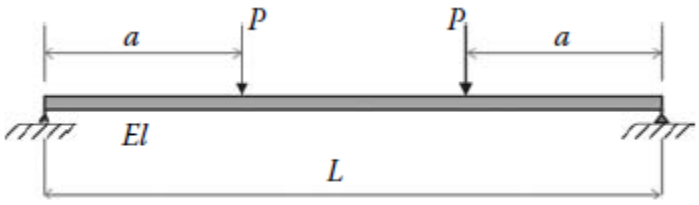
```
l=l+1;
g(l)=nf(connec(i,k),j);
end
end
%
% End function elem_q8
```

fmquad.m

```
function[der,fun] = fmquad(samp, ig,jg)
%
% This function returns the vector of the shape function and
their
% derivatives with respect to xi and eta at the gauss points
for
% an 8-nodded quadrilateral
%
xi=samp(ig,1);
eta=samp(jg,1);
etam=(1.-eta);
etap=(1.+eta);
xim=(1.-xi);
xip=(1.+xi);
%
fun(1) = -0.25*xim*etam*(1.+ xi + eta);
fun(2) = 0.5*(1.- xi^2)*etam;
fun(3) = -0.25*xip*etam*(1. - xi + eta);
fun(4) = 0.5*xip*(1. - eta^2);
fun(5) = -0.25*xip*etap*(1. - xi - eta);
fun(6) = 0.5*(1. - xi^2)*etap;
fun(7) = -0.25*xim*etap*(1. + xi - eta);
fun(8) = 0.5*xim*(1. - eta^2);
%
der(1,1)=0.25*etam*(2.*xi + eta); der(1,2)=-1.*etam*xi;
der(1,3)=0.25*etam*(2.*xi-eta); der(1,4)=0.5*(1-eta^2);
der(1,5)=0.25*etap*(2.*xi+eta); der(1,6)=-1.*etap*xi;
der(1,7)=0.25*etap*(2.*xi-eta); der(1,8)=-0.5*(1.-eta^2);
%
der(2,1)=0.25*xim*(2.*eta+xi); der(2,2)=-0.5*(1. - xi^2);
der(2,3)=-0.25*xip*(xi-2.*eta); der(2,4)=-1.*xip*eta;
der(2,5)=0.25*xip*(xi+2.*eta); der(2,6)=0.5*(1.-xi^2);
der(2,7)=-0.25*xim*(xi-2.*eta); der(2,8)=-1.*xim*eta;
%
% end function fmquad
```

نتایج بدست آمده با مش درشت

برای بررسی دقت این نتیجه تیر عمیق حاضر را به صورت یک تیر باریک درنظر گرفته و از تئوری تیر مهندسی برای محاسبه خمش حوزه میانی ایستفاده می کنیم. برای تیر باریک با سختی EI خمش در حوزه میانی به صورت زیر بدست می آید:



$$\delta_{max} = \frac{Pa(3L^2 - 4a^2)}{24EI}$$

طبق فرمول بالا جابجایی حوزه میانی برابر ۰.۱۲ میلی متر است که از برنامه ۰.۱۵ میلی متر بدست آمده. این تفاوت منطقی ست چون تئوری تیر مهندسی که فرمول تحلیلی بر آن مبناست خمش های برشی اضافی را که در تیرهای عمیق وجود دارد در نظر نمی گیرد. بنابراین می توان با اطمینان بیان کرد که جابجایی بدست آمده با برنامه به همان درستیست که با یک مش درشت بدست می آید.

برنامه ای برای ایجاد مش خودکار

در برنامه **Plane_Q8_MESH.m** مش بطور خودکار با مدول **Q8_mesh.m** ساخته می شود. این مدول ماتریس های اتصال المان و ماتریس های هندسی گره ای را آماده می کند و بعد از برنامه اصلی بدین صورت می آید.

Plane_Q8_mesh.m

```
% THIS PROGRAM USES AN 8-NODDED QUADRILATERAL ELEMENT FOR THE
LINEAR ELASTIC
% STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM. IT CONTAINS
AN AUTOMATIC
% MESH GENERATION MODULE Q8_mesh.m
%
```

```
% Make these variables global so they can be shared by other
functions
%
clc
clear all
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads

global Length Width NXE NYE X_origin Y_origin dhx dhy
%
format long g
%
% To change the size of the problem alter the next lines
%
%
Length = 800.; % Length of the model
Width = 400.; % Width
NXE = 32; % Number of rows in the x direction
NYE = 16; % Number of rows in the y direction
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = 0. ; % Y origin of the global coordinate system
%
nne = 8;
nodof = 2;
eldof = nne*nodof;
ngp = 2;
%
Q8_mesh % Generate the mesh
%
E = 40000.; % Elastic modulus in MPa
vu = 0.17; % Poisson's ratio
thick = 100.; % Beam thickness in mm
%
% Form the elastic matrix for plane stress
%
dee = formdsig(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
% Restrain in all directions the nodes situated @
% (x = Length)
%
```

```
for i=1:nnd
if geom(i,1) == Length;
nf(i,:) = [0 1];
end
if geom(i,1) == 100. && geom(i,2) == 0.;
nf(i,:) = [1 0];
end
end
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply a concentrated at the node having x = 0, and y = 0.

%

Force = 30000.; % N
%
for i=1:nnd
if geom(i,1) == 500. && geom(i,2) == 400.
Nodal_loads(i,:) = [0. -Force]; % Force acting in negative
% direction
end
end
%
%%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
```

```
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Form the matrix containing the abscissas and the weights of
Gauss points
%
samp=gauss(ngp);
%
% Numerical integration and assembly of the global stiffness
matrix
%
% initialize the global stiffness matrix to zero
kk = zeros(n, n);
%
for i=1:nel
[coord,g] = elem_q8(i) ; % coordinates of the nodes of
element i,
% and its steering vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
% matrix to zero
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmquad(samp, ig,jg); % Derivative of shape
functions
% in local coordinates
jac=der*coord; % Compute Jacobian matrix
d=det(jac); % Compute determinant of Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in
% global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*wj*bee'*dee*bee; % Integrate stiffness
matrix
end
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
```

```
%
delta = kk\fg ; % solve for unknown displacements
disp('node x_disp y_disp ') %
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end

%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
disp([i x_disp y_disp]) ; % Display displacements of each
node
DISP(i,:) = [x_disp y_disp];
end
%
%
ngp=1; % Calculate stresses and strains at
% the center of each element
samp=gauss(ngp);
%
for i=1:nel
[coord,g] = elem_q8(i); % coordinates of the nodes of element
i,
% and its steering vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
eld(m)=delta(g(m)); % Retrieve element displacement from the
% global displacement vector
end
end
%
for ig=1: ngp
wi = samp(ig,2);
for jg=1: ngp
wj=samp(jg,2);
[der,fun] = fmquad(samp, ig,jg); % Derivative of shape
functions in
% local coordinates
```

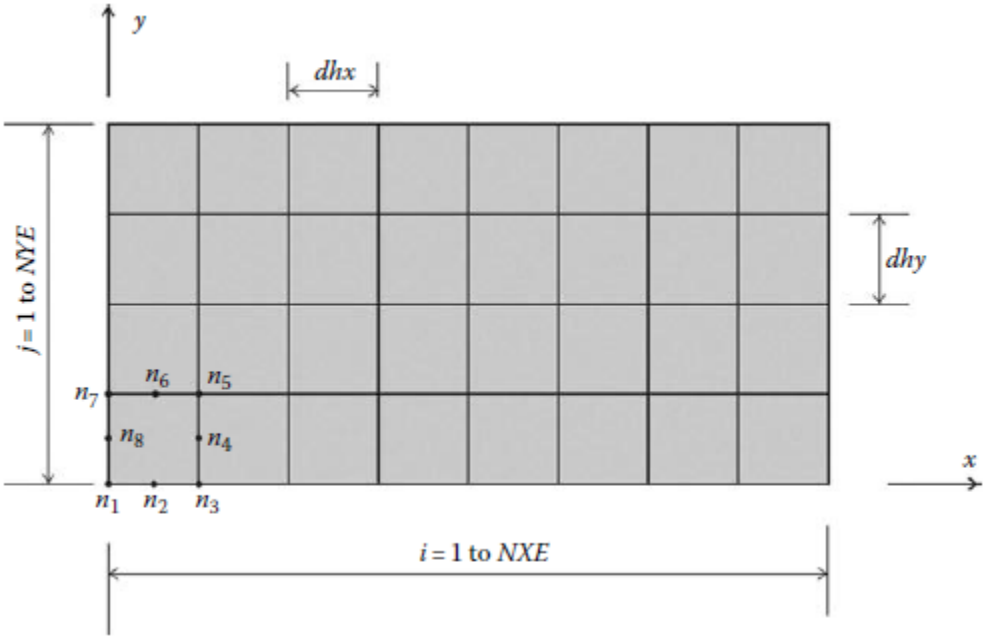
```
jac=der*coord; % Compute Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions
% in global coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
eps=bee*eld % Compute strains
sigma=dee*eps % Compute stresses
end
end
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
%
[ZX, ZY, ZT, Z1, Z2]=stresses_at_nodes_Q8(SIGMA);
%
%
% Plot stresses in the x_direction
%
U2 = DISP(:,2);
cmin = min(U2);
cmax = max(U2);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',U2,...
'Facecolor','interp','Marker','.');
colorbar;
```

Q8_mesh.m

```
% This module generates a mesh of 8-nodded quadrilateral
elements
%
global nnd nel nne nodof eldof n

global geom connec dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin dhx dhy
%
%
nnd = 0;
k = 0;
for i = 1:NXE
for j=1:NYE
k = k + 1;
%
n1 = (i-1)*(3*NYE+2)+2*j - 1;
n2 = i*(3*NYE+2)+j - NYE - 1;
n3 = i*(3*NYE+2)+2*j-1;
n4 = n3 + 1;
n5 = n3 + 2;
```

```
n6 = n2 + 1;
n7 = n1 + 2;
n8 = n1 + 1;
%
geom(n1,:) = [(i-1)*dhx - X_origin (j-1)*dhy - Y_origin ];
geom(n3,:) = [i*dhx - X_origin (j-1)*dhy - Y_origin ];
geom(n2,:) = [(geom(n1,1)+geom(n3,1))/2
(geom(n1,2)+geom(n3,2))/2];
geom(n5,:) = [i*dhx- X_origin j*dhy - Y_origin ];
geom(n4,:) = [(geom(n3,1)+ geom(n5,1))/2 (geom(n3,2)+
geom(n5,2))/2];
geom(n7,:) = [(i-1)*dhx - X_origin j*dhy - Y_origin ];
geom(n6,:) = [(geom(n5,1)+ geom(n7,1))/2 (geom(n5,2)+
geom(n7,2))/2];
geom(n8,:) = [(geom(n1,1)+ geom(n7,1))/2 (geom(n1,2)+
geom(n7,2))/2];
%
nel = k;
nnd = n5;
connec(k,:) = [n1 n2 n3 n4 n5 n6 n7 n8];
end
end
```

متغیر های NXE و NYE بترتیب تعداد فواصل در طول جهت های xو y را نشان می دهند. برای هر فاصله iو j یک المان با گره های n8 , . . . ,n1 ایجاد می گردد. این مدول ماتریس های connec(nel, nne), geom(nnd, 2) و نیز تعداد المانهای nel و تعداد گره های nnd را برمی گرداند.



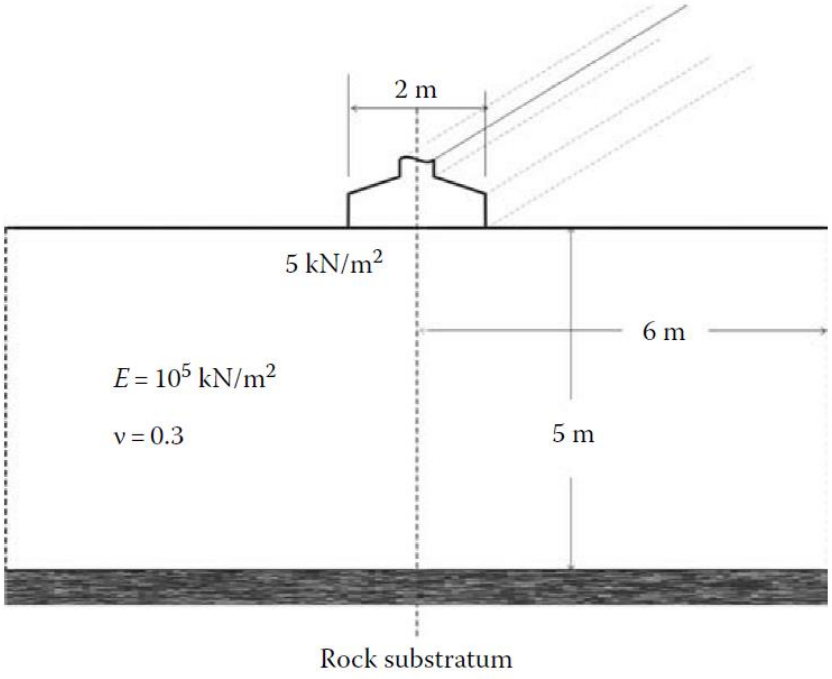
در یک مش ریز جزئیات بیشتری وجود دارد. برای مثال تمرکز تنش در نقطه اعمال بار و شکل نوار برشی.

مسائل حل شده

ستون نواری با المان CST

شکل زیر یک پایه نواری روی خاک ماسه ای را با مدول الاستیسیته $E = 105\ kN/m2$ و نسبت پواسون $\mu = 0.3$. نشان می دهد. عرض پایه پل ۲ متر بوده و تحت بار یکنواخت $5\ kN/m2$. قرار دارد. ۵ متر پایین تر از پایه خاک از سنگ صلبی ساخته شده که می توان آنرا صلب فرض کرد.

بعلاوه فرض کنید ۶ متر دورتر از مرکز پایه جابجایی افقی خاک قابل صرف نظر باشد. با در نظر گرفتن اینکه طول المان ۰.۵ متر باشد پایه را با استفاده از المان های CST,LST تحلیل می کنیم.



پایه نواری محدود یک صلب سه بعدی ست اما جهت طولی بسیار مهم است که در نتیجه از تغییر ضخامت جلوگیری می کند. از حرکت فونداسیون نواری در جهت z ممانعت به عمل می آید. بنابراین جابجایی w صفر است. با توجه به تقارن در میانه پایه نیز باید w صفر باشد و فرض می کنیم جابجایی های u,v فقط توابعی از x,y باشندچنین بیانی را می توان اینگونه نشان داد که $\epsilon_{zz} = \epsilon_{xz} = \epsilon_{yz} = 0$ و این حالت کرنش صفحه ایست. برای بدست آوردن ماتریس خواص الاستیک از تابع **formdeps.m** استفاده می کنیم.بعلاوه هندسه پایه متقارن است و فقط نیمه راست آن گسسته سازی می شود. این دامنه با استفاده از ۱۲ فاصله در جهت x NXE=12 و ۱۰ فاصله در طول y NYE=10 گسسته سازی می شود. این گسسته سازی المان هایی به طول ۰.۵ در هرجهت می سازد. شرایط مرزی گره های محدود شده با استفاده از مختصات آن ها به صورت زیر بدست می آیند:

طول=x در جهت x محدود می گردند:

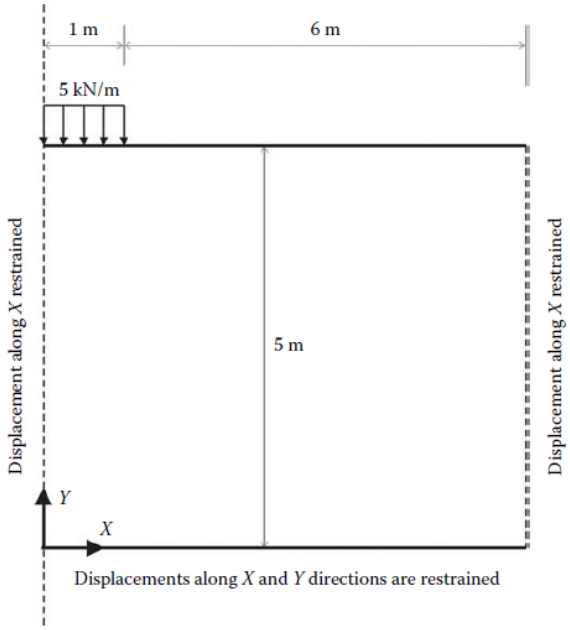
```
if geom(i,1) == 0. | geom(i,1) == Length;
nf(i,:) = [0 1];
end
```

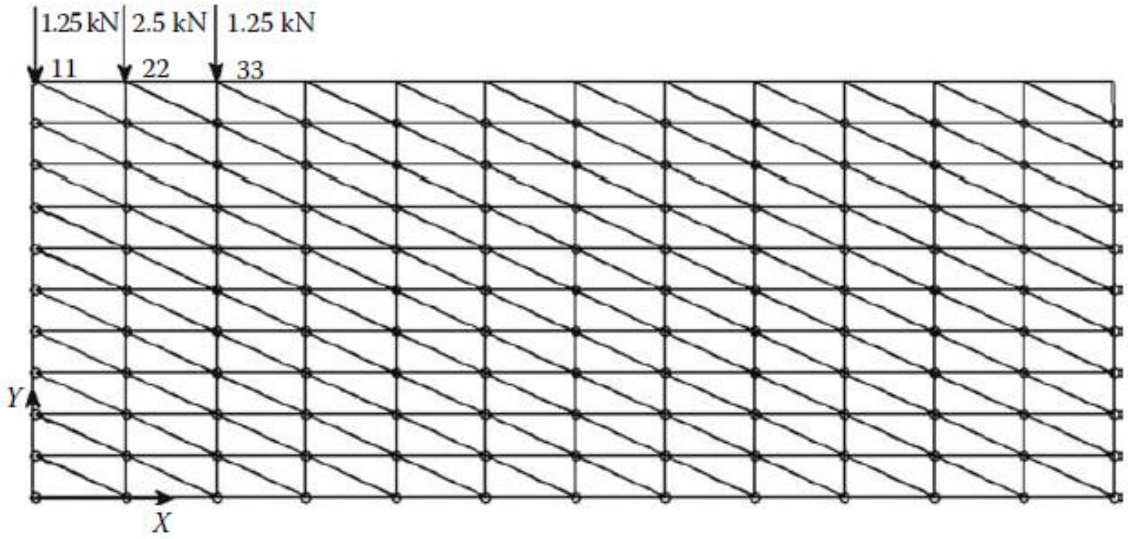
گره هایی که روی لایه سنگی قرار می گیرند y=0 در همه جهات محدود می شوند:

```
if geom(i,2) == 0. ;
nf(i,:) = [0 0];
end
```

formdeps.m

```
function[dee] = formdeps(E,vu)
%
% This function forms the elasticity matrix for a plane
strain problem
%
v1=1.-vu
c=E/((1.+vu)*(1.-2.*vu))

%
dee=[v1*c vu*c 0. ;...
vu*c v1*c 0. ;...
0. 0. .5*c*(1.-2.*vu)];
%
% end function fromdeps
```





مش بندی که تابع **T3_mesh.m** ایجاد می کند در واقع بارگذاری را نشان نمی دهد. با رگذاری به صورت دستی به شکل اضافه شده است. بارای متمرکزی که از نظر استاتیکی معادل هستند به صورت زیر بر گره ها وارد می شوند.

```
Nodal_loads(11,:) = [0. -1.25];
Nodal_loads(22,:) = [0. -2.50];
Nodal_loads(33,:) = [0. -1.25];
```

CST_STRIP_FOOTING.m

```
% THIS PROGRAM USES AN 3-NODE LINEAR TRIANGULAR ELEMENT FOR
THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
% IT INCLUDES AN AUTOMATIC MESH GENERATION
%
% Make these variables global so they can be shared by other
functions
%
clear all
clc
global nnd nel nne nodof eldof n
global geom dee nf Nodal_loads
global Length Width NXE NYE X_origin Y_origin
%
format long g
%
%
% To change the size of the problem or change elastic
properties
```

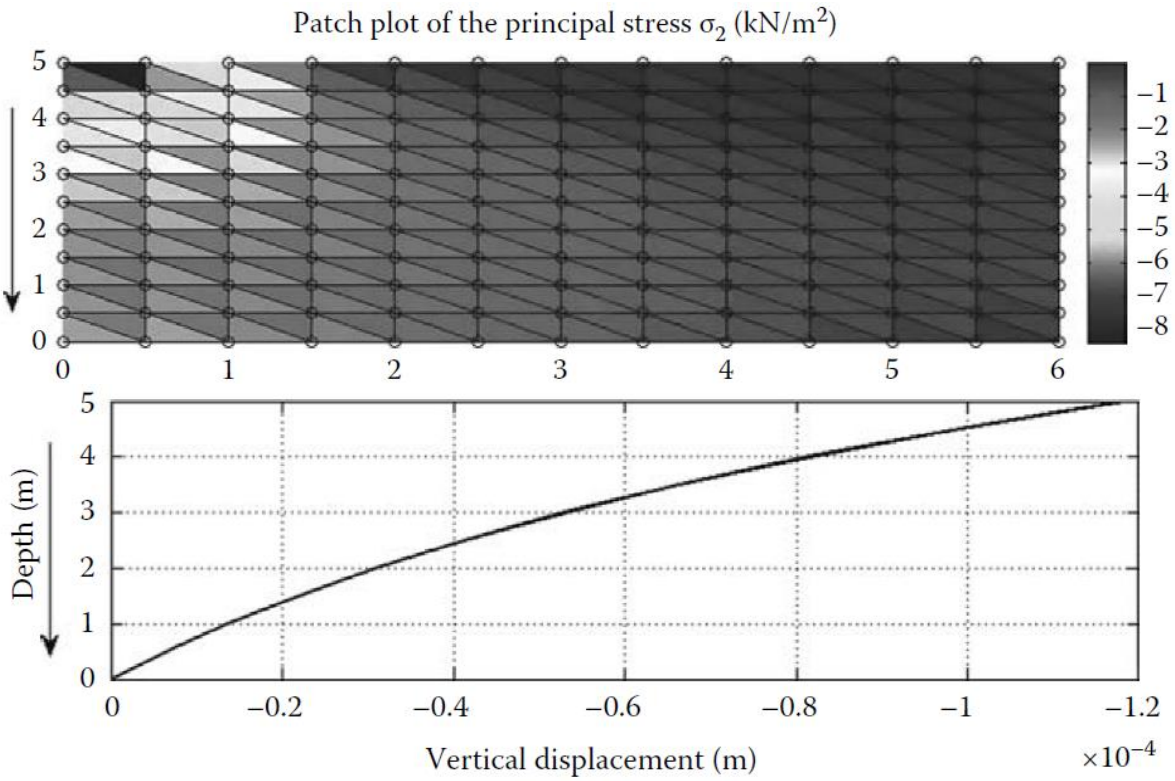
```
% supply another input file
%
Length = 6.; % Length of the model
Width =5.; % Width
NXE = 12; % Number of rows in the x direction
NYE = 10; % Number of rows in the y direction
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = 0. ; % Y origin of the global coordinate system
%
nne = 3;
nodof = 2;
eldof = nne*nodof;
%
T3_mesh ; % Generate the mesh
%
% Material
%
E = 100000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 1.; % Beam thickness in mm
%
% Form the elastic matrix for plane strain
%
dee = formdeps(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
% Restrain in the x-direction the nodes situated @
% (x = 0 or x = Length)
%
for i=1:nnd
if geom(i,1) == 0. | geom(i,1) == Length;
nf(i,:) = [0 1];
end
end
%
% Restrain in all directions the nodes situated @
% (y = 0)
%
for i=1:nnd
if geom(i,2) == 0. ;
nf(i,:) = [0 0];
```

```
end
end

%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
% Apply equivalent concentrated loads on nodes 11, 22, and 33
in the
% y-direction.
%
Nodal_loads(11,:) = [0. -1.25];
Nodal_loads(22,:) = [0. -2.50];
Nodal_loads(33,:) = [0. -1.25];
%
%%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector

%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Assembly of the global stiffness matrix
%
% initialize the global stiffness matrix to zero
```

```
%
kk = zeros(n, n);
%
for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and stering
vector
ke=thick*A*bee'*dee*bee; % Compute stiffness matrix
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2));

 %

end
node_disp(i,:) =[x_disp y_disp];
end
%
%
% Retrieve the y_disp of the nodes located on center line
beneath
% the footing
%
k = 0;
vertical_disp=zeros(1,NYE+1);
for i=1:nnd;
if geom(i,1)== 0.
k=k+1;
y_coord(k) = geom(i,2);
vertical_disp(k)=node_disp(i,2);
end
```

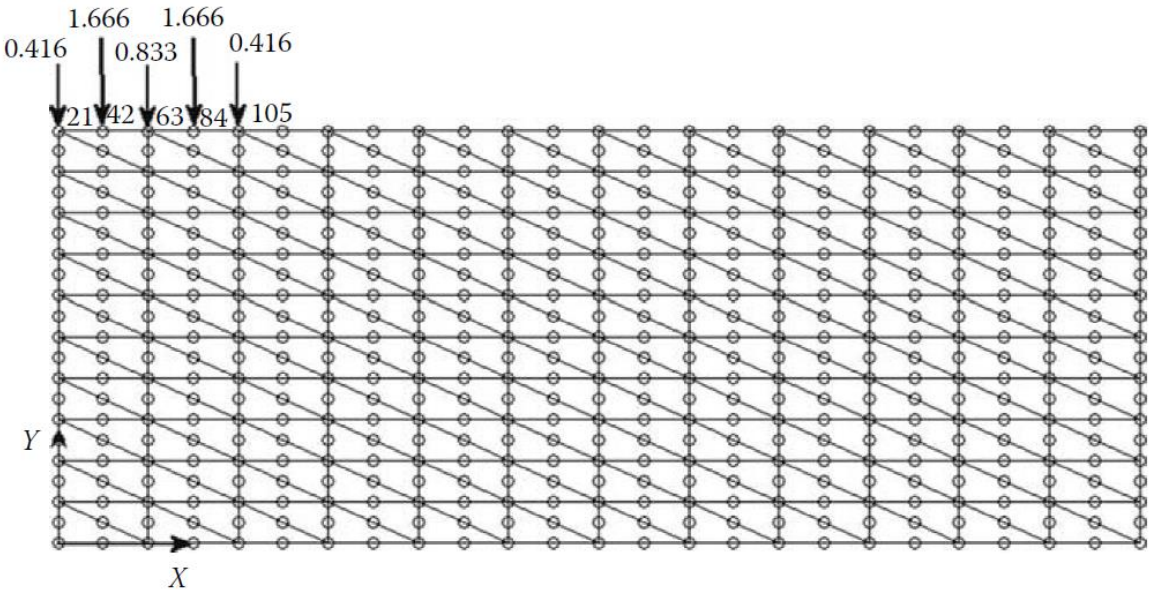
```
end
%
for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and stering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof
if g(m)==0
eld(m)=0.;
else %
eld(m)=delta(g(m)); % Retrieve element displacement
end
end
%
eps=bee*eld; % Compute strains
EPS(i,:)=eps ; % Store strains for all elements
sigma=dee*eps; % Compute stresses
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
% Calculate the principal stresses
%
SIG1=zeros(nel,1); SIG2=zeros(nel,1);
for i = 1:nel
SIG1(i)=(SIGMA(i,1)+SIGMA(i,2))/2 + ...
sqrt(((SIGMA(i,1)+SIGMA(i,2))/2)^2 +SIGMA(i,3)^2);
SIG2(i)=(SIGMA(i,1)+SIGMA(i,2))/2 - ...
sqrt(((SIGMA(i,1)+SIGMA(i,2))/2)^2 +SIGMA(i,3)^2);
end
cmin = min(SIG2);
cmax = max(SIG2);
caxis([cmin cmax]);
patch('Faces', connec, 'Vertices', geom,
'FaceVertexCData',SIG2, ...
'Facecolor','flat','Marker','o');
colorbar;
%
plottools;
```

نتایج محاسبه شده در شکل نشان داده شده است. حداکثر جابجایی و تنش به ترتیب 0.12mmو 8 کیو نیوتن بر متر مربع هستند که درست زیر پایه قرار دارد.

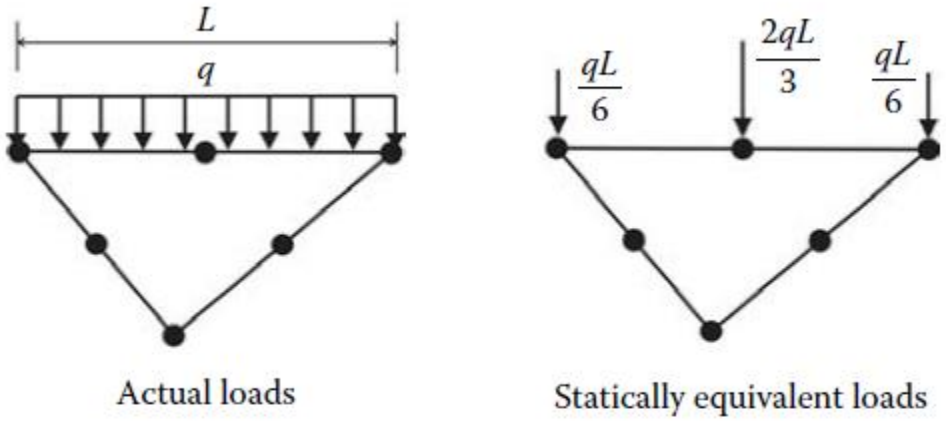


پایه نواری با LST

مجددا این بازه را در طول محور x با ۱۲ فاصله NXE=12 و در طول محور y با ۱۰ فاصله NYE=10 گسسته می سازیم.با این گسسته سازی در هر دو جهت المان هایی به اندازه ۰.۵ ایجاد می شود.



شرایط مرزی گره های محدود شده به همان روشی که قبلا با مختصات گرهی انجام شد وارد می شود.



Statically equivalent loads for the LST element.

```
Nodal_loads(21,:) = [0. -0.416];
Nodal_loads(42,:) = [0. -1.666];
Nodal_loads(63,:) = [0. -0.833];
Nodal_loads(84,:) = [0. -1.666];
Nodal_loads(105,:) = [0. -0.416];
```

LST_STRIP_FOOTING.m

```
% THIS PROGRAM USES A 6-NODE LINEAR TRIANGULAR ELEMENT FOR
THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
% IT INCLUDES AN AUTOMATIC MESH GENERATION
%
% Make these variables global so they can be shared by other
functions
%
clear all
clc
global nnd nel nne nodof eldof n
global connec geom dee nf Nodal_loads XIG YIG
global Length Width NXE NYE X_origin Y_origin
%
format long g
%
%
% To change the size of the problem or change elastic
properties
% supply another input file
```

```
%
Length = 6.; % Length of the model
Width =5.; % Width
NXE = 12; % Number of rows in the x direction
NYE = 10; % Number of rows in the y direction
XIG = zeros(2*NXE+1,1); YIG=zeros(2*NYE+1,1); % Vectors
holding grid coordinates
dhx = Length/NXE; % Element size in the x direction
dhy = Width/NYE; % Element size in the x direction
X_origin = 0. ; % X origin of the global coordinate system
Y_origin = 0. ; % Y origin of the global coordinate system
%
nne = 6;
nodof = 2;
eldof = nne*nodof;
%
T6_mesh ; % Generate the mesh
%
% Material
%
E = 100000.; % Elastic modulus in MPa
vu = 0.3; % Poisson's ratio
thick = 1.; % Beam thickness in mm
nhp = 3; % Number of sampling points
%
% Form the elastic matrix for plane stress
%
dee = formdeps(E,vu);
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1

%
% Restrain in the x-direction the nodes situated @
% (x = 0 or x = Length)
%
for i=1:nnd
if geom(i,1) == 0. | geom(i,1) == Length;
nf(i,:) = [0 1];
end
end
%
% Restrain in all directions the nodes situated @
% (y = 0)
%
```

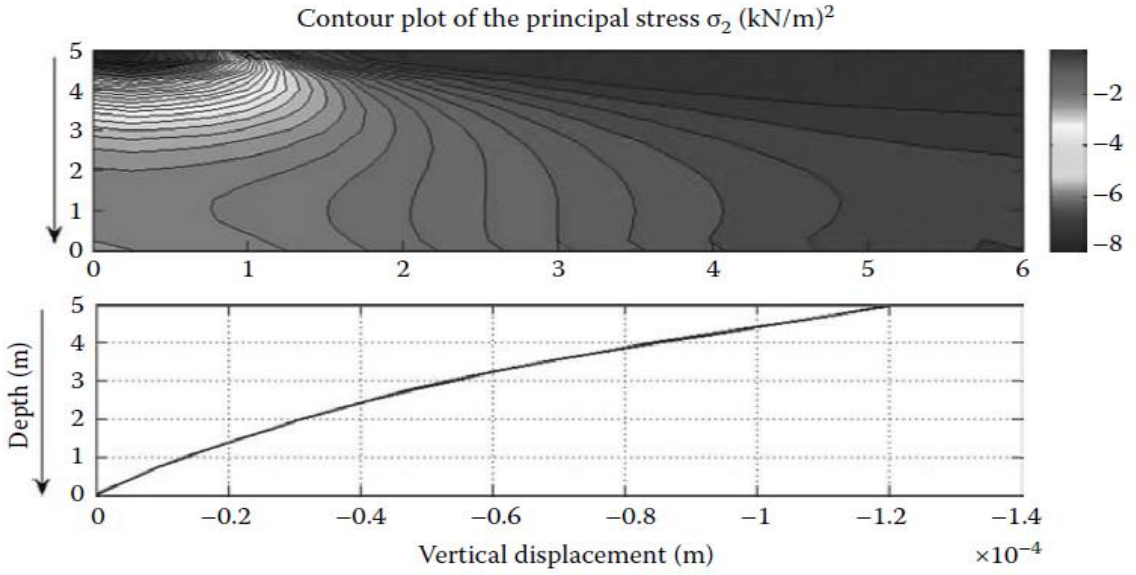
```
for i=1:nnd
if geom(i,2) == 0. ;
nf(i,:) = [0 0];
end
end
%
%
% Counting of the free degrees of freedom
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
%
% loading
%
Nodal_loads= zeros(nnd, 2); % Initialize the matrix of nodal
loads to 0
%
%
% Apply equivalent concentrated loads on nodes 21, 42, 63, 84
and 105 in the
% y-direction.
%
Nodal_loads(21,:) = [0. -0.416];
Nodal_loads(42,:) = [0. -1.666];
Nodal_loads(63,:) = [0. -0.833];
Nodal_loads(84,:) = [0. -1.666];
Nodal_loads(105,:) = [0. -0.416];
%
%
%%%%%%%%%%%%%%%%%%%%%%%% End of
input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Assemble the global force vector
%
fg=zeros(n,1);
for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
```

```
fg(nf(i,2))= Nodal_loads(i,2);
end
end
%
% Assembly of the global stiffness matrix
%
%
% Form the matrix containing the abscissas and the weights of
Hammer points
%
samp=hammer(nhp);

%
% initialize the global stiffness matrix to zero
%
kk = zeros(n, n);
%
for i=1:nel
[coord,g] = elem_T6(i); % Form strain matrix, and stering
vector
ke=zeros(eldof,eldof) ; % Initialize the element stiffness
matrix to zero
for ig = 1:nhp
wi = samp(ig,3);
[der,fun] = fmT6_quad(samp, ig);
jac = der*coord;
d = det(jac);
jac1=inv(jac); % Compute inverse of the Jacobian
deriv=jac1*der; % Derivative of shape functions in global
coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
ke=ke + d*thick*wi*bee'*dee*bee; % Integrate stiffness matrix
end
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end
%
%
%%%%%%%%%%%%%%%%%%%%%%% End of assembly
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
delta = kk\fg ; % solve for unknown displacements
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
```

```
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
node_disp(i,:) =[x_disp y_disp];
end
%
%
% Retrieve the x_coord and y_disp of the nodes located on the
neutral axis
%
k = 0;
for i=1:nnd;
if geom(i,1)== 0.
k=k+1;
y_coord(k) = geom(i,2);
vertical_disp(k)=node_disp(i,2);
end
end
%
nhp = 1; % Calculate stresses at the centroid of the element
samp=hammer(nhp);
%
for i=1:nel
[coord,g] = elem_T6(i); % Retrieve coordinates and stering
vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof %
if g(m)==0 %
eld(m)=0.; %
else %
eld(m)=delta(g(m)); % Retrieve element displacement from
% the global displacement vector

end
end
%
for ig=1: nhp
[der,fun] = fmT6_quad(samp,ig); % Derivative of shape
functions in local coordinates
jac=der*coord; % Compute Jacobian matrix
jac1=inv(jac); % Compute inverse of the Jacobian
```

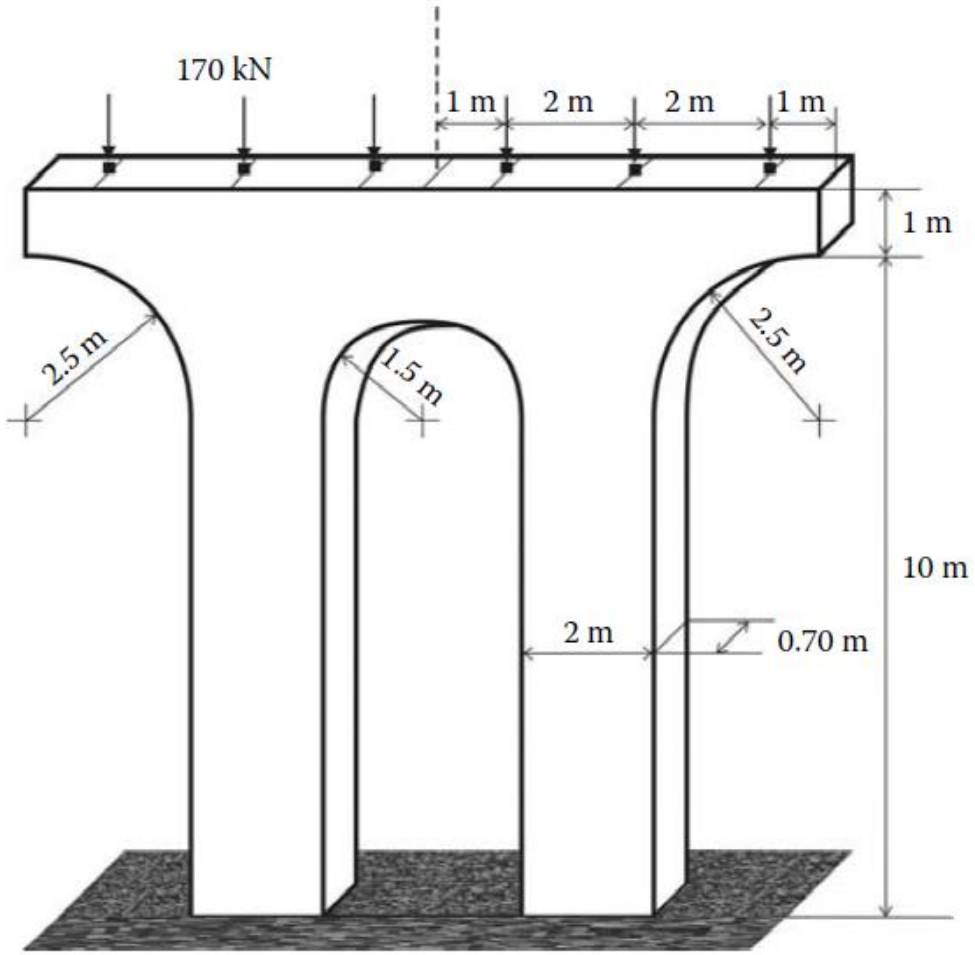
```
deriv=jac1*der; % Derivative of shape functions in global
coordinates
bee=formbee(deriv,nne,eldof); % Form matrix [B]
eps=bee*eld; % Compute strains
sigma=dee*eps ; % Compute stresses
end
SIGMA(i,:)=sigma ; % Store stresses for all elements
end
%
% Prepare stresses for plotting
%
[ZX, ZY, ZT, Z1, Z2]=prepare_contour_data(SIGMA);
%
% Plot mesh using patches
%
%
patch('Faces',connec,'Vertices',geom,'FaceVertexCData',hsv(ne
l), ...
% 'Facecolor','none','Marker','o');
%
% Plot stresses in the x_direction
%
[C,h]= contourf(XIG,YIG,Z2,40);
%clabel(C,h);
colorbar plottools;
```

نتایج محاسبه شده با LST در شکل نشان داده شده است. مانند المان CST جابجایی و تنش حداکثر درست زیر پایه و به ترتیب ۰.۱۲ میلی متر و ۸ کیلو نیوتن بر متر مربع است.



پایه پل با المان Q8

پلیه پل در معرض ۶ بار متمرکز ۱۷۰ کیلو نیوتنی قرار دارد. مصالح سازنده پل بتنی ست که مدول آن ۵۰۰۰۰ مگا پاسکال و نسبت پواسون آن ۰.۱۷ است. فرض می کنیم تکیه گاه پایه بی نهایت سخت است. اولین قدم این است که راهی برای ساده سازی پیدا کنیم. ضخامت پایه ۰.۷ متر است که در مقایسه با ابعاد افقی و عمودی نسبتا کوچک است بنابراین می توان این پل را به شکل تنش صفحه ای تحلیل کرد. بعلاوه هم هندسه هم بارگذاری متقارن هستند و می توان نصف آن راتحلیل کرد.قدم بعدی تعیین شرایط مرزی ست و قدم سوم ساختن مش بندی مناسب.بدلیل پیچیدگی هندسه نمی توان از تابع سازنده مش **Q8_mesh.m** استفاده کرد. برای مش بندی شکلی که در اینجا داریم از نسخه تعاملی آباکوس استفاده می کنیم. آباکوس فایل ورودی ایجاد می کند مختصات گرهی و اتصالات در MATLAB وارد می شوند اما این یک فرآیند مستقیم نیست. در واقع در المان چهارضلعی ۸ گرهی آباکوس گره هارا بطور متفاوتی شماره گذاری می کند.



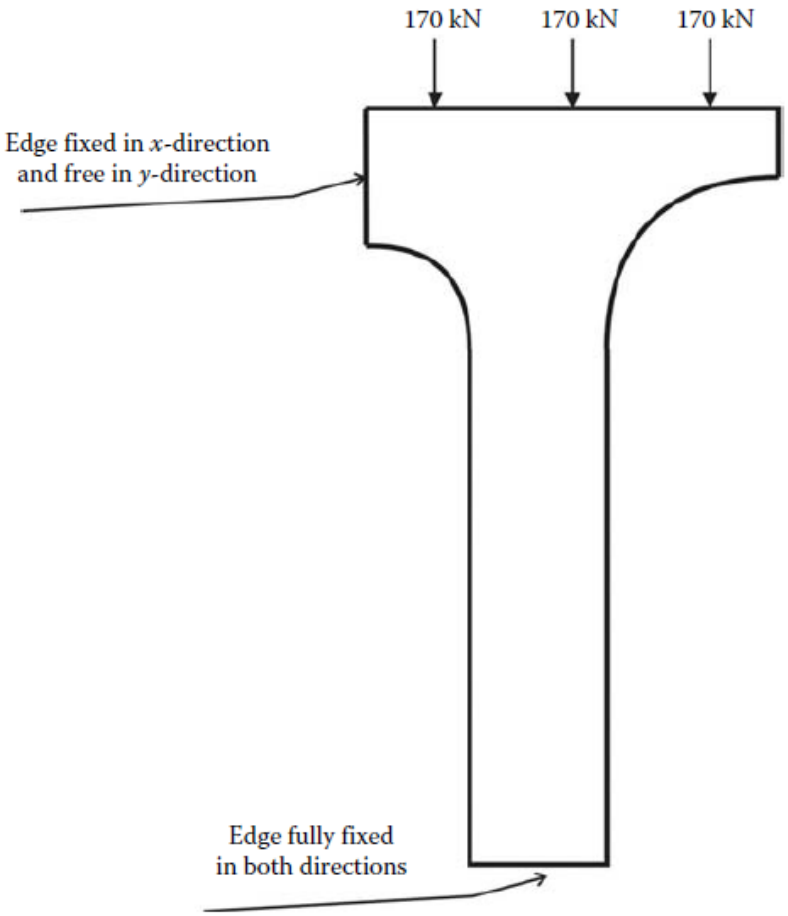
Bridge pier.

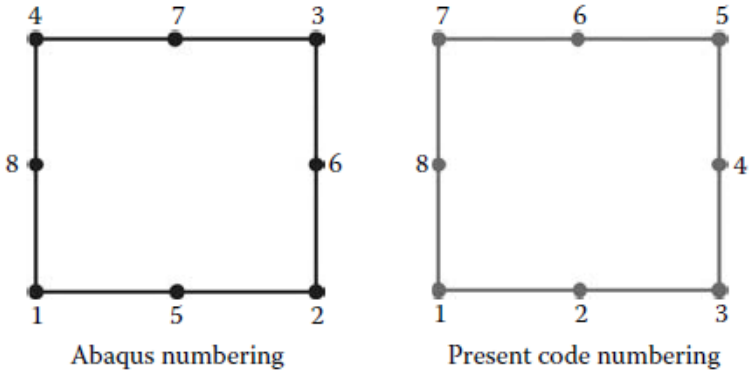
در ادامه قسمتی از فایل ورودی آباکوس **pier.inp** را میبینیم که اتصال های ۱ تا ۱۰ را فهرست کرده:

```
*Element, type=CPS8R
1, 65, 67, 117, 64, 173, 174, 175, 176
2, 67, 65, 66, 116, 173, 177, 178, 179
3, 62, 20, 114, 68, 180, 181, 182, 183
4, 118, 113, 69, 120, 184, 185, 186, 187
5, 145, 158, 161, 156, 188, 189, 190, 191
6, 70, 69, 2, 1, 192, 193, 194, 195
7, 141, 78, 79, 152, 196, 197, 198, 199
8, 120, 69, 70, 119, 186, 192, 200, 201
9, 112, 73, 60, 61, 202, 203, 204, 205
10, 81, 137, 121, 167, 206, 207, 208, 209
.....
```

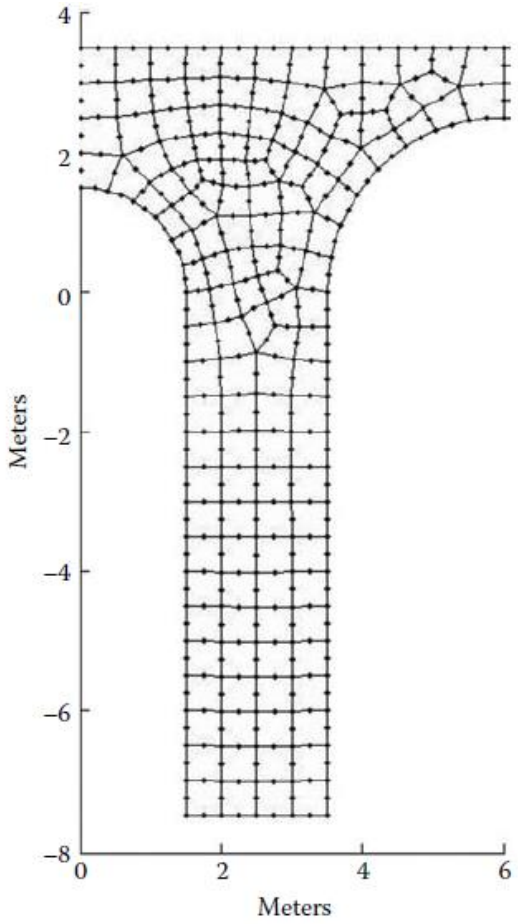
برای آنکه این داده ها در برنامه MATLAB **Q8_PLANE_STRESS.m** مورد استفاده قرار بگیرد به صورت زیر باز آرایی می شوند:

```
connec = [65 173 67 174 117 175 64 176 ;... % Element 1
67 173 65 177 66 178 116 179 ;...
62 180 20 181 114 182 68 183 ;...
118 184 113 185 69 186 120 187 ;...
145 188 158 189 161 190 156 191 ;...
70 192 69 193 2 194 1 195 ;...
141 196 78 197 79 198 152 199 ;...
120 186 69 192 70 200 119 201 ;...
112 202 73 203 60 204 61 205 ;...
81 206 137 207 121 208 167 209 ;...
```





این کار با وارد کردن ستونها به برنامه اکسل و باز آرایی ستونها بصورت دستی انجام شده است. این داده از ۱۳۸ المان و ۴۸۱ گره تشکیل شده است.جزئیات ان در برنامه **PIER_Q8_data.m** آمده و شبکه واقعی را در شکل ملاحظه می فرمایید. همه گره هایی که در x=0 قرار دارند در جهت x ثابت هستند و همه گره هایی که در y=-7.5 قرار دارند در هر دو جهت ثابت هستند. گره های ۱۸ ، ۱۹ و ۲۰ هریک در معرض یک نیروی ۱۷۰- قرار دارند.



PIER_Q8_data.m

```
%%%%%%%%%%%%%%%%%%%%%%%% Beginning of data input
%%%%%%%%%%%%%%%%%%%%%%%%
%
% Data file for the bridge pier analysis using 8-nodded
quadrilaterals
%
global nnd nel nne nodof eldof n ngp
global geom connec dee nf Nodal_loads
%
nnd = 481 ; % Number of nodes:
nel = 138; % Number of elements:
nne = 8 ; % Number of nodes per element:
nodof =2; % Number of degrees of freedom per node
ngp = 2 % number of Gauss points
eldof = nne*nodof; % Number of degrees of freedom per element
%
% Thickness of the domain

thick = 0.7 ; % Thickness in meters
% Material properties
```

```
%
E=50.e6; vu=0.17; % Young's modulus (kN/m^2)and Poisson's
ratio
%
% Form the matrix of elastic properties
%
dee=formdsig(E,vu); % Matrix of elastic properties plane
strain
%
% Nodes coordinates x and y
%
geom = [1.4489 0.3882 ;... x and y coordinates of node
481
1.2990 0.7500 ;...
1.0607 1.0607 ;...
0.7500 1.2990 ;...
0.3882 1.4489 ;...
1.5000 0.0000 ;...
0.0000 1.5000 ;...
1.5000 -7.5000 ;...
3.5000 -7.5000 ;...
3.5000 0.0000 ;...
3.5480 0.4877 ;...
3.9213 1.3889 ;...
4.2322 1.7678 ;...
4.6111 2.0787 ;...
5.5123 2.4520 ;...
6.0000 2.5000 ;...
6.0000 3.5000 ;...
5.0000 3.5000 ;...
3.0000 3.5000 ;...
1.0000 3.5000 ;...
0.0000 3.5000 ;...
1.5000 -0.5000 ;...
1.5000 -1.0000 ;...
1.5000 -1.5000 ;...
1.5000 -2.0000 ;...
1.5000 -2.5000 ;...
1.5000 -3.0000 ;...
1.5000 -3.5000 ;...
1.5000 -4.0000 ;...
1.5000 -4.5000 ;...
1.5000 -5.0000 ;...
1.5000 -5.5000 ;...
1.5000 -6.0000 ;...
1.5000 -6.5000 ;...
1.5000 -7.0000 ;...
```

```
2.0000 -7.5000 ;...
2.5000 -7.5000 ;...
3.0000 -7.5000 ;...
3.5000 -7.0000 ;...
3.5000 -6.5000 ;...
3.5000 -6.0000 ;...
3.5000 -5.5000 ;...
3.5000 -5.0000 ;...
3.5000 -4.5000 ;...
3.5000 -4.0000 ;...
3.5000 -3.5000 ;...
3.5000 -3.0000 ;...
3.5000 -2.5000 ;...
3.5000 -2.0000 ;...
3.5000 -1.5000 ;...
3.5000 -1.0000 ;...
3.5000 -0.5000 ;...
3.6903 0.9567 ;...
5.0433 2.3097 ;...
6.0000 3.0000 ;...

5.5000 3.5000 ;...
4.5000 3.5000 ;...
4.0000 3.5000 ;...
3.5000 3.5000 ;...
2.5000 3.5000 ;...
2.0000 3.5000 ;...
1.5000 3.5000 ;...
0.5000 3.5000 ;...
0.0000 3.0000 ;...
0.0000 2.5000 ;...
0.0000 2.0000 ;...
0.5279 2.5104 ;...
1.4959 3.0626 ;...
1.5745 0.8916 ;...
1.7401 0.4784 ;...
3.0678 0.0527 ;...
3.2060 0.5869 ;...
2.4709 3.0574 ;...
3.6579 1.6005 ;...
4.4579 2.3344 ;...
4.5316 2.9797 ;...
4.9797 3.1580 ;...
1.8298 0.0390 ;...
1.9024 -0.4486 ;...
1.9821 -0.9532 ;...
3.9824 2.0741 ;...
```

```
2.9942 -1.9923 ;...
2.0045 -2.5012 ;...
3.0014 -3.5050 ;...
2.0045 -4.0063 ;...
3.0035 -5.0060 ;...
2.0042 -5.5061 ;...
3.0033 -6.0056 ;...
2.0026 -6.0047 ;...
3.0027 -6.5043 ;...
2.0013 -6.5036 ;...
3.0016 -7.0020 ;...
2.0002 -7.0015 ;...
3.0035 -5.5056 ;...
2.0042 -5.0056 ;...
2.0038 -4.5049 ;...
3.0033 -4.5053 ;...
3.0027 -4.0051 ;...
2.0039 -3.5054 ;...
2.0034 -3.0033 ;...
2.9993 -3.0030 ;...
2.9970 -2.4998 ;...
2.0057 -1.9916 ;...
2.0044 -1.4742 ;...
2.9993 -1.4785 ;...
3.0327 -0.9731 ;...
3.0998 -0.4957 ;...
3.3175 1.1463 ;...
3.9957 3.0076 ;...
3.4702 2.9078 ;...
2.9566 3.0118 ;...
1.9863 3.0733 ;...
1.3008 1.2396 ;...
1.0014 3.0384 ;...
0.9828 1.5734 ;...
0.5929 1.9562 ;...
0.5039 3.0118 ;...
1.5590 1.4193 ;...
2.0705 0.5842 ;...
1.9288 1.0906 ;...
3.6925 2.5511 ;...
4.8666 2.6297 ;...

2.5004 -2.4996 ;...
2.5046 -4.0070 ;...
2.5049 -5.5071 ;...
2.5027 -6.5046 ;...
2.5035 -6.0055 ;...
```

```
2.5047 -5.0066 ;...
2.5043 -4.5058 ;...
2.5031 -3.5058 ;...
2.5017 -3.0033 ;...
2.4994 -1.9860 ;...
2.5006 -1.4524 ;...
2.5053 -0.8546 ;...
2.7641 -0.4931 ;...
2.8417 1.1161 ;...
3.9934 2.5661 ;...
1.3049 1.7687 ;...
2.8367 0.6671 ;...
3.1567 1.5168 ;...
2.1675 0.1239 ;...
2.5011 -7.0019 ;...
2.6413 -0.1504 ;...
2.4210 2.6375 ;...
2.8474 2.5634 ;...
1.9773 2.6633 ;...
1.5193 2.6440 ;...
1.0320 2.5819 ;...
2.0271 1.5224 ;...
2.4001 1.0934 ;...
2.4406 0.6514 ;...
2.2846 -0.3356 ;...
2.8123 1.5885 ;...
1.9850 2.2736 ;...
1.1200 2.1537 ;...
3.2207 2.4235 ;...
1.5744 2.2513 ;...
2.7330 2.1697 ;...
2.3705 2.2499 ;...
1.6578 1.8873 ;...
3.0443 2.0121 ;...
2.3874 1.5056 ;...
2.0083 1.8949 ;...
2.3454 1.8741 ;...
5.3776 2.9409 ;...
4.3237 2.6067 ;...
3.5630 2.2212 ;...
2.5059 0.2226 ;...
3.3638 1.8329 ;...
1.7419 1.5990 ;...
2.7962 0.3080 ;...
2.6233 1.8785 ;...
0.2640 2.5052 ;...
0.5159 2.7611 ;...
```

```
0.2520 3.0059 ;...
0.0000 2.7500 ;...
0.0000 2.2500 ;...
0.2964 1.9781 ;...
0.5604 2.2333 ;...
1.2500 3.5000 ;...
1.0007 3.2692 ;...
1.2487 3.0505 ;...
1.4980 3.2813 ;...
1.4299 1.3294 ;...
1.4377 1.0656 ;...
1.7516 0.9911 ;...
1.7439 1.2549 ;...
2.7902 2.3665 ;...
2.8887 2.0909 ;...

3.1325 2.2178 ;...
3.0341 2.4934 ;...
1.6573 0.6850 ;...
1.4368 0.8208 ;...
1.3858 0.5740 ;...
1.5945 0.4333 ;...
1.9987 0.0815 ;...
1.8661 -0.2048 ;...
2.0935 -0.3921 ;...
2.2260 -0.1058 ;...
1.9053 0.5313 ;...
1.9996 0.8374 ;...
2.2286 3.0653 ;...
2.4854 3.2787 ;...
2.2500 3.5000 ;...
1.9932 3.2866 ;...
3.9879 2.3201 ;...
3.8430 2.5586 ;...
3.6277 2.3861 ;...
3.7727 2.1477 ;...
5.1221 2.7853 ;...
4.9550 2.4697 ;...
5.2743 2.3924 ;...
5.4449 2.6964 ;...
4.6623 2.4821 ;...
4.5345 2.2065 ;...
4.8215 2.2048 ;...
5.4388 3.2205 ;...
5.2500 3.5000 ;...
4.9898 3.3290 ;...
5.1786 3.0494 ;...
```

```
6.0000 2.7500 ;...
5.6888 2.9705 ;...
5.7550 2.4880 ;...
1.6649 0.0195 ;...
1.5000 -0.2500 ;...
1.7012 -0.4743 ;...
3.2999 -0.4979 ;...
3.0663 -0.7344 ;...
3.2664 -0.9866 ;...
3.5000 -0.7500 ;...
1.9423 -0.7009 ;...
1.5000 -0.7500 ;...
1.7411 -0.9766 ;...
4.2202 2.2043 ;...
4.1073 1.9210 ;...
4.4140 1.9325 ;...
3.2496 -1.4892 ;...
3.5000 -1.2500 ;...
3.0160 -1.2258 ;...
2.9956 -2.2460 ;...
2.7468 -1.9891 ;...
2.4999 -2.2428 ;...
2.7487 -2.4997 ;...
3.2497 -3.0015 ;...
3.5000 -2.7500 ;...
3.2485 -2.4999 ;...
2.9981 -2.7514 ;...
3.0020 -3.7550 ;...
2.7522 -3.5054 ;...
2.5038 -3.7564 ;...
2.7536 -4.0061 ;...
3.2517 -4.5026 ;...
3.5000 -4.2500 ;...
3.2513 -4.0026 ;...
3.0030 -4.2552 ;...
3.0035 -5.2558 ;...

2.7541 -5.0063 ;...
2.5048 -5.2569 ;...
2.7542 -5.5063 ;...
2.2531 -6.0051 ;...
2.0019 -6.2541 ;...
2.2520 -6.5041 ;...
2.5031 -6.2551 ;...
2.2507 -7.0017 ;...
2.5019 -6.7533 ;...
2.0007 -6.7525 ;...
```

```
3.2517 -6.0028 ;...
3.0030 -6.2549 ;...
3.2514 -6.5021 ;...
3.5000 -6.2500 ;...
2.7527 -6.5045 ;...
2.7534 -6.0056 ;...
2.5042 -5.7563 ;...
3.0034 -5.7556 ;...
3.0022 -6.7531 ;...
3.2508 -7.0010 ;...
3.5000 -6.7500 ;...
2.0001 -7.2507 ;...
1.7501 -7.0007 ;...
1.5000 -7.2500 ;...
1.7500 -7.5000 ;...
1.7506 -6.5018 ;...
1.5000 -6.7500 ;...
3.2500 -7.5000 ;...
3.5000 -7.2500 ;...
3.0008 -7.2510 ;...
3.2517 -5.0030 ;...
3.2517 -5.5028 ;...
3.5000 -5.2500 ;...
2.2545 -5.5066 ;...
2.2545 -5.0061 ;...
2.0042 -5.2559 ;...
2.5045 -4.7562 ;...
3.0034 -4.7556 ;...
2.7538 -4.5055 ;...
2.2541 -4.5054 ;...
2.0040 -4.7553 ;...
3.2507 -3.5025 ;...
3.5000 -3.7500 ;...
2.2546 -4.0067 ;...
2.2535 -3.5056 ;...
2.0042 -3.7558 ;...
2.5024 -3.2546 ;...
3.0004 -3.2540 ;...
2.7505 -3.0031 ;...
2.2525 -3.0033 ;...
2.0036 -3.2544 ;...
3.2471 -1.9961 ;...
3.5000 -2.2500 ;...
2.2524 -2.5004 ;...
2.2525 -1.9888 ;...
2.0051 -2.2464 ;...
2.5000 -1.7192 ;...
```

```
2.2525 -1.4633 ;...
2.0051 -1.7329 ;...
2.5030 -1.1535 ;...
2.2437 -0.9039 ;...
1.9933 -1.2137 ;...
2.6347 -0.6739 ;...
2.7690 -0.9138 ;...
2.9320 -0.4944 ;...
3.2839 0.0264 ;...
3.0838 -0.2215 ;...

3.5000 -0.2500 ;...
1.9780 1.3065 ;...
2.1645 1.0920 ;...
2.3938 1.2995 ;...
2.2072 1.5140 ;...
4.7557 3.0688 ;...
4.7500 3.5000 ;...
4.5158 3.2399 ;...
6.0000 3.2500 ;...
5.7500 3.5000 ;...
3.2134 2.9598 ;...
3.4851 3.2039 ;...
3.2500 3.5000 ;...
2.9783 3.2559 ;...
2.7138 3.0346 ;...
2.7500 3.5000 ;...
1.7411 3.0679 ;...
1.7500 3.5000 ;...
1.1807 1.1501 ;...
1.1900 0.9131 ;...
0.5020 3.2559 ;...
0.2500 3.5000 ;...
0.0000 3.2500 ;...
1.1418 1.4065 ;...
0.8664 1.4362 ;...
0.9131 1.1900 ;...
0.8565 2.0550 ;...
0.7878 1.7648 ;...
1.1438 1.6711 ;...
1.2125 1.9612 ;...
0.1958 1.4872 ;...
0.4905 1.7026 ;...
0.0000 1.7500 ;...
1.4319 1.5940 ;...
0.7800 2.5461 ;...
1.0167 2.8101 ;...
```

```
0.7527 3.0251 ;...
3.7896 1.4947 ;...
3.4877 1.3734 ;...
3.5039 1.0515 ;...
3.7952 1.1785 ;...
1.4872 0.1958 ;...
1.7850 0.2587 ;...
3.3770 0.5373 ;...
3.1369 0.3198 ;...
3.5120 0.2450 ;...
3.6076 0.7257 ;...
3.2618 0.8666 ;...
2.8392 0.8916 ;...
2.6209 1.1048 ;...
2.4204 0.8724 ;...
2.6386 0.6593 ;...
3.8201 1.8373 ;...
3.4634 2.0270 ;...
3.5109 1.7167 ;...
4.0675 1.5860 ;...
4.2636 2.9936 ;...
4.2500 3.5000 ;...
3.9978 3.2538 ;...
4.6991 2.8047 ;...
2.7027 -0.3218 ;...
2.8545 -0.0488 ;...
1.5000 -1.2500 ;...
1.7522 -1.4871 ;...
3.7330 2.9577 ;...
3.7500 3.5000 ;...
1.7528 -1.9958 ;...

1.5000 -2.2500 ;...
1.7522 -2.5006 ;...
3.5000 -1.7500 ;...
2.9968 -1.7354 ;...
1.5000 -2.7500 ;...
1.7517 -3.0017 ;...
2.0039 -2.7523 ;...
1.7519 -3.5027 ;...
1.5000 -3.7500 ;...
1.7523 -4.0032 ;...
3.5000 -3.2500 ;...
1.5000 -4.2500 ;...
1.7519 -4.5025 ;...
2.0041 -4.2556 ;...
1.7521 -5.0028 ;...
```

```
1.5000 -5.2500 ;...
1.7521 -5.5031 ;...
3.5000 -4.7500 ;...
1.5000 -5.7500 ;...
1.7513 -6.0023 ;...
2.0034 -5.7554 ;...
3.5000 -5.7500 ;...
1.5000 -6.2500 ;...
2.7514 -7.0020 ;...
2.5006 -7.2509 ;...
2.7500 -7.5000 ;...
2.2500 -7.5000 ;...
2.5045 -4.2564 ;...
1.5000 -4.7500 ;...
2.5010 -2.7515 ;...
1.5000 -3.2500 ;...
2.7500 -1.4655 ;...
1.5000 -1.7500 ;...
2.4732 0.4370 ;...
2.6511 0.2653 ;...
2.8164 0.4876 ;...
2.3950 -0.5951 ;...
2.4629 -0.2430 ;...
2.1768 1.8845 ;...
2.0177 1.7087 ;...
2.3664 1.6899 ;...
3.5814 2.7294 ;...
3.9945 2.7868 ;...
3.0796 1.1312 ;...
3.2371 1.3315 ;...
2.9845 1.5527 ;...
2.8270 1.3523 ;...
2.4459 2.8474 ;...
2.6342 2.6004 ;...
2.9020 2.7876 ;...
1.5076 2.8533 ;...
1.7483 2.6537 ;...
1.9818 2.8683 ;...
1.5468 2.4477 ;...
1.7797 2.2624 ;...
1.9812 2.4685 ;...
0.7500 3.5000 ;...
1.2757 2.6130 ;...
0.5740 1.3858 ;...
2.1992 2.6504 ;...
2.1190 0.3541 ;...
2.3367 0.1733 ;...
```

```
2.2555 0.6178 ;...
3.3455 2.6656 ;...
4.1586 2.5864 ;...
4.4277 2.7932 ;...
2.5736 0.0361 ;...

3.0213 0.6270 ;...
3.2603 1.6748 ;...
4.3908 2.4706 ;...
1.0760 2.3678 ;...
1.3472 2.2025 ;...
2.9320 0.1804 ;...
2.9283 1.8003 ;...
3.2041 1.9225 ;...
2.3579 2.0620 ;...
2.1778 2.2618 ;...
1.9967 2.0842 ;...
2.5517 2.2098 ;...
2.4844 1.8763 ;...
2.6782 2.0241 ;...
1.4813 1.8280 ;...
1.6161 2.0693 ;...
1.8845 1.5607 ;...
1.6504 1.5091 ;...
2.5999 1.5471 ;...
2.7178 1.7335 ;...
2.3957 2.4437 ;...
1.6998 1.7431 ;...
3.3919 2.3224 ;...
1.8330 1.8911 ]; % x and y coordinates of node 481
%
% Element connectivity
%
connec = [65 173 67 174 117 175 64 176 ;... % Element 1
67 173 65 177 66 178 116 179 ;...
62 180 20 181 114 182 68 183 ;...
118 184 113 185 69 186 120 187 ;...
145 188 158 189 161 190 156 191 ;...
70 192 69 193 2 194 1 195 ;...
141 196 78 197 79 198 152 199 ;...
120 186 69 192 70 200 119 201 ;...
112 202 73 203 60 204 61 205 ;...
81 206 137 207 121 208 167 209 ;...
165 210 122 211 54 212 15 213 ;...
54 211 122 214 75 215 14 216 ;...
165 217 56 218 18 219 77 220 ;...
16 221 55 222 165 213 15 223 ;...
```

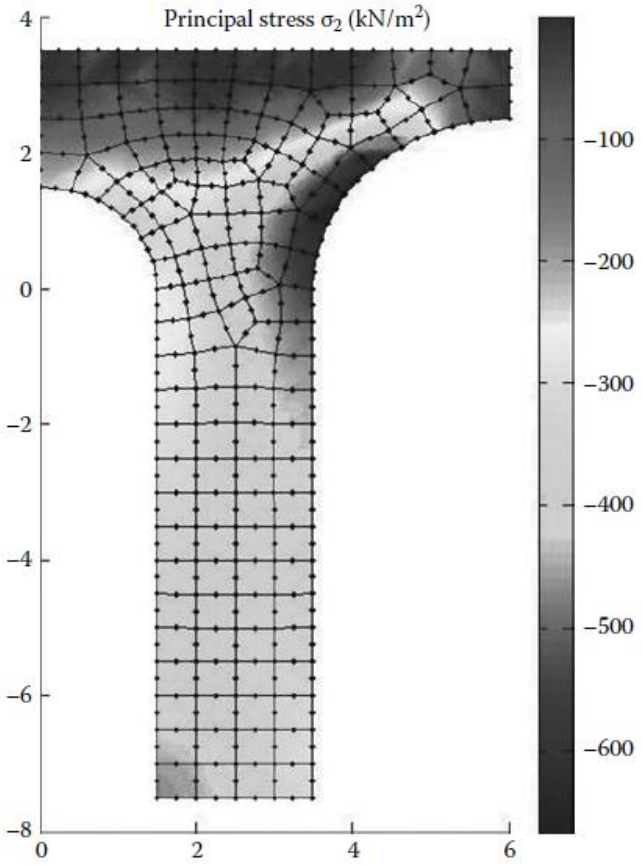
```
79 197 78 224 6 225 22 226 ;...
52 227 107 228 106 229 51 230 ;...
80 231 79 226 22 232 23 233 ;...
14 215 75 234 81 235 13 236 ;...
105 237 50 238 51 229 106 239 ;...
102 240 82 241 132 242 123 243 ;...
101 244 47 245 48 246 102 247 ;...
98 248 84 249 130 250 124 251 ;...
97 252 44 253 45 254 98 255 ;...
94 256 86 257 128 258 125 259 ;...
127 260 89 261 91 262 126 263 ;...
93 264 142 265 126 262 91 266 ;...
41 267 88 268 90 269 40 270 ;...
127 263 126 271 90 268 88 272 ;...
125 273 127 272 88 274 94 259 ;...
40 269 90 275 92 276 39 277 ;...
36 278 93 279 35 280 8 281 ;...
93 266 91 282 34 283 35 279 ;...
38 284 9 285 39 276 92 286 ;...
43 287 86 256 94 288 42 289 ;...
87 290 125 258 128 291 95 292 ;...
129 293 128 257 86 294 97 295 ;...
95 291 128 293 129 296 96 297 ;...
46 298 84 248 98 254 45 299 ;...
85 300 124 250 130 301 99 302 ;...
131 303 130 249 84 304 101 305 ;...

99 301 130 303 131 306 100 307 ;...
49 308 82 240 102 246 48 309 ;...
83 310 123 242 132 311 103 312 ;...
103 311 132 313 133 314 104 315 ;...
133 316 134 317 80 318 104 314 ;...
135 319 134 320 106 228 107 321 ;...
10 322 71 323 107 227 52 324 ;...
149 325 120 326 150 327 162 328 ;...
76 329 77 219 18 330 57 331 ;...
55 332 17 333 56 217 165 222 ;...
111 334 110 335 59 336 19 337 ;...
73 338 111 337 19 339 60 203 ;...
68 340 112 205 61 341 62 183 ;...
69 185 113 342 3 343 2 193 ;...
64 175 117 344 63 345 21 346 ;...
113 347 115 348 4 349 3 342 ;...
155 350 116 351 115 352 138 353 ;...
7 354 5 355 116 178 66 356 ;...
113 184 118 357 138 352 115 347 ;...
117 174 67 358 148 359 114 360 ;...
```

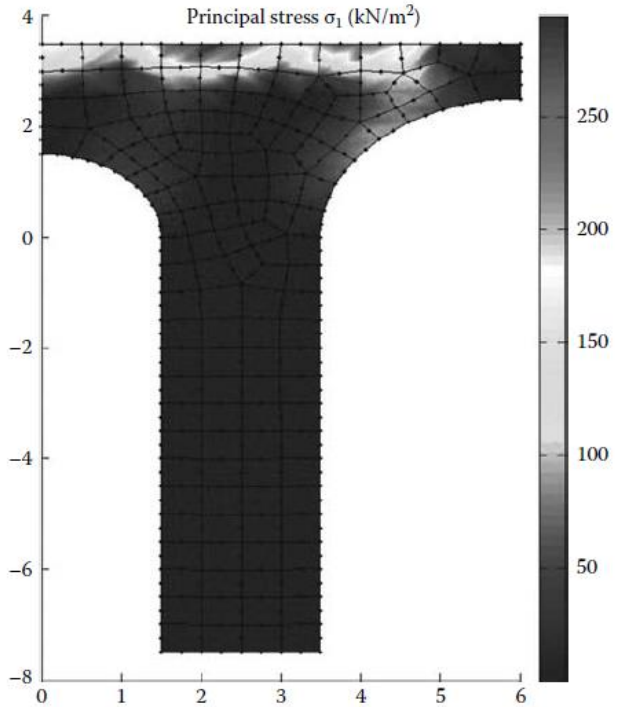
```
12 361 74 362 108 363 53 364 ;...
1 365 6 224 78 366 70 195 ;...
11 367 72 368 71 322 10 369 ;...
72 367 11 370 53 363 108 371 ;...
139 372 136 373 150 374 151 375 ;...
74 376 81 209 167 377 169 378 ;...
74 361 12 379 13 235 81 376 ;...
109 380 76 331 57 381 58 382 ;...
77 329 76 383 122 210 165 220 ;...
143 384 135 321 107 323 71 385 ;...
23 386 24 387 104 318 80 233 ;...
110 388 109 382 58 389 59 335 ;...
83 312 103 390 25 391 26 392 ;...
82 308 49 393 50 237 105 394 ;...
26 395 27 396 100 397 83 392 ;...
85 302 99 398 28 399 29 400 ;...
84 298 46 401 47 244 101 304 ;...
29 402 30 403 96 404 85 400 ;...
87 292 95 405 31 406 32 407 ;...
86 287 43 408 44 252 97 294 ;...
32 409 33 410 89 411 87 407 ;...
88 267 41 412 42 288 94 274 ;...
33 413 34 282 91 261 89 410 ;...
38 286 92 414 142 415 37 416 ;...
93 278 36 417 37 415 142 264 ;...
124 418 129 295 97 255 98 251 ;...
30 419 31 405 95 297 96 403 ;...
123 420 131 305 101 247 102 243 ;...
27 421 28 398 99 307 100 396 ;...
133 313 132 241 82 394 105 422 ;...
24 423 25 390 103 315 104 387 ;...
133 422 105 239 106 320 134 316 ;...
151 424 168 425 171 426 139 375 ;...
152 427 134 319 135 384 143 428 ;...
164 429 163 430 149 328 162 431 ;...
110 432 121 207 137 433 109 388 ;...
136 434 108 435 140 436 153 437 ;...
73 438 144 439 145 440 111 338 ;...
68 441 147 442 146 443 112 340 ;...
147 444 157 445 154 446 146 442 ;...
117 360 114 181 20 447 63 344 ;...
114 359 148 448 147 441 68 182 ;...
5 449 4 348 115 351 116 355 ;...
112 443 146 450 144 438 73 202 ;...
70 366 78 196 141 451 119 200 ;...
141 452 168 424 151 453 119 451 ;...
119 453 151 374 150 326 120 201 ;...
```

```
156 454 110 334 111 440 145 191 ;...
137 455 166 456 76 380 109 433 ;...
123 310 83 397 100 306 131 420 ;...
124 300 85 404 96 296 129 418 ;...
125 290 87 411 89 260 127 273 ;...
92 275 90 271 126 265 142 414 ;...
141 199 152 428 143 457 168 452 ;...
108 434 136 372 139 458 72 371 ;...
140 435 108 362 74 378 169 459 ;...
166 460 75 214 122 383 76 456 ;...
148 461 155 462 157 444 147 448 ;...
72 458 139 426 171 463 71 368 ;...
161 464 153 436 140 459 169 465 ;...
163 429 164 466 159 467 154 468 ;...
158 469 159 466 164 470 172 471 ;...
138 472 160 473 157 462 155 353 ;...
155 461 148 358 67 179 116 350 ;...
120 325 149 474 170 475 118 187 ;...
136 437 153 476 162 327 150 373 ;...
152 198 79 231 80 317 134 427 ;...
161 189 158 471 172 477 153 464 ;...
146 446 154 467 159 478 144 450 ;...
118 475 170 479 160 472 138 357 ;...
121 432 110 454 156 480 167 208 ;...
158 188 145 439 144 478 159 469 ;...
154 445 157 473 160 481 163 468 ;...
161 465 169 377 167 480 156 190 ;...
162 476 153 477 172 470 164 431 ;...
75 460 166 455 137 206 81 234 ;...
171 425 168 457 143 385 71 463 ;...
170 474 149 430 163 481 160 479 ]; % Element 138
%
%
% Boundary conditions
%
nf = ones(nnd, nodof); % Initialize the matrix nf to 1
%
for i=1:nnd
if geom(i,1) == 0.;
nf(i,:) = [0 1];
end
if geom(i,2) == -7.5 ;
nf(i,:) = [0 0];
end
end
%
% Counting of the free degrees of freedom
```

```
%
n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end
disp ('Nodal freedom')
nf
disp ('Total number of active degrees of freedom')
n
%
% loading
%
Nodal_loads = zeros(nnd, 2);
Nodal_loads(18,2)=-170.; % Vertical load on node 18
Nodal_loads(19,2)=-170.; % Vertical load on node 19

Nodal_loads(20,2)=-170.; % Vertical load on node 20

%

End input%
```

برای انجام این مثال در برنامه **Q8_PLANE_STRESS.m** به جای **Q8_coarse_mesh_data** ، **PIER_Q8_data.m** را قرار می دهیم. نتایج بدست امده نشان داده شده اند. کانتورهای مربوط به تنش های اصلی شاید دقیق نباشند چون در مرکز المان ها محاسبه شده اند و در گره ها میانگین گیری می شوند. واضح است با استفاده از مش ریزتر میتوان جزئیات بیشتری بدست آورد.


Principal stress σ_2 (kN/m^2)
4
2
0
−2
−4
−6
−8
0
2
4
6
−100
−200
−300
−400
−500
−600


Principal stress σ_1 (kN/m^2)
4
2
0
−2
−4
−6
−8
0
1
2
3
4
5
6
250
200
150
100
50