# ELEMENTO TRIANGULAR LINEAL - TENSION PLANA

## 1 - DEFINICION MATRIZ RIGIDEZ

```
StiffnessOf3NodePlaneStressTriangle[{{x1_, y1_}, {x2_, y2_}, {x3_, y3_}}, Emat_, {h_}] :=
  Module[{x21, x13, x32, y12, y31, y23, A, Be, Ke},
   A = Simplify[((x2 * y3 - x3 * y2) + (x3 * y1 - x1 * y3) + (x1 * y2 - x2 * y1)) / 2];
   {x21, x13, x32, y12, y31, y23} = {x2 - x1, x1 - x3, x3 - x2, y1 - y2, y3 - y1, y2 - y3};
   Be = {{y23, 0, y31, 0, y12, 0}, {0, x32, 0, x13, 0, x21}, {x32, y23, x13, y31, x21, y12}} / (2 * A);
   Ke = (A * h) * Transpose[Be].Emat.Be;
   Return[Ke];];
```

### DETALLES

```
A = Simplify[((x2 * y3 - x3 * y2) + (x3 * y1 - x1 * y3) + (x1 * y2 - x2 * y1)) / 2]
```

$$\frac{1}{2}\left(x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)\right)$$

```
{x21, x13, x32, y12, y31, y23} = {x2 - x1, x1 - x3, x3 - x2, y1 - y2, y3 - y1, y2 - y3}
```

$$\{-x1+x2,\ x1-x3,\ -x2+x3,\ y1-y2,\ -y1+y3,\ y2-y3\}$$

```
x21
```

$$-x1+x2$$

```
MatrixForm[{{y23, 0, y31, 0, y12, 0}, {0, x32, 0, x13, 0, x21}, {x32, y23, x13, y31, x21, y12}}]
```

$$\begin{pmatrix} y2-y3 & 0 & -y1+y3 & 0 & y1-y2 & 0 \\ 0 & -x2+x3 & 0 & x1-x3 & 0 & -x1+x2 \\ -x2+x3 & y2-y3 & x1-x3 & -y1+y3 & -x1+x2 & y1-y2 \end{pmatrix}$$

```
Be = {{y23, 0, y31, 0, y12, 0}, {0, x32, 0, x13, 0, x21}, {x32, y23, x13, y31, x21, y12}} / (2 * A)
```

$$\left\{\left\{\frac{y2-y3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ 0,\ \frac{-y1+y3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\right.\right.$$
$$\left.0,\ \frac{y1-y2}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ 0\right\},\ \left\{0,\ \frac{-x2+x3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\right.$$
$$\left.\left.0,\ \frac{x1-x3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ 0,\ \frac{-x1+x2}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)}\right\},\right.$$
$$\left\{\frac{-x2+x3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ \frac{y2-y3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\right.$$
$$\frac{x1-x3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ \frac{-y1+y3}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},$$
$$\left.\left.\frac{-x1+x2}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)},\ \frac{y1-y2}{x3\,(y1-y2)+x1\,(y2-y3)+x2\,(-y1+y3)}\right\}\right\}$$

### EJEMPLO UTILIZACION

```
ClearAll[a, b, h, Em]; a = 2; b = 3; Em = 120; h = 1;
```

```
MatrixForm[{{Em, 0, 0}, {0, Em, 0}, {0, 0, Em / 2}}]
```

$$\begin{pmatrix} 120 & 0 & 0 \\ 0 & 120 & 0 \\ 0 & 0 & 60 \end{pmatrix}$$

```
StiffnessOf3NodePlaneStressTriangle[{{0, 0}, {a, 0}, {a, b}},
 {{Em, 0, 0}, {0, Em, 0}, {0, 0, Em / 2}}, {h}]
```

```
{{90, 0, -90, 0, 0, 0}, {0, 45, 30, -45, -30, 0}, {-90, 30, 110, -30, -20, 0},
 {0, -45, -30, 85, 30, -40}, {0, -30, -20, 30, 20, 0}, {0, 0, 0, -40, 0, 40}}
```

```
MatrixForm[%]
```

$$\begin{pmatrix} 90 & 0 & -90 & 0 & 0 & 0 \\ 0 & 45 & 30 & -45 & -30 & 0 \\ -90 & 30 & 110 & -30 & -20 & 0 \\ 0 & -45 & -30 & 85 & 30 & -40 \\ 0 & -30 & -20 & 30 & 20 & 0 \\ 0 & 0 & 0 & -40 & 0 & 40 \end{pmatrix}$$

## ■ 2 - ENSAMBLADO MATRICES CONJUNTO ELEMENTOS

```
AssembleMasterStiffOfPlaneStressModel[xycoords_, elenod_, elemat_, elefab_] :=
  Module[{numele = Length[elenod], numnod = Length[xycoords], ne, n1, n2, n3,
    eftab, Em, nu, fnu, Emat, h, Ke, K}, K = Table[0, {2 * numnod}, {2 * numnod}];
   For[ne = 1, ne ≤ numele, ne++, {n1, n2, n3} = elenod[[ne]];
    eftab = {2 * n1 - 1, 2 * n1, 2 * n2 - 1, 2 * n2, 2 * n3 - 1, 2 * n3};
    h = elefab[[ne]]; {Em, nu} = elemat[[ne]];
    fnu = 1 - nu^2;     Print["Elem=", ne, "  nodes=", {n1, n2, n3}, "  eftab=", eftab];
    Emat = {{Em, Em * nu, 0} / fnu, {Em * nu, Em, 0} / fnu, {0, 0, Em / (2 * (1 + nu))}};
Ke = StiffnessOf3NodePlaneStressTriangle[{xycoords[[n1]], xycoords[[n2]], xycoords[[n3]]}, Emat, {h}];
K = MergeElemIntoMasterStiff[Ke, eftab, K];
];
Return[K];
];

MergeElemIntoMasterStiff[Ke_, eftab_, Km_] := Module[{i, j, ii, jj, neldof = Length[Ke], K}, K = Km;
   For[i = 1, i ≤ neldof, i++, ii = eftab[[i]];
    For[j = i, j ≤ neldof, j++, jj = eftab[[j]];
     K[[jj, ii]] = K[[ii, jj]] += Ke[[i, j]];];]; Return[K];];
```

### □ EJEMPLO UTILIZACION

```
K = AssembleMasterStiffOfPlaneStressModel[
 {{0, 1}, {0, 0}, {2, 1}, {2, 0}}, {{1, 2, 4}, {1, 4, 3}},
  {{12, 1 / 4}, {6, 1 / 2}}, {1, 1} ];
Print["Master Stiffness of Plane Stress Model:"];
Print[K // MatrixForm];
Print[Chop[Eigenvalues[N[K]]]];
```

```
Elem=1  nodes={1, 2, 4}  eftab={1, 2, 3, 4, 7, 8}

Elem=2  nodes={1, 4, 3}  eftab={1, 2, 7, 8, 5, 6}

Master Stiffness of Plane Stress Model:
```

$$\begin{pmatrix} \frac{34}{5} & 0 & -\frac{24}{5} & -\frac{12}{5} & -2 & -2 & 0 & \frac{22}{5} \\ 0 & \frac{133}{10} & -\frac{8}{5} & -\frac{64}{5} & -1 & -\frac{1}{2} & \frac{13}{5} & 0 \\ -\frac{24}{5} & -\frac{8}{5} & 8 & 4 & 0 & 0 & -\frac{16}{5} & -\frac{12}{5} \\ -\frac{12}{5} & -\frac{64}{5} & 4 & 14 & 0 & 0 & -\frac{8}{5} & -\frac{6}{5} \\ -2 & -1 & 0 & 0 & 4 & 3 & -2 & -2 \\ -2 & -\frac{1}{2} & 0 & 0 & 3 & \frac{17}{2} & -1 & -8 \\ 0 & \frac{13}{5} & -\frac{16}{5} & -\frac{8}{5} & -2 & -1 & \frac{26}{5} & 0 \\ \frac{22}{5} & 0 & -\frac{12}{5} & -\frac{6}{5} & -2 & -8 & 0 & \frac{46}{5} \end{pmatrix}$$

```
{28.756, 19.7763, 10.2834, 6.60581, 3.57846, 0, 0, 0}
```

## 3 - APLICACION CONDICIONES DE CONTORNO DE DESPLAZAMIENTOS Y FUERZAS

```
ModifyMasterStiffForDBC[pdof_, Km_] := Module[{i, j, k, K}, K = Km;
    For[k = 1, k ≤ Length[pdof], k++, i = pdof[[k]];
     For[j = 1, j ≤ Length[K], j++, K[[i, j]] = K[[j, i]] = 0]; K[[i, i]] = 1;]; Return[K];];
ModifyNodeForcesForDBC[pdof_, pdofv_, Km_, nfv_] :=
  Module[{i, j, k, l, d, kk = Length[pdof], n = Length[Km], fixed, rhs},
   rhs = nfv; d = pdofv; fixed = Table[False, {n}];
   Do[i = pdof[[k]]; fixed[[i]] = True, {k, 1, kk}];
   For[k = 1, k ≤ kk, k++, i = pdof[[k]];
    For[j = 1, j ≤ n, j++, If[fixed[[j]], Continue[]];
     rhs[[j]] = rhs[[j]] - Km[[i, j]] * d[[k]];]; rhs[[i]] = d[[k]];];
   Return[rhs];];
```

### EJEMPLO UTILIZACION

```
K = Array[Ks, {6, 6}];
Print["Master stiffness matrix:"]; Print[K // TableForm];
p = Array[ps, {6}];
Print["Node force vector:"]; Print[p];
p = ModifyNodeForcesForDBC[{1, 2, 4}, {d1, d2, d3}, K, p];
Print["Node Force vector modified for displacement B.C.:"];
Print[p];
K = ModifyMasterStiffForDBC[{1, 2, 4}, K];
Print["Master stiffness modified for displacement B.C.:"];
Print[K // TableForm];
```

Master stiffness matrix:

```
Ks[1, 1]    Ks[1, 2]    Ks[1, 3]    Ks[1, 4]    Ks[1, 5]    Ks[1, 6]
Ks[2, 1]    Ks[2, 2]    Ks[2, 3]    Ks[2, 4]    Ks[2, 5]    Ks[2, 6]
Ks[3, 1]    Ks[3, 2]    Ks[3, 3]    Ks[3, 4]    Ks[3, 5]    Ks[3, 6]
Ks[4, 1]    Ks[4, 2]    Ks[4, 3]    Ks[4, 4]    Ks[4, 5]    Ks[4, 6]
Ks[5, 1]    Ks[5, 2]    Ks[5, 3]    Ks[5, 4]    Ks[5, 5]    Ks[5, 6]
Ks[6, 1]    Ks[6, 2]    Ks[6, 3]    Ks[6, 4]    Ks[6, 5]    Ks[6, 6]
```

Node force vector:

```
{ps[1], ps[2], ps[3], ps[4], ps[5], ps[6]}
```

Node Force vector modified for displacement B.C.:

```
{d1, d2, -d1 Ks[1, 3] - d2 Ks[2, 3] - d3 Ks[4, 3] + ps[3], d3,
 -d1 Ks[1, 5] - d2 Ks[2, 5] - d3 Ks[4, 5] + ps[5], -d1 Ks[1, 6] - d2 Ks[2, 6] - d3 Ks[4, 6] + ps[6]}
```

Master stiffness modified for displacement B.C.:

```
1    0    0           0    0           0
0    1    0           0    0           0
0    0    Ks[3, 3]    0    Ks[3, 5]    Ks[3, 6]
0    0    0           1    0           0
0    0    Ks[5, 3]    0    Ks[5, 5]    Ks[5, 6]
0    0    Ks[6, 3]    0    Ks[6, 5]    Ks[6, 6]
```

## 4 - CALCULO TENSIONES EN UN ELEMENTO TRIANGULAR Y EN UN CONJUNTO DE ELEMENTOS

```
StressesIn3NodePlaneStressTriangle[{{x1_, y1_}, {x2_, y2_}, {x3_, y3_}}, Emat_, ue_] :=
  Module[{x21, x13, x32, y12, y31, y23, A, e, sige},
   A = Simplify[((x2 * y3 - x3 * y2) + (x3 * y1 - x1 * y3) + (x1 * y2 - x2 * y1)) / 2];
   {x21, x13, x32, y12, y31, y23} = {x2 - x1, x1 - x3, x3 - x2, y1 - y2, y3 - y1, y2 - y3};
   e = {{y23, 0, y31, 0, y12, 0}, {0, x32, 0, x13, 0, x21}, {x32, y23, x13, y31, x21, y12}}.ue / (2 * A);
   sige = Simplify[Emat.e];
   Return[sige];];

StressesInPlaneStressModel[xycoords_, elenod_, elemat_, elefab_, u_] :=
  Module[{numele = Length[elenod], numnod = Length[xycoords], i, ii, ne, n1, n2, n3, eftab,
    Em, nu, fnu, Emat, ue = Table[0, {6}], sige, sig}, sig = Table[{0, 0, 0}, {numele}];
   For[ne = 1, ne ≤ numele, ne++, {n1, n2, n3} = elenod[[ne]];
    eftab = {2 * n1 - 1, 2 * n1, 2 * n2 - 1, 2 * n2, 2 * n3 - 1, 2 * n3};
    Do[ii = eftab[[i]]; ue[[i]] = u[[ii]], {i, 1, 6}];
    {Em, nu} = elemat[[ne]]; fnu = 1 - nu^2;
    Emat = {{Em, Em * nu, 0} / fnu, {Em * nu, Em, 0} / fnu, {0, 0, Em / (2 * (1 + nu))}};
    sig[[ne]] =
     StressesIn3NodePlaneStressTriangle[{xycoords[[n1]], xycoords[[n2]], xycoords[[n3]]}, Emat, ue];
   Return[
    sig];];
```

### EJEMPLO UTILIZACION

```
ClearAll[a, b, h, Em]; a = 2; b = 3; Em = 1000; exx = 0.15; eyy = -0.125;
sige = StressesIn3NodePlaneStressTriangle[{{0, 0}, {a, 0}, {a, b}},
  {{Em, 0, 0}, {0, Em, 0}, {0, 0, Em / 2}}, {0, 0, exx * a, 0, exx * a, eyy * b}];
Print[Chop[N[sige]]];
```

```
{150., -125., 0}
```

## 5 - SOLUCION DE LAS ECUACIONES DE UN MODELO COMPLETO

```
LinearSolutionOfPlaneStressModel[xycoords_, elenod_, elemat_, elefab_, doftag_, dofval_] :=
  Module[{K, Kmod, u, f, j, n, ns, supdof, supval, numnod = Length[xycoords], numele = Length[elenod]},
   K = AssembleMasterStiffOfPlaneStressModel[xycoords, elenod, elemat, elefab];
   K = N[K];
   ns = 0; Do[Do[If[doftag[[n, j]] > 0, ns++], {j, 1, 2}], {n, 1, numnod}];
   supdof = supval = Table[0, {ns}];
   k = 0; Do[Do[If[doftag[[n, j]] > 0, k++; supdof[[k]] = 2 * (n - 1) + j;
       supval[[k]] = dofval[[n, j]]], {j, 1, 2}], {n, 1, numnod}]; Print["supval=", supval];
   Print["supdof=", supdof]; f = ModifyNodeForcesForDBC[supdof, supval, K, Flatten[dofval]];
 Print["f=", f]; Kmod = ModifyMasterStiffForDBC[supdof, K];
u = Simplify[Inverse[Kmod].f]; u = Chop[u];
   f = Simplify[K.u]; f = Chop[f];
   sig = StressesInPlaneStressModel[xycoords, elenod, elemat, elefab, u];
sig = Chop[sig];
Return[{u, f, sig}];
];
```

## 6 - VISUALIZACION DE LA MALLA DE ELEMENTOS FINITOS, RESULTADOS NODALES, RESULTADOS EN ELEMENTOS, RESULTADOS EN ELEMENTOS TRIANGULAR Y EN ELEMENTO CUADRADO

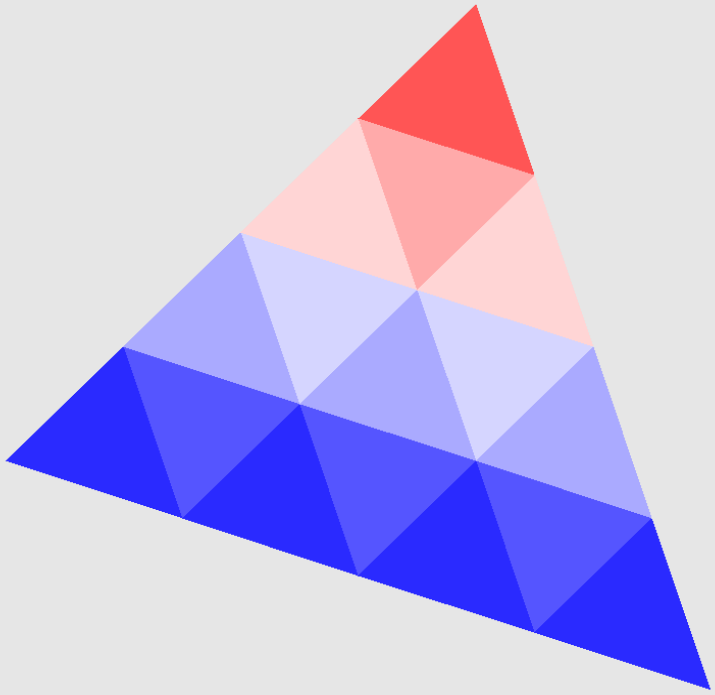
```
Plot2DMesh[xycoords_, elenod_, aspect_, label_] := Module[
   {eNL, n, nc, xyc, poly = sides = {}, numele = Length[elenod]}, Do[eNL = elenod[[ne]]; nc = Length[eNL];
    If[nc ≠ 3 && nc ≠ 4, Continue[]]; xyc = Table[0, {nc + 1}];
    eNL = AppendTo[eNL, eNL[[1]]];  Do[n = eNL[[i]]; xyc[[i]] = xycoords[[n]], {i, 1, nc + 1}];
  poly = AppendTo[poly, Graphics[Polygon[Take[xyc, nc]]]];
sides = AppendTo[sides, Graphics[Line[xyc]]], {ne, 1, numele}];
Show[Graphics[RGBColor[1, 1, 0]], poly, Graphics[Thickness[0.008]],
    Graphics[RGBColor[0, 0, 0]], sides, AspectRatio → aspect, PlotLabel → label]
];

ContourPlotNodeFuncOver2DMesh[xycoords_, elenod_, f_, fmax_, Nsub_, aspect_, label_] :=
  Module[{eNL, n, nc, xyc, fc, poly = {}, p},
   For[ne = 1, ne ≤ Length[elenod], ne++, eNL = elenod[[ne]]; nc = Length[eNL];
    If[nc ≠ 3 && nc ≠ 4, Continue[]];
    fc = xyc = Table[0, {nc}];
    Do[n = eNL[[i]]; xyc[[i]] = xycoords[[n]]; fc[[i]] = f[[n]], {i, 1, nc}];
    If[nc == 3, p = PlotFunctionOverTriangle[xyc, fc, fmax, Nsub]];
    If[nc == 4, p = PlotFunctionOverQuadrilateral[xyc, fc, fmax, Nsub]];
    poly = Join[poly, p]];
   Show[poly, AspectRatio → aspect, PlotLabel → label]];

ContourPlotElemFuncOver2DMesh[xycoords_, elenod_, fe_, fmax_, Nsub_, aspect_, label_] :=
  Module[{eNL, n, nc, xyc, fc, poly = {}, p},
   For[ne = 1, ne ≤ Length[elenod], ne++, eNL = elenod[[ne]]; nc = Length[eNL];
    If[nc ≠ 3 && nc ≠ 4, Continue[]];
    fc = xyc = Table[fe[[ne]], {nc}];
    Do[n = eNL[[i]]; xyc[[i]] = xycoords[[n]], {i, 1, nc}];
    Print[{ne, xyc, fc}]; If[nc == 3, p = PlotFunctionOverTriangle[xyc, fc, fmax, 1]];
If[nc == 4, p = PlotFunctionOverQuadrilateral[xyc, fc, fmax, 1]];
poly = Join[poly, p]];
Show[poly, AspectRatio → aspect, PlotLabel → label]
];

PlotFunctionOverTriangle[xyc_, fc_, fmax_, Nsub_] :=
  Module[{Ni, zc1, zc2, zc3, xc, yc, x1, x2, x3, y1, y2, y3, iz1, iz2, iz3,
    c1, c2, c3, d, f1, f2, f3, f, poly = {}}, {{x1, y1}, {x2, y2}, {x3, y3}} = xyc;
   xc = {x1, x2, x3}; yc = {y1, y2, y3}; {f1, f2, f3} = fc; Ni = Nsub * 3;
   Do[Do[iz3 = Ni - iz1 - iz2; If[iz3 ≤ 0, Continue[]]; d = 0;
```

```
      If[Mod[iz1 - 2, 3] == 0 && Mod[iz2 + 1, 3] == 0, d = -1];
      If[d == 0, Continue[]];
      zc1 = N[{iz1 + d + d, iz2 - d, iz3 - d} / Ni];
      zc2 = N[{iz1 - d, iz2 + d + d, iz3 - d} / Ni];
      zc3 = N[{iz1 - d, iz2 - d, iz3 + d + d} / Ni];
      f = N[(f1 * iz1 + f2 * iz2 + f3 * iz3) / Ni];
      {c1, c2, c3} = ContourPolyColor[f, fmax];
      AppendTo[poly, Graphics[RGBColor[c1, c2, c3]]];
      AppendTo[poly, Graphics[Polygon[{{xc.zc1, yc.zc1}, {xc.zc2, yc.zc2}, {xc.zc3, yc.zc3}}]]],
      {iz2, 1, Ni - iz1}], {iz1, 1, Ni}];
    Return[poly];];

PlotFunctionOverQuadrilateral[xyc_, fc_, fmax_, Nsub_] :=
  Module[{Ne, Nev, xy1, xy2, xy3, i, j, n, ixi, ieta, xi, eta, x1, x2, x3, x4, y1, y2, y3, y4, xc, yc,
    c1, c2, c3, d, f1, f2, f3, f4, f, poly = {}}, {{x1, y1}, {x2, y2}, {x3, y3}, {x4, y4}} = xyc;
   xc = {x1, x2, x3, x4}; yc = {y1, y2, y3, y4}; {f1, f2, f3, f4} = fc;
   Ne[xi_, eta_] := N[{(1 - xi) * (1 - eta), (1 + xi) * (1 - eta), (1 + xi) * (1 + eta), (1 - xi) * (1 + eta)} / 4];
   n = Nsub;
   Do[Do[ixi = (2 * i - n - 1) / n; ieta = (2 * j - n - 1) / n;
     {xi, eta} = N[{ixi - 1 / n, ieta - 1 / n}]; Nev = Ne[xi, eta];
     xy1 = {xc.Nev, yc.Nev};
     {xi, eta} = N[{ixi + 1 / n, ieta - 1 / n}]; Nev = Ne[xi, eta];
     xy2 = {xc.Nev, yc.Nev};
     {xi, eta} = N[{ixi + 1 / n, ieta + 1 / n}]; Nev = Ne[xi, eta];
     xy3 = {xc.Nev, yc.Nev};
     {xi, eta} = N[{ixi - 1 / n, ieta + 1 / n}]; Nev = Ne[xi, eta];
     xy4 = {xc.Nev, yc.Nev};
     Nev = Ne[N[ixi], N[ieta]];
     {c1, c2, c3} = ContourPolyColor[fc.Nev, fmax];
     AppendTo[poly, Graphics[RGBColor[c1, c2, c3]]];
     AppendTo[poly, Graphics[Polygon[{xy1, xy2, xy3, xy4}]]], {i, 1, Nsub}], {j, 1, Nsub}];
   Return[poly];];

ContourPolyColor[f_, fmax_] :=
  Module[{r, RGBmax = {1, 0, 0}, RGBmin = {0, 0, 1}, RGBzero = {1, 1, 1}, RGBout = {0, 0, 0}},
   If[f == 0 || fmax == 0, Return[RGBzero]]; (*White if f=0*)If[f > fmax || f < -fmax, Return[RGBout]];
   (*Black if outside range*)If[f > 0, r = N[f / fmax]; Return[r * RGBmax + (1 - r) * RGBzero]];
   (*positive*)If[f < 0, r = -N[f / fmax]; Return[r * RGBmin + (1 - r) * RGBzero]];(*negative*)];
```

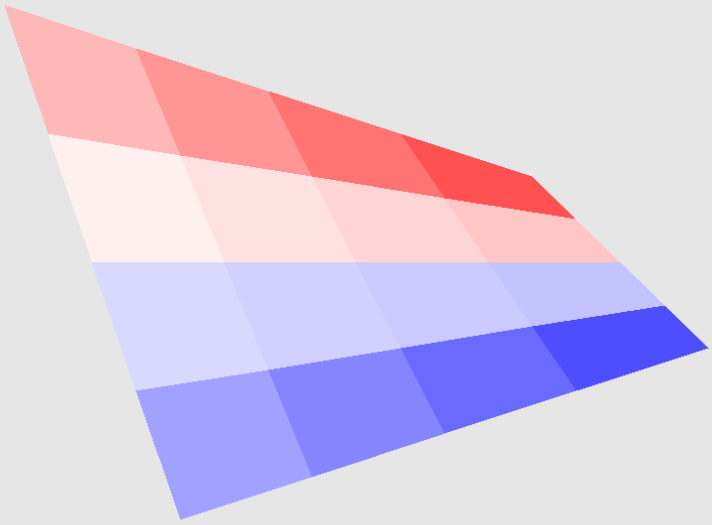
```
Off[General::"spell1"]
Off[General::"spell"]
```

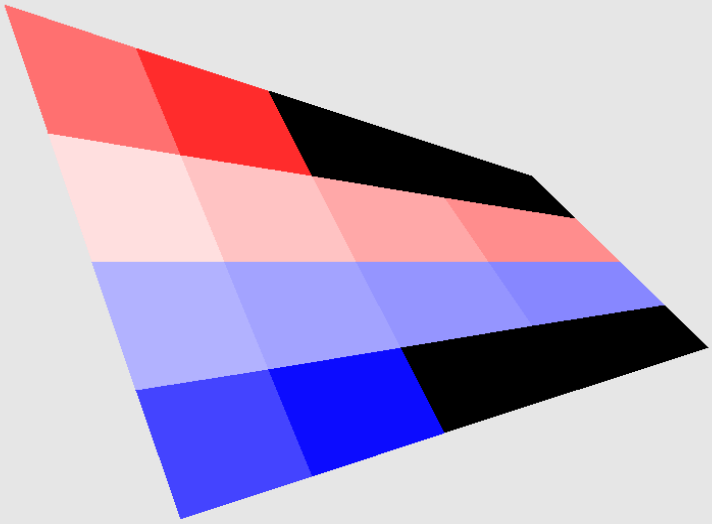
## EJEMPLOS UTILIZACION

```
xyc1 = {0, 0}; xyc2 = {3, -1}; xyc3 = {2, 2};
xyc = N[{xyc1, xyc2, xyc3}]; Nsub = 4;
poly = PlotFunctionOverTriangle[xyc, {-10, -10, 10}, 10, Nsub];
Show[poly]
```



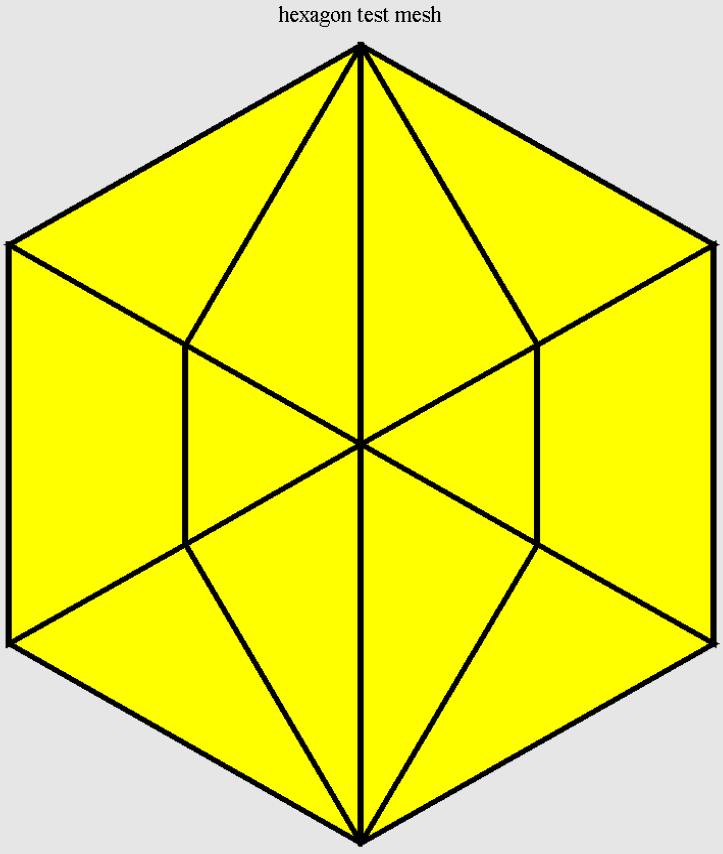
```
xyc1 = {0, 0}; xyc2 = {3, 1}; xyc3 = {2, 2}; xyc4 = {-1, 3};
xyc = N[{xyc1, xyc2, xyc3, xyc4}]; Nsub = 4;
poly = PlotFunctionOverQuadrilateral[xyc, {-4, -10, 10, 3}, 10, Nsub]; Show[poly]
poly = PlotFunctionOverQuadrilateral[xyc, {-4, -10, 10, 3}, 5, Nsub]; Show[poly]
```



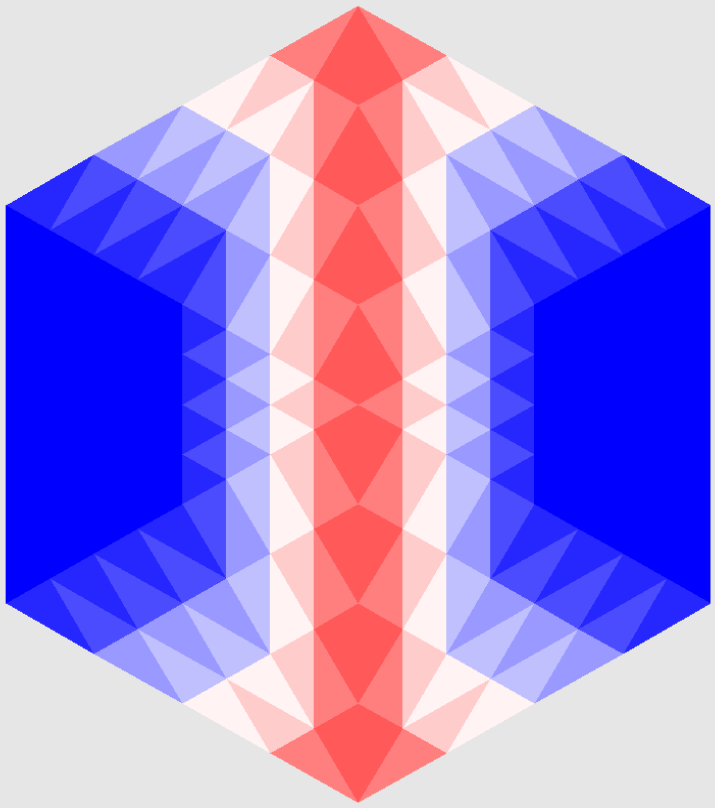


```
xycoords = N[{{0, 10}, {-8.66, 5}, {-8.66, -5}, {0, -10}, {8.66, -5}, {8.66, 5},
    {-4.33, 2.5}, {-4.33, -2.5}, {4.33, -2.5}, {4.33, 2.5}, {0, 0}}];
elenod =  {{1, 2, 7}, {2, 3, 8, 7}, {3, 4, 8}, {1, 7, 11}, {7, 8, 11}, {8, 4, 11}, {1, 11, 10},
    {11, 9, 10}, {11, 4, 9}, {1, 10, 6}, {10, 9, 5, 6}, {4, 5, 9}};
```

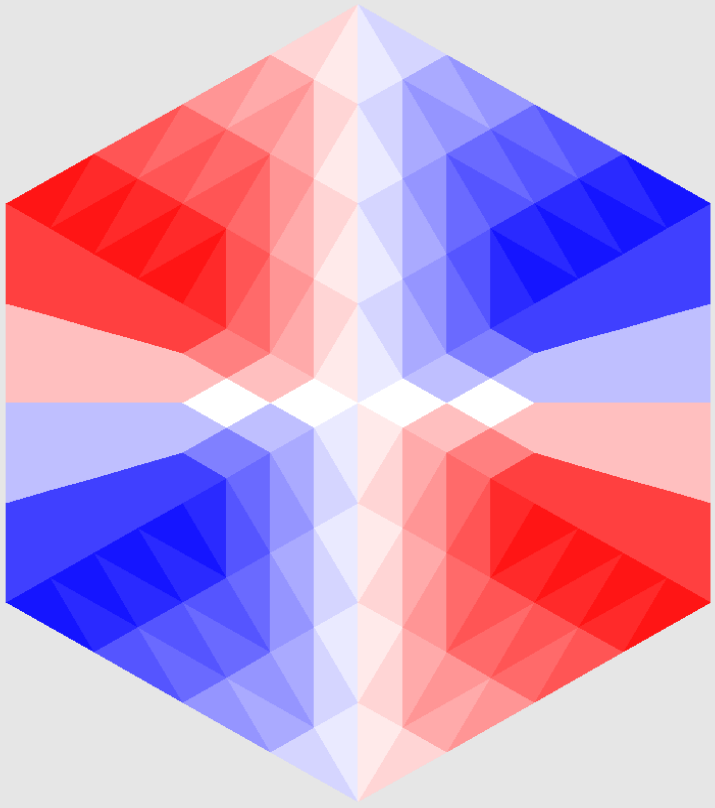
```
(*  Test mesh and node-defined contour plot *)
numnod = Length[xycoords]; numele = Length[elenod];
sigxx = sigyy = sigxy = Table[0, {numnod}]; sxxmax = syymax = sxymax = 0;
Do [{x, y} = xycoords[[n]]; r = Sqrt[x^2 + y^2];
  s = c = 0; If [r > 0, s = y / r; c = x / r];
  {sigxx[[n]], sigyy[[n]], sigxy[[n]]} = N[{0.25 - c^4,
     0.125 - s^2 * c^2, -s * c^3}];
  sxxmax = Max[sxxmax, Abs[sigxx[[n]]]];
  syymax = Max[syymax, Abs[sigyy[[n]]]];
  sxymax = Max[sxymax, Abs[sigxy[[n]]]],
{n, 1, numnod}];
aspect = 10 / 8.66;
Plot2DMesh[xycoords, elenod, aspect, "hexagon test mesh"]
ContourPlotNodeFuncOver2DMesh[xycoords, elenod, sigxx, sxxmax, 4,
  aspect, "sigxx"]
ContourPlotNodeFuncOver2DMesh[xycoords, elenod, sigxy, sxymax, 4,
  aspect, "sigxy"]
```



sigxx

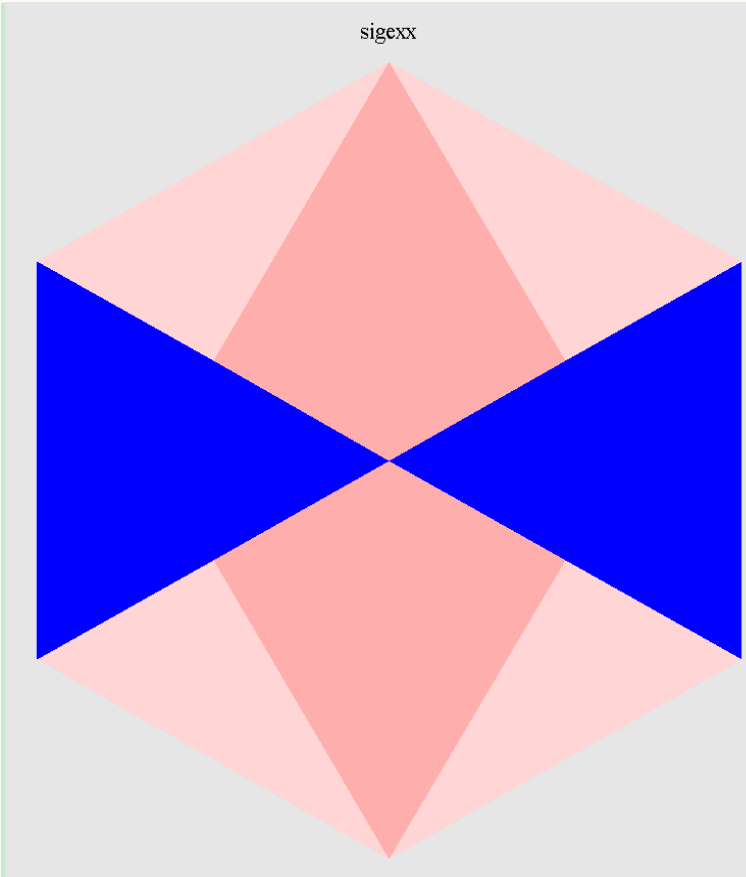


sigxy

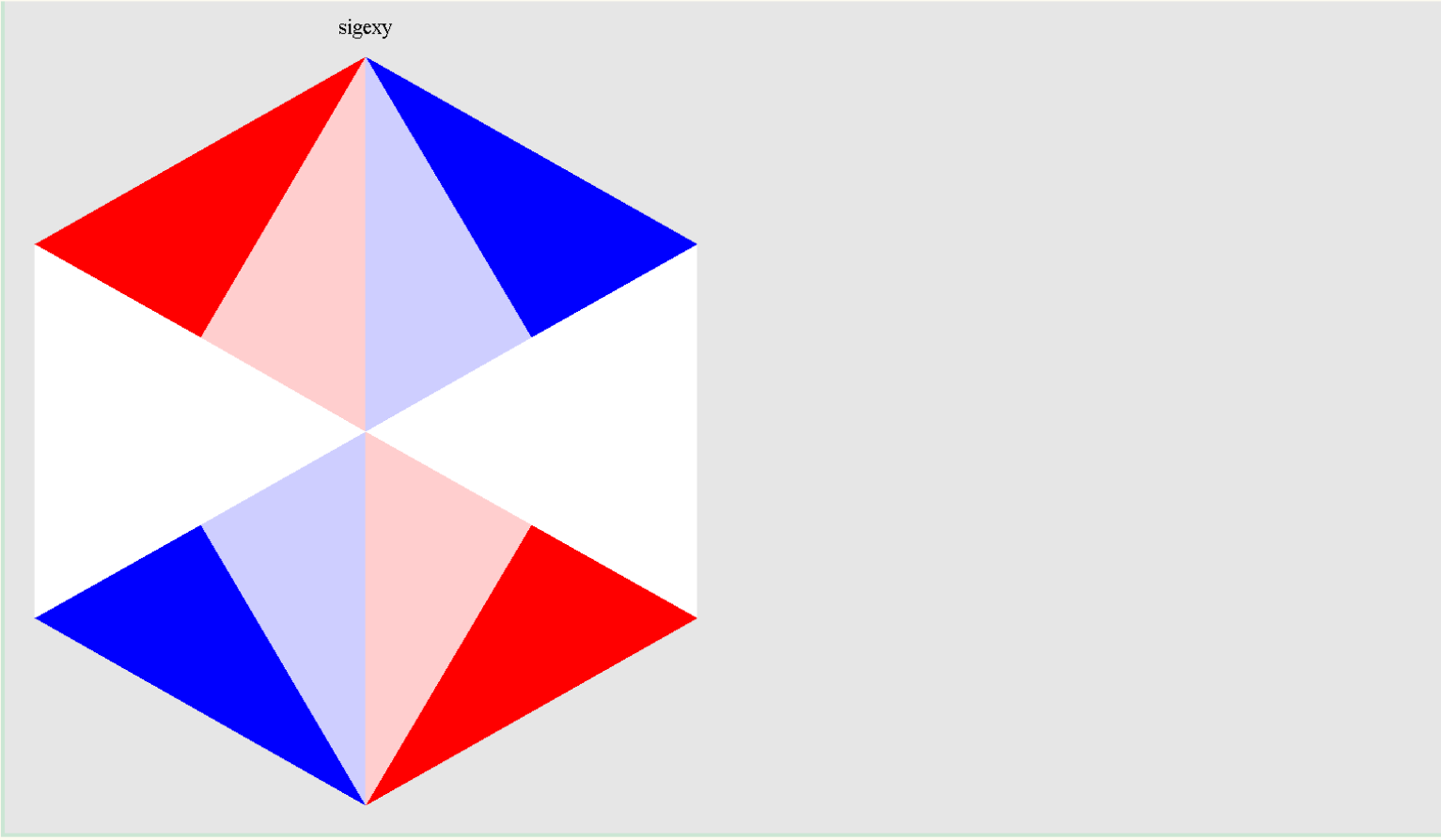


```
(*  Test element-defined contour plot *)
sigexx = sigeyy = sigexy = Table[0, {numele}]; sxxemax = syyemax = sxyemax = 0;
Do [eNL = elenod[[ne]]; nc = Length[eNL]; x0 = y0 = 0;
  Do [n = eNL[[i]];  {x0, y0} += xycoords[[n]], {i, 1, nc}];
   {x0, y0} = {x0, y0} / nc; r = Sqrt[x0^2 + y0^2];
   s = c = 0; If [r > 0, s = y0 / r; c = x0 / r];
   {sigexx[[ne]], sigeyy[[ne]], sigexy[[ne]]} = N[{0.25 - c^4,
       0.125 - s^2 * c^2, -s * c^3}];
   sxxemax = Max[sxxemax, Abs[sigexx[[ne]]]];
   syyemax = Max[syyemax, Abs[sigeyy[[ne]]]];
   sxyemax = Max[sxyemax, Abs[sigexy[[ne]]]],
{ne, 1, numele}];
ContourPlotElemFuncOver2DMesh[xycoords, elenod, sigexx, sxxemax, 1,
  aspect, "sigexx"]
ContourPlotElemFuncOver2DMesh[xycoords, elenod, sigexy, sxyemax, 1,
  aspect, "sigexy"]
```

```
{1, {{0., 10.}, {-8.66, 5.}, {-4.33, 2.5}}, {0.123798, 0.123798, 0.123798}}

{2, {{-8.66, 5.}, {-8.66, -5.}, {-4.33, -2.5}, {-4.33, 2.5}}, {-0.75, -0.75, -0.75, -0.75}}

{3, {{-8.66, -5.}, {0., -10.}, {-4.33, -2.5}}, {0.123798, 0.123798, 0.123798}}

{4, {{0., 10.}, {-4.33, 2.5}, {0., 0.}}, {0.238522, 0.238522, 0.238522}}

{5, {{-4.33, 2.5}, {-4.33, -2.5}, {0., 0.}}, {-0.75, -0.75, -0.75}}

{6, {{-4.33, -2.5}, {0., -10.}, {0., 0.}}, {0.238522, 0.238522, 0.238522}}

{7, {{0., 10.}, {0., 0.}, {4.33, 2.5}}, {0.238522, 0.238522, 0.238522}}

{8, {{0., 0.}, {4.33, -2.5}, {4.33, 2.5}}, {-0.75, -0.75, -0.75}}

{9, {{0., 0.}, {0., -10.}, {4.33, -2.5}}, {0.238522, 0.238522, 0.238522}}

{10, {{0., 10.}, {4.33, 2.5}, {8.66, 5.}}, {0.123798, 0.123798, 0.123798}}

{11, {{4.33, 2.5}, {4.33, -2.5}, {8.66, -5.}, {8.66, 5.}}, {-0.75, -0.75, -0.75, -0.75}}

{12, {{0., -10.}, {8.66, -5.}, {4.33, -2.5}}, {0.123798, 0.123798, 0.123798}}
```



```
{1, {{0., 10.}, {-8.66, 5.}, {-4.33, 2.5}}, {0.170019, 0.170019, 0.170019}}

{2, {{-8.66, 5.}, {-8.66, -5.}, {-4.33, -2.5}, {-4.33, 2.5}}, {0., 0., 0., 0.}}

{3, {{-8.66, -5.}, {0., -10.}, {-4.33, -2.5}}, {-0.170019, -0.170019, -0.170019}}

{4, {{0., 10.}, {-4.33, 2.5}, {0., 0.}}, {0.0331362, 0.0331362, 0.0331362}}

{5, {{-4.33, 2.5}, {-4.33, -2.5}, {0., 0.}}, {0., 0., 0.}}

{6, {{-4.33, -2.5}, {0., -10.}, {0., 0.}}, {-0.0331362, -0.0331362, -0.0331362}}

{7, {{0., 10.}, {0., 0.}, {4.33, 2.5}}, {-0.0331362, -0.0331362, -0.0331362}}

{8, {{0., 0.}, {4.33, -2.5}, {4.33, 2.5}}, {0., 0., 0.}}

{9, {{0., 0.}, {0., -10.}, {4.33, -2.5}}, {0.0331362, 0.0331362, 0.0331362}}

{10, {{0., 10.}, {4.33, 2.5}, {8.66, 5.}}, {-0.170019, -0.170019, -0.170019}}

{11, {{4.33, 2.5}, {4.33, -2.5}, {8.66, -5.}, {8.66, 5.}}, {0., 0., 0., 0.}}

{12, {{0., -10.}, {8.66, -5.}, {4.33, -2.5}}, {0.170019, 0.170019, 0.170019}}
```

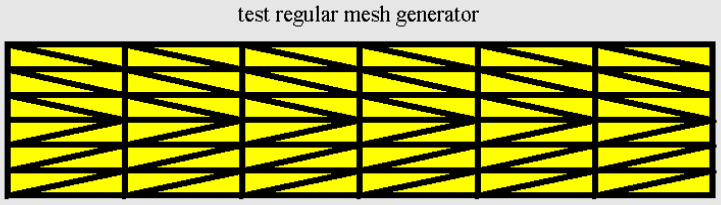


## 7 - EJEMPLO DE VIGA EN VOLADIZO - GENERACION DE NODOS, TRIANGULOS Y FUERZAS APLICADAS

```
GenerateNodes[Lx_, Ly_, nx_, ny_] :=
  Module[{k, i, j, dx, dy, nxy, numnod}, numnod = (nx + 1) * (ny + 1); nxy = Table[0, {numnod}];
   dx = Lx / nx; dy = Ly / ny; k = 0;
   Do[Do[nxy[[++k]] = {dx * (i - 1), dy * (j - 1)}, {j, 1, ny + 1}], {i, 1, nx + 1}];
   Return[nxy]];

GenerateTriangles[nx_, ny_, pat_] := Module[{k, i, j, c1, c2, numele, enl},
   numele = ({2, 2, 4, 2}[[pat]]) * nx * ny; enl = Table[{0, 0, 0}, {numele}]; k = 0;
   Do[Do[c1 = (ny + 1) * (i - 1) + j; c2 = c1 + ny + 1;
     If[pat == 1, enl[[++k]] = {c1, c2, c1 + 1};
      enl[[++k]] = {c2 + 1, c1 + 1, c2}];
     If[pat == 2, enl[[++k]] = {c1, c2, c2 + 1};
      enl[[++k]] = {c2 + 1, c1 + 1, c1}];
     If[pat == 3, enl[[++k]] = {c1, c2, c1 + 1};
      enl[[++k]] = {c2 + 1, c1 + 1, c2};
      enl[[++k]] = {c1, c2, c2 + 1};
      enl[[++k]] = {c2 + 1, c1 + 1, c1}];
     If[pat == 4, If[j ≤ ny / 2, enl[[++k]] = {c1, c2, c2 + 1};
       enl[[++k]] = {c2 + 1, c1 + 1, c1}, enl[[++k]] = {c1, c2, c1 + 1};
       enl[[++k]] = {c2 + 1, c1 + 1, c2}]];
     Continue[], {j, 1, ny}], {i, 1, nx}]; Return[enl]];

GenerateEndShearForces[ny_, numnod_, shear_, shdist_, height_, nfv_] :=
  Module[{dy, n, n1, n2, xi1, xi2, xi3, f1, f2, f3, smax, shfor, f},
   f = nfv; smax = 6 * shear / height; dy = height / ny; xi1 = 0;
   Do[xi2 = xi1 + 1 / (2 * ny); xi3 = xi2 + 1 / (2 * ny);
    n1 = numnod - ny + (n - 1); n2 = n1 + 1;
    If[shdist == "P", f1 = smax * (xi1 - xi1^2); f2 = smax * (xi2 - xi2^2);
     f3 = smax * (xi3 - xi3^2), f1 = f2 = f3 = shear / height];
    f[[n1, 2]] += (f1 + 2 * f2) * dy / 6; f[[n2, 2]] += (2 * f2 + f3) * dy / 6;
    xi1 = xi3, {n, 1, ny}];
   shfor = 0; Do[shfor += f[[n, 2]], {n, numnod - ny, numnod}];
   Print[" Total end shear force= ", shfor]; Return[f]];

GenerateEndMomentForces[ny_, numnod_, moment_, height_, nfv_] := Module[
   {dy, df, n, n1, n2, f1, f2, bmom, f}, f = nfv; f1 = -6 * moment / height^2; df = 2 * f1 / ny; dy = height / ny;
   Do[n1 = numnod - ny + (n - 1); n2 = n1 + 1; f2 = f1 - df;
    f[[n1, 1]] -= (2 * f1 + f2) * dy / 6;
    f[[n2, 1]] -= (f1 + 2 * f2) * dy / 6; f1 = f2, {n, 1, ny}]; yn = height / 2;
   bmom = 0; Do[bmom += f[[n, 1]] * yn; yn -= dy, {n, numnod - ny, numnod}];
   Print[" Total end moment= ", bmom]; Return[f]];

GenerateEndAxialForces[ny_, numnod_, axial_, height_, nfv_] :=
  Module[{df, n, n1, n2, axforce, f}, f = nfv; df = axial / ny;
   Do[n1 = numnod - ny + (n - 1); n2 = n1 + 1;
    f[[n1, 1]] += df / 2; f[[n2, 1]] += df / 2;
    Continue[], {n, 1, ny}];
   axforce = 0; Do[axforce += f[[n, 1]], {n, numnod - ny, numnod}];
   Print[" Total axial force= ", axforce]; Return[f]];
```

## EJEMPLOS NUMERICOS

```
xyc = GenerateNodes[12, 3, 6, 6]; enl = GenerateTriangles[6, 6, 4];
Plot2DMesh[xyc, enl, 3 / 12, "test regular mesh generator"]
Print[GenerateEndShearForces [4, 8, 40., "P", 12, Table[{0, 0}, {8}]]];
Print[GenerateEndMomentForces[4, 8, 96.,  12, Table[{0, 0}, {8}]]];
Print[GenerateEndAxialForces [4, 8, 30.,  12, Table[{0, 0}, {8}]]];
```

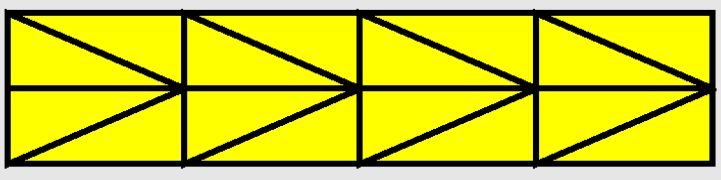



```
 Total end shear force= 40.
{{0, 0}, {0, 0}, {0, 0}, {0, 2.1875}, {0, 10.625}, {0, 14.375}, {0, 10.625}, {0, 2.1875}}
 Total end moment= 96.
{{0, 0}, {0, 0}, {0, 0}, {5., 0}, {6., 0}, {0., 0}, {-6., 0}, {-5., 0}}
 Total axial force= 30.
{{0, 0}, {0, 0}, {0, 0}, {3.75, 0}, {7.5, 0}, {7.5, 0}, {7.5, 0}, {3.75, 0}}
```

```
(*Berkeley cantilever benchmark*)ClearAll[Em, nu, Lx, Ly, nx, ny, nmeshes, P, M, S, loadcase];
nmeshes = 3; Em = 30 000; nu = 0;
nx = 4; ny = 2; Lx = 48.0; Ly = 12.0; aspect = Ly / Lx;
P = 60.; M = 960.; S = 40.; loadcase = "Shear";
```

```
(*Generate FEM model*)
For[nruns = 1, nruns ≤ nmeshes, nruns++, NodeCoordinates = GenerateNodes[Lx, Ly, nx, ny]];
ElemNodeLists = GenerateTriangles[nx, ny, 4];
numnod = Length[NodeCoordinates]; numele = Length[ElemNodeLists];
ElemMaterial = Table[{Em, nu}, {Length[ElemNodeLists]}];
ElemFabrication = Table[1, {Length[ElemNodeLists]}];
FreedomValues = FreedomTags = Table[{0, 0}, {numnod}];
If[loadcase == "Axial", FreedomValues = GenerateEndAxialForces[ny, numnod, P, Ly, FreedomValues]];
If[loadcase == "Moment", FreedomValues = GenerateEndMomentForces[ny, numnod, M, Ly, FreedomValues]];
If[loadcase == "Shear", FreedomValues = GenerateEndShearForces[ny, numnod, S, "P", Ly, FreedomValues]];
Do[FreedomTags[[n]] = {1, 1}, {n, 1, ny + 1}];
Plot2DMesh[NodeCoordinates, ElemNodeLists, aspect,
 "Generated " <> ToString[nx] <> " x " <> ToString[ny] <> " mesh"]
```

```
 Total end shear force= 40.
```




```
(*Solve problem and print results*)
{u, f, sig} = LinearSolutionOfPlaneStressModel[NodeCoordinates, ElemNodeLists,
   ElemMaterial, ElemFabrication, FreedomTags, FreedomValues];
```

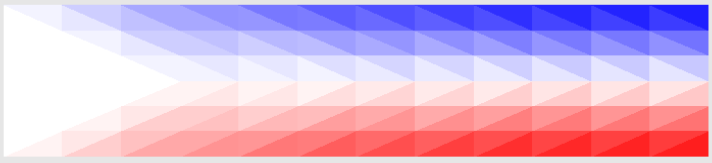
```
Elem=1  nodes={1, 4, 5}  eftab={1, 2, 7, 8, 9, 10}

Elem=2  nodes={5, 2, 1}  eftab={9, 10, 3, 4, 1, 2}

Elem=3  nodes={2, 5, 3}  eftab={3, 4, 9, 10, 5, 6}

Elem=4  nodes={6, 3, 5}  eftab={11, 12, 5, 6, 9, 10}

Elem=5  nodes={4, 7, 8}  eftab={7, 8, 13, 14, 15, 16}

Elem=6  nodes={8, 5, 4}  eftab={15, 16, 9, 10, 7, 8}

Elem=7  nodes={5, 8, 6}  eftab={9, 10, 15, 16, 11, 12}

Elem=8  nodes={9, 6, 8}  eftab={17, 18, 11, 12, 15, 16}

Elem=9  nodes={7, 10, 11}  eftab={13, 14, 19, 20, 21, 22}

Elem=10  nodes={11, 8, 7}  eftab={21, 22, 15, 16, 13, 14}

Elem=11  nodes={8, 11, 9}  eftab={15, 16, 21, 22, 17, 18}

Elem=12  nodes={12, 9, 11}  eftab={23, 24, 17, 18, 21, 22}

Elem=13  nodes={10, 13, 14}  eftab={19, 20, 25, 26, 27, 28}

Elem=14  nodes={14, 11, 10}  eftab={27, 28, 21, 22, 19, 20}

Elem=15  nodes={11, 14, 12}  eftab={21, 22, 27, 28, 23, 24}

Elem=16  nodes={15, 12, 14}  eftab={29, 30, 23, 24, 27, 28}

supval={0, 0, 0, 0, 0, 0}

supdof={1, 2, 3, 4, 5, 6}

f={0, 0, 0, 0, 0, 0, 0., 0., 0., 0., 0., 0., 0.,
  0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 7.5, 0., 25., 0., 7.5}
```

```
Print["Computed displacements u=", u];
Print["Recovered forces f=", f];
Print["Element stresses sig=", sig];
ntip = numnod - ny / 2;
Print[" Computed tip displacements= ", {u[[2 * ntip - 1]], u[[2 * ntip]]}];
Iz = Ly^3 / 12;
If[loadcase == "Axial", Print[" vs. analytical=", {0, P * Lx / (Em * Ly)}]];
If[loadcase == "Moment", Print[" vs. analytical=", {0, M * Lx^2 / (2 * Em * Iz)}]];
If[loadcase == "Shear",
  Print[" vs. analytical=", {0, S * Lx^3 / (3 * Em * Iz) + ((4 + 5 * nu) / 4) * S * Lx / (Em * 2 * Ly)}]];
```

```
Computed displacements u={0, 0, 0, 0, 0, 0, 0.00937437, 0.0121231, 0, 0.011959, -0.00937437,
  0.0121231, 0.0160417, 0.040207, 0, 0.0400404, -0.0160417, 0.040207, 0.0200414, 0.0789561, 0,
  0.0787907, -0.0200414, 0.0789561, 0.0213548, 0.122979, 0, 0.122893, -0.0213548, 0.122979}

Recovered forces f={-160., 24.8461, 0, -89.6922, 160.,
  24.8461, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7.5, 0, 25., 0, 7.5}

Element stresses sig=
 {{23.4359, -0.820703, -8.28204}, {0, 0, 14.9487}, {0, 0, 14.9487}, {-23.4359, 0.820703, -8.28204},
  {16.6682, -0.833049, -4.99923}, {0, -0.820703, 11.6659}, {0, 0.820703, 11.6659},
  {-16.6682, 0.833049, -4.99923}, {9.99927, -0.827204, -1.66703}, {0, -0.833049, 8.3337},
  {0, 0.833049, 8.3337}, {-9.99927, 0.827204, -1.66703}, {3.28357, -0.429107, 1.64179},
  {0, -0.827204, 5.02488}, {0, 0.827204, 5.02488}, {-3.28357, 0.429107, 1.64179}}

 Computed tip displacements= {0, 0.122893}

 vs. analytical={0, 0.344}
```

```
(*Plot Displacement Component Distribution*)ux = uy = Table[0, {numnod}];
Do[ux[[n]] = u[[2 * n - 1]]; uy[[n]] = u[[2 * n]], {n, 1, numnod}];
uxmax = uymax = 0;
Do[uxmax = Max[Abs[ux[[n]]], uxmax]; uymax = Max[Abs[uy[[n]]], uymax], {n, 1, numnod}];
ContourPlotNodeFuncOver2DMesh[NodeCoordinates,
 ElemNodeLists, ux, uxmax, 3, aspect, "Displacement component ux"]
ContourPlotNodeFuncOver2DMesh[NodeCoordinates, ElemNodeLists,
 uy, uymax, 3, aspect, "Displacement component uy"]
```

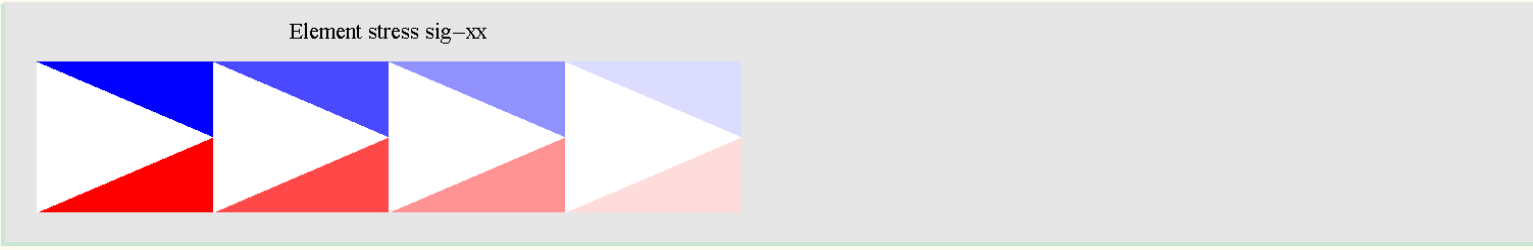
Displacement component ux



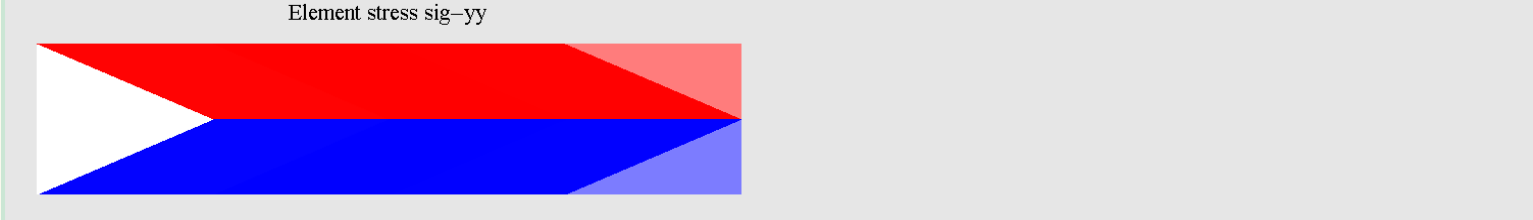
Displacement component uy

```
sxx = syy = sxy = Table[0, {numele}];
Do[{sxx[[ne]], syy[[ne]], sxy[[ne]]} = sig[[ne]], {ne, 1, numele}];
sxxmax = syymax = sxymax = 0;
Do[sxxmax = Max[Abs[sxx[[ne]]], sxxmax]; syymax = Max[Abs[syy[[ne]]], syymax];
   sxymax = Max[Abs[sxy[[ne]]], sxymax], {ne, 1, numele}];
ContourPlotElemFuncOver2DMesh[NodeCoordinates,
 ElemNodeLists, sxx, sxxmax, 1, aspect, "Element stress sig-xx"]
ContourPlotElemFuncOver2DMesh[NodeCoordinates, ElemNodeLists,
 syy, syymax, 1, aspect, "Element stress sig-yy"]
ContourPlotElemFuncOver2DMesh[NodeCoordinates, ElemNodeLists,
 sxy, sxymax, 1, aspect, "Element stress sig-xy"]

nx = 2 * nx; ny = 2 * ny;(*end For loop*)
```

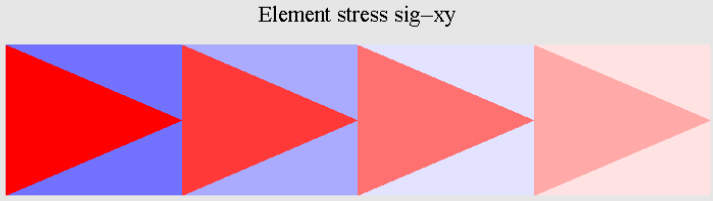
```
{1, {{0., 0.}, {12., 0.}, {12., 6.}}, {23.4359, 23.4359, 23.4359}}

{2, {{12., 6.}, {0., 6.}, {0., 0.}}, {0, 0, 0}}

{3, {{0., 6.}, {12., 6.}, {0., 12.}}, {0, 0, 0}}

{4, {{12., 12.}, {0., 12.}, {12., 6.}}, {-23.4359, -23.4359, -23.4359}}

{5, {{12., 0.}, {24., 0.}, {24., 6.}}, {16.6682, 16.6682, 16.6682}}

{6, {{24., 6.}, {12., 6.}, {12., 0.}}, {0, 0, 0}}

{7, {{12., 6.}, {24., 6.}, {12., 12.}}, {0, 0, 0}}

{8, {{24., 12.}, {12., 12.}, {24., 6.}}, {-16.6682, -16.6682, -16.6682}}

{9, {{24., 0.}, {36., 0.}, {36., 6.}}, {9.99927, 9.99927, 9.99927}}

{10, {{36., 6.}, {24., 6.}, {24., 0.}}, {0, 0, 0}}

{11, {{24., 6.}, {36., 6.}, {24., 12.}}, {0, 0, 0}}

{12, {{36., 12.}, {24., 12.}, {36., 6.}}, {-9.99927, -9.99927, -9.99927}}

{13, {{36., 0.}, {48., 0.}, {48., 6.}}, {3.28357, 3.28357, 3.28357}}

{14, {{48., 6.}, {36., 6.}, {36., 0.}}, {0, 0, 0}}

{15, {{36., 6.}, {48., 6.}, {36., 12.}}, {0, 0, 0}}

{16, {{48., 12.}, {36., 12.}, {48., 6.}}, {-3.28357, -3.28357, -3.28357}}
```




```
{1, {{0., 0.}, {12., 0.}, {12., 6.}}, {-0.820703, -0.820703, -0.820703}}

{2, {{12., 6.}, {0., 6.}, {0., 0.}}, {0, 0, 0}}

{3, {{0., 6.}, {12., 6.}, {0., 12.}}, {0, 0, 0}}

{4, {{12., 12.}, {0., 12.}, {12., 6.}}, {0.820703, 0.820703, 0.820703}}

{5, {{12., 0.}, {24., 0.}, {24., 6.}}, {-0.833049, -0.833049, -0.833049}}

{6, {{24., 6.}, {12., 6.}, {12., 0.}}, {-0.820703, -0.820703, -0.820703}}

{7, {{12., 6.}, {24., 6.}, {12., 12.}}, {0.820703, 0.820703, 0.820703}}

{8, {{24., 12.}, {12., 12.}, {24., 6.}}, {0.833049, 0.833049, 0.833049}}

{9, {{24., 0.}, {36., 0.}, {36., 6.}}, {-0.827204, -0.827204, -0.827204}}

{10, {{36., 6.}, {24., 6.}, {24., 0.}}, {-0.833049, -0.833049, -0.833049}}

{11, {{24., 6.}, {36., 6.}, {24., 12.}}, {0.833049, 0.833049, 0.833049}}

{12, {{36., 12.}, {24., 12.}, {36., 6.}}, {0.827204, 0.827204, 0.827204}}

{13, {{36., 0.}, {48., 0.}, {48., 6.}}, {-0.429107, -0.429107, -0.429107}}

{14, {{48., 6.}, {36., 6.}, {36., 0.}}, {-0.827204, -0.827204, -0.827204}}

{15, {{36., 6.}, {48., 6.}, {36., 12.}}, {0.827204, 0.827204, 0.827204}}

{16, {{48., 12.}, {36., 12.}, {48., 6.}}, {0.429107, 0.429107, 0.429107}}
```



```
{1, {{0., 0.}, {12., 0.}, {12., 6.}}, {-8.28204, -8.28204, -8.28204}}

{2, {{12., 6.}, {0., 6.}, {0., 0.}}, {14.9487, 14.9487, 14.9487}}

{3, {{0., 6.}, {12., 6.}, {0., 12.}}, {14.9487, 14.9487, 14.9487}}

{4, {{12., 12.}, {0., 12.}, {12., 6.}}, {-8.28204, -8.28204, -8.28204}}

{5, {{12., 0.}, {24., 0.}, {24., 6.}}, {-4.99923, -4.99923, -4.99923}}

{6, {{24., 6.}, {12., 6.}, {12., 0.}}, {11.6659, 11.6659, 11.6659}}

{7, {{12., 6.}, {24., 6.}, {12., 12.}}, {11.6659, 11.6659, 11.6659}}

{8, {{24., 12.}, {12., 12.}, {24., 6.}}, {-4.99923, -4.99923, -4.99923}}

{9, {{24., 0.}, {36., 0.}, {36., 6.}}, {-1.66703, -1.66703, -1.66703}}

{10, {{36., 6.}, {24., 6.}, {24., 0.}}, {8.3337, 8.3337, 8.3337}}

{11, {{24., 6.}, {36., 6.}, {24., 12.}}, {8.3337, 8.3337, 8.3337}}

{12, {{36., 12.}, {24., 12.}, {36., 6.}}, {-1.66703, -1.66703, -1.66703}}

{13, {{36., 0.}, {48., 0.}, {48., 6.}}, {1.64179, 1.64179, 1.64179}}

{14, {{48., 6.}, {36., 6.}, {36., 0.}}, {5.02488, 5.02488, 5.02488}}

{15, {{36., 6.}, {48., 6.}, {36., 12.}}, {5.02488, 5.02488, 5.02488}}

{16, {{48., 12.}, {36., 12.}, {48., 6.}}, {1.64179, 1.64179, 1.64179}}
```




```
sxx = syy = sxy = Table[0, {numele}];
```

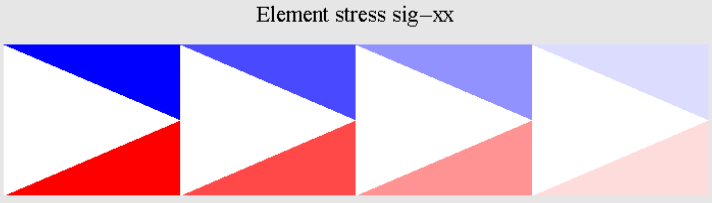
```
Do[{sxx[[ne]], syy[[ne]], sxy[[ne]]} = sig[[ne]], {ne, 1, numele}];
```

```
sxxmax = syymax = sxymax = 0;
```

```
Do[sxxmax = Max[Abs[sxx[[ne]]], sxxmax]; syymax = Max[Abs[syy[[ne]]], syymax];
  sxymax = Max[Abs[sxy[[ne]]], sxymax], {ne, 1, numele}];
```

```
ContourPlotElemFuncOver2DMesh[NodeCoordinates,
 ElemNodeLists, sxx, sxxmax, 1, aspect, "Element stress sig-xx"]
```

```
{1, {{0., 0.}, {12., 0.}, {12., 6.}}, {23.4359, 23.4359, 23.4359}}

{2, {{12., 6.}, {0., 6.}, {0., 0.}}, {0, 0, 0}}

{3, {{0., 6.}, {12., 6.}, {0., 12.}}, {0, 0, 0}}

{4, {{12., 12.}, {0., 12.}, {12., 6.}}, {-23.4359, -23.4359, -23.4359}}

{5, {{12., 0.}, {24., 0.}, {24., 6.}}, {16.6682, 16.6682, 16.6682}}

{6, {{24., 6.}, {12., 6.}, {12., 0.}}, {0, 0, 0}}

{7, {{12., 6.}, {24., 6.}, {12., 12.}}, {0, 0, 0}}

{8, {{24., 12.}, {12., 12.}, {24., 6.}}, {-16.6682, -16.6682, -16.6682}}

{9, {{24., 0.}, {36., 0.}, {36., 6.}}, {9.99927, 9.99927, 9.99927}}

{10, {{36., 6.}, {24., 6.}, {24., 0.}}, {0, 0, 0}}

{11, {{24., 6.}, {36., 6.}, {24., 12.}}, {0, 0, 0}}

{12, {{36., 12.}, {24., 12.}, {36., 6.}}, {-9.99927, -9.99927, -9.99927}}

{13, {{36., 0.}, {48., 0.}, {48., 6.}}, {3.28357, 3.28357, 3.28357}}

{14, {{48., 6.}, {36., 6.}, {36., 0.}}, {0, 0, 0}}

{15, {{36., 6.}, {48., 6.}, {36., 12.}}, {0, 0, 0}}

{16, {{48., 12.}, {36., 12.}, {48., 6.}}, {-3.28357, -3.28357, -3.28357}}
```




```
ContourPlotElemFuncOver2DMesh[NodeCoordinates,
 ElemNodeLists, syy, syymax, 1, aspect, "Element stress sig-yy"]
```

```
{1, {{0., 0.}, {12., 0.}, {12., 6.}}, {-0.820703, -0.820703, -0.820703}}

{2, {{12., 6.}, {0., 6.}, {0., 0.}}, {0, 0, 0}}

{3, {{0., 6.}, {12., 6.}, {0., 12.}}, {0, 0, 0}}

{4, {{12., 12.}, {0., 12.}, {12., 6.}}, {0.820703, 0.820703, 0.820703}}

{5, {{12., 0.}, {24., 0.}, {24., 6.}}, {-0.833049, -0.833049, -0.833049}}

{6, {{24., 6.}, {12., 6.}, {12., 0.}}, {-0.820703, -0.820703, -0.820703}}

{7, {{12., 6.}, {24., 6.}, {12., 12.}}, {0.820703, 0.820703, 0.820703}}

{8, {{24., 12.}, {12., 12.}, {24., 6.}}, {0.833049, 0.833049, 0.833049}}

{9, {{24., 0.}, {36., 0.}, {36., 6.}}, {-0.827204, -0.827204, -0.827204}}

{10, {{36., 6.}, {24., 6.}, {24., 0.}}, {-0.833049, -0.833049, -0.833049}}

{11, {{24., 6.}, {36., 6.}, {24., 12.}}, {0.833049, 0.833049, 0.833049}}

{12, {{36., 12.}, {24., 12.}, {36., 6.}}, {0.827204, 0.827204, 0.827204}}

{13, {{36., 0.}, {48., 0.}, {48., 6.}}, {-0.429107, -0.429107, -0.429107}}

{14, {{48., 6.}, {36., 6.}, {36., 0.}}, {-0.827204, -0.827204, -0.827204}}

{15, {{36., 6.}, {48., 6.}, {36., 12.}}, {0.827204, 0.827204, 0.827204}}

{16, {{48., 12.}, {36., 12.}, {48., 6.}}, {0.429107, 0.429107, 0.429107}}
```