ภาคผนวก

ภาคผนวก ก

โปรแกรมที่ใช้ในวิทยานิพนธ์

```
< Graphics`ParametricPlot3D`;
```

"____

PART 1: INTRODUCTION

____";

"   This
program is mainly applied to analyse both stress and strain of material which are obtained from the Airy stress function. The function must be satisfies equations of equilibrium, strain compatibility equations, stress-strain relations, and boundary conditions. The program is restrictly used to solve only two-dimensional problems on rectangular and polar coordinates. The physical properties of the analysed materials must be homogeneous, isotropic, and linear elastic. The stress obtained from the program had no effected of body force and thermal stress.

The program consists of :

part 1 : Introduction
part 2 : Input type of coordinate system
part 3 : Input variables for rectangular coordinates
part 4 : Input variables for polar coordinates
part 5 : Program

Applications of the program are as follow :
1. Input data on the right hand side of the *under line variables in between the = and ; mark.*
2. Select type of coordinate system between rectangular and polar coordinates from part 2 of the program,
2.1 If rectangular coordinates are chosen, input the data in the third part of the program, then press Enter bottom.
2.2 If polar coordinates are chosen, input the data in the fourth part of the program, then press Enter bottom.
3. No need to do anything with the fifth part of the program which consists of all program details.

All data input to part 3 and part 4 of the program are stress function, type of problem, constant values, boundary region, shape of material, and considered points or sections.

Results obtained from the application of the program are :
1. Calculated results of the components of stress and components of strain equations, components of stress and components of strain at the considered points, maximum values and minimum values of components of stress and components of strain, stress equations at the boundary and the considered sections and maximum values and minimum values of normal stress and shear stress at the considered sections.

```
2. Graphic results of stress distribution, strain distribution,
stress at the boundarys and stress at the considered sections.";

"____________________________________________________________

PART 2: INPUT TYPE OF COORDINATE SYSTEM
____________________________________________________________";

type@of@coordinates = 2;

"note:
      type@of@coordinates = 1; for rectangular coordinates
      type@of@coordinates = 2; for polar coordinates";

"____________________________________________________________

PART 3; INPUT VARIABLES FOR RECTANGULAR COORDINATES
____________________________________________________________";

"INPUT DATA";

degpolynomial = {2, 3, 5};
const@poly[1] = {-25, 0, 0};
const@poly[2] = {0, -37.5, 0};
const@poly[3] = {0, 0, 0};
const@poly[4] = {0, -1245, 12.5};
const@four[1] = 0;
const@four[2] = 0;
const@four[3] = 0;
const@four[4] = 0;
const@four[5] = 0;
const@four[6] = 0;
const@four[7] = 0;
const@four[8] = 0;
minimum@summation@rec = 1;
maximum@summation@rec = 10;
type@of@problem@rec = 2;
young@modu@rec = 200;
shear@modu@rec = 70;
poisson@ratio@rec = 0.3;
minimum@x = -10;
maximum@x = 10;
```

minimum@y = -1;
maximum@y = 1;

"REQUIRED DATA";

coordinatesrec = {{5, 0}, {5, 1}};
csx = {5};
csy = {0, 0.5};

"If you are new user, see below.

Meaning of variables.

1. degpolynomial:
   Polynomial degree in equation of stress function etc. $\Phi = 2x^3$, degpolynomial = {3};. If there are not polynomial term in equation of stress function, we must set variable degpolynomial = {};.
2. const@poly[1] - const@poly[4]:
   Coefficient in equation of stress function (in the form of polynomial) from this equation.

$$\Phi = \sum_{n=2}^{\infty}($$

```
const@poly[1] x^n +
const@poly[2] x^(n-1) y +
const@poly[3] x^(n-2) y^2 +
const@poly[4] x^(n-3) y^3 +
const@poly[5] x^(n-4) y^4 +
....................... +
const@poly[n] x y^(n-1) +
const@poly[n+1] y^n) kN
```

We must set input variables only const@poly[1] - const@poly[4] in each degree of polynomial, because the other constants will be calculated automatically from biharmonic equation from the program. If there are many degree of polynomial in stress function, we can set input variables degpolynomial and const@poly[1] - const@poly[4] in the form of list etc. $\Phi = 2x^2 + x^2y + 3y^3 + 5x^2y^3 - 0.5y^5$, this stress function compose with three degree of polynomial that are 2, 3 and 5. Equation of polynomial degree 2 is const@poly[1]x^2 + const@poly[2]xy + const@poly[3]y^2, from equation of stress function, const@poly[1] = 2;. Equation of polynomial degree 3 is const@poly[1]x^3 + const@poly[2]x^2y + const@poly[3]xy^2 + const@poly[4]y^3, from equation of stress function, const@poly[2] = 1; and const@poly[4] = 3;. Equation of polynomial degree 5 is const@poly[1] x^5 + const@poly[2]x^4y + const@poly[3]x^3y^2 + const@poly[4]x^2y^3 + const@ poly[5]xy^4 + const@poly[6]y^5, from equation of stress function, const@ poly[4] = 5; and const@poly[6] = -0.5;, in this case we must set input variables only const@poly[4]. The method for set input variables in the form of list, we will input all degree of polynomial in the samme rows in between {} mark by fill comma (,) mark between each data and set columes of variables const@poly[1] - const@poly[4] in the same line of each degree of polynomial are as follow.

```
degpolynomial = {2,3,5};
const@poly[1] = {2,0,0};
const@poly[2] = {0,1,0};
const@poly[3] = {0,0,0};
const@poly[4] = {0,3,5};
```

3. const@four[1] - const@four[8]:
   Coefficient in equation of stress function (in the form of fourier series) from this equation.

$$\Phi = \sum_{n=1}^{\infty}($$

```
(const@four[1] Cosh[αy] +
 const@four[2] Sinh[αy] +
 const@four[3] αy Cosh[αy] +
 const@four[4] αy Sinh[αy]) Cos[αx] +
(const@four[5] Cosh[αy] +
 const@four[6] Sinh[αy] +
 const@four[7] αy Cosh[αy] +
 const@four[8] αy Sinh[αy]) Sin[αx]) kN
```

Variable const@four[1] - const@four[8] can be expression in the function of n, while n is a number of summation etc. const@four[1] = 1/n (α = nπ/length of material in x direction).

4. minimum@summation@rec and maximum@summation@rec:
   Beginning and ending of summation term in equation of stress function (in the form of fourier series).
5. type@of@problem@rec:
   Type of problem = 1 for plane strain problem or type of problem = 2 for plane stress problem.
6. young@modu@rec:
   Young's Modulus of Material (GPa).
7. shear@modu@rec:
   Shear's Modulus of Material (GPa).
8. poisson@ratio@rec:
   Poisson's Ratio of Material.
9. minimum@x and maximum@x:
   Boundary region of material in x (m) direction (ratio of material in x and y directions should be greater than 10 for acceptable results).
10. minimum@y and maximum@y:
    Boundary region of material in y (m) direction (ratio of material in x and y directions should be greater than 10 for acceptable results).
11. coordinatesrec:
    Value of coordinates (x,y) at the consider points. If we want to calculate components of stress and components of strain at the considered points, we can set required variables in the form of list etc. The consider points has coordinates = (0,0), coordinatesrec = {{0,0}}; or

The consider two points has coordinates = (0,0) and (1,1), coordinatesrec = {{0,0},{1,1}};. If we do not want to calculate components of stress and components of strain at the considered points, we must set required variable coordinatesrec = {};.

12. csx:
Considered sections normal to x (m) directions. If we want to see graphics of stress at the considered sections, we can set required variables csx in the form of list etc. We want to find stress at the section x = 5m, csx = {5};. or want to find stress at the two sections x = 5m and x = 10m, csx = {5,10};. If we do not want to see graphics of stress at the considered sections, we must set variables csx = {};.

13. csy:
Considered sections normal to y (m) directions. If we want to see graphics of stress at the considered sections, we can set required variables csy in the form of list etc. We want to find stress at the section y = 5m, csy = {5};. or want to find stress at the two sections y = 5m and y = 10m, csy = {5,10};. If we do not want to see graphics of stress at the considered sections, we must set variables csy = {};.";

"____

PART 4; INPUT VARIABLES FOR POLAR COORDINATES
____";

```
"INPUT DATA";

const@polar[1] = 400 / 3;
const@polar[2] = -400 / 6;
const@polar[3] = 0;
const@polar[4] = 0;
const@polar[5] = 0;
const@polar[6] = 0;
const@polar[7] = 0;
const@polar[8] = 0;
const@polar[9] = 0;
const@polar[10] = 0;
const@polar[11] = 0;
const@polar[12] = 0;
const@polar[13] = 0;
const@polar[14] = 0;
const@polar[15] = 0;
const@polar[16] = 0;
const@polar[17] = 0;
const@polar[18] = 0;
const@polar[19] = 0;
```

```
const@polar[20] = 0;
const@polar[21] = 0;
minimum@summation@pol = 2;
maximum@summation@pol = 10;
type@of@problem@pol = 2;
young@modu@pol = 200;
shear@modu@pol = 70;
poisson@ratio@pol = 0.3;
minimum@r = 1;
maximum@r = 2;
minimum@θ = 0;
maximum@θ = 2 Pi;


"REQUIRED DATA";

coordinatespol = {{1.5, 0}, {1.5, Pi / 2}};
csr = {1.5};
csθ = {Pi / 2, Pi};


"If you are new user, see below.


Meaning of variables.

1. const@polar[1] - const@polar[21]:
   Coefficient in equation of stress
  function (in the form of fourier series) from this equation.

   Φ  =  const@polar[1] Log[r] +
         const@polar[2] r^2 +
         const@polar[3] r^2 Log[r] +
         const@polar[4] r^2θ +
         const@polar[5] θ +
         const@polar[6] r θ Cos[θ] +
        (const@polar[7] r^3 +
         const@polar[8]/r +
         const@polar[9] r Log[r]) Cos[θ] +
         const@polar[10] r θ Sin[θ] +
        (const@polar[11] r^3 +
         const@polar[12]/r +
         const@polar[13] r Log[r]) Sin[θ] +

         ∞
         Σ (
        n=2
        (const@polar[14] r^n +
         const@polar[15] r^(n+2) +
         const@polar[16] r^(-n) +
         const@polar[17] r^(-n+2)) Cos[nθ] +
        (const@polar[18] r^n +
```

const@polar[19] r^(n+2) +
const@polar[20] r^(-n) +
const@polar[21] r^(-n+2)) Sin[nθ]) kN

Variable
const@polar[14] - const@polar[21] can be expression in the function of n, while n is a number of summation etc. const@polar[14] = 1/n.

2. minimum@summation@pol and maximum@summation@pol:
Beginning and ending of summation term in equation of stress function (term const@polar[14] - const@polar[21]).
3. type@of@problem@pol:
Type of problem = 1 for plane strain problem or type of problem = 2 for plane stress problem.
4. young@modu@pol:
Young's Modulus of Material (GPa).
5. shear@modu@pol:
Shear's Modulus of Material (GPa).
6. poisson@ratio@pol:
Poisson's Ratio of Material.
7. minimum@r and maximum@r:
Boundary region of material in r(m) direction, variables r base on r = $\sqrt{(x^2+y^2)}$.
8. minimum@θ and maximum@θ:
Boundary region of material in θ(rad) direction variables θ base on θ = arctan[(y/x)].
For circular disk boundary, range of θ are
variable minimum@θ = 0;
variable maximum@θ = 2Pi;.
For straight boundary, range of θ are
variable minimum@θ = -Pi/2;
variable maximum@θ = Pi/2;.
For sector of circulardisk boundary, range of variable minimum@θ and variable maximum@θ are between 0 to 2Pi.
9. coordinatespol:
Value of coordinates (r,θ) at the consider points. If we want to calculate components of stress and components of strain at the considered points, we can set required variables in the form of list etc. The consider points has coordinates = (0,0), coordinatespol = {{0,0}}; or The consider two points has coordinates = (0,0) and (1,1), coordinatespol = {{0,0},{1,1}};. If we do not want to calculate components of stress and components of strain at the considered points, we must set required variable coordinatespol = {};.
10. csr:
Considered sections normal to r (m) directions. If we want to see graphics of stress at the considered sections, we can set required variables csr in the form of list etc. We want to find stress at the section r = 5m, csr = {5};. or want to find stress at the two sections r = 5m and r = 10m, csr = {5,10};. If we do not want to see graphics of stress at the considered sections, we must set variables csr = {};.

11. csθ:

Considered sections normal to θ (rad) directions. If we want to see graphics of stress at the considered sections, we can set required variables csθ in the form of list etc. We want to find stress at the section θ = Pi/2 rad, csθ = {Pi/2};. or want to find stress at the two sections θ = Pi/2 rad and θ = Pi rad, csθ = {pi/2,Pi};. If we do not want to see graphics of stress at the considered sections, we must set variables csθ = {};.";

"

PART 5: PROGRAM

";

```
If[

type@of@coordinates == 1,

stress@func = 0;

x =.; y =.;

   If[

      degpolynomial =!= {},

      a = Array[const, {Max[degpolynomial] + 1, Length[degpolynomial]}];

      a[[1]] = const@poly[1];
      a[[2]] = const@poly[2];
      a[[3]] = const@poly[3];
      a[[4]] = const@poly[4];

      Do[

         d = degpolynomial[[i]];

         If[

            d < 4,

            polynomial@equation =
     Sum[a[[j, i]] * x^(d - j + 1) * y^(j - 1) , {j, 1, d + 1}],

            b = 0; c =.;

            While[
```

```
                d >= 4,

                c[b + 1] = (a[[b + 1, i]] * d!) / (d - 4)! +
      (2 * a[[b + 3, i]] * (d - 2)! * (b + 2)!) / ((d - 4)! * b!) +
      (a[[b + 5, i]] * (b + 4)!) / b!; b = b + 1; d = d - 1

               ];

           d = degpolynomial[[i]];

           polynomial@equation =
     Sum[Flatten[Take[Transpose[a][[i]], d + 1] /. Solve[Array[c, d - 3] == 0,
           Drop[Take[Transpose[a][[i]], d + 1], 4]]][[j]] *
       y^(j - 1) * x^(d - j + 1), {j, 1, d + 1}]

          ];

        stress@func =
    stress@func + polynomial@equation, {i, Length[degpolynomial]}

        ]

     ];

four@equation = Sum[

   (const@four[1] * Cosh[n * Pi * y / maximum@x] +
    const@four[2] * Sinh[n * Pi * y / maximum@x] +
    const@four[3] * n * Pi * y / maximum@x * Cosh[n * Pi * y / maximum@x] +
    const@four[4] * n * Pi * y / maximum@x * Sinh[n * Pi * y / maximum@x]) *
    Cos[n * Pi * x / maximum@x] +

   (const@four[5] * Cosh[n * Pi * y / maximum@x] +
    const@four[6] * Sinh[n * Pi * y / maximum@x] +
    const@four[7] * n * Pi * y / maximum@x * Cosh[n * Pi * y / maximum@x] +
    const@four[8] * n * Pi * y / maximum@x * Sinh[n * Pi * y / maximum@x]) *
    Sin[n * Pi * x / maximum@x],

  {n, minimum@summation@rec, maximum@summation@rec}];

stress@func = stress@func + four@equation;

stress@func = Collect[N[stress@func], {x, y}];

Show[Graphics[Text[
  StyleForm["Program Stress Function",
    FontSize -> 30, FontWeight -> "Bold", FontFamily -> "Times",
    FontColor -> RGBColor[Random[], Random[], Random[]]],
   {0, 0}]], ImageSize -> {650, 100}];
```

```
Print[];
Print[];

Print["Stress Function = ", stress@func, " kN"];

Print[];
Print[];

Print["Equations of Stress"];

Print[];

stress@xx = Collect[D[ stress@func * 10^-3, {y, 2}], {x, y}];

Print["σxx = ", stress@xx, " MPa"];

stress@yy = Collect[D[stress@func * 10^-3 , {x, 2}], {x, y}];

Print["σyy = ", stress@yy, " MPa"];

stress@xy = Collect[-D[stress@func * 10^-3, x, y], {x, y}];

Print["τxy = ", stress@xy, " MPa"];

Print[];
Print[];

   If[

     type@of@problem@rec < 2,

     strain@xx = Collect[Expand[1000 * (1 - poisson@ratio@rec^2) *
     (stress@xx - poisson@ratio@rec / (1 - poisson@ratio@rec) * stress@yy) /
      young@modu@rec],
    {x, y}];
     strain@yy = Collect[Expand[1000 * (1 - poisson@ratio@rec^2) *
     (stress@yy - poisson@ratio@rec / (1 - poisson@ratio@rec) * stress@xx) /
      young@modu@rec],
    {x, y}];
     strain@xy = Collect[Expand[1000 * stress@xy / shear@modu@rec], {x, y}],

     strain@xx = Collect[Expand[1000 *
     (stress@xx - (poisson@ratio@rec * stress@yy)) /young@modu@rec], {x, y}]
     strain@yy = Collect[Expand[1000 *
     (stress@yy - (poisson@ratio@rec * stress@xx)) /young@modu@rec], {x, y}]
     strain@xy = Collect[Expand[1000 * stress@xy / shear@modu@rec], {x, y}]

    ];

Print["Equations of Strain"];
```

```
Print[];

Print["εxx = ", strain@xx, " μ"];
Print["εyy = ", strain@yy, " μ"];
Print["γxy = ", strain@xy, " μ"];

Print[];
Print[];

   If[

      coordinatesrec =!= {},

      innercoordinatesrec = {};

      Do[

         If[

            minimum@x <= coordinatesrec[[i, 1]] <= maximum@x &&
      minimum@y <= coordinatesrec[[i, 2]] <= maximum@y,

            innercoordinatesrec =
      Append[innercoordinatesrec, coordinatesrec[[i]]]

            ],

         {i, 1, Length[coordinatesrec]}

         ];

      coordinatesrec = innercoordinatesrec;

      Do[

         x = coordinatesrec[[i, 1]];
         y = coordinatesrec[[i, 2]];

         Print["Stress Components at (x,y) = ", "(", x, ",", y, ")"];

         Print[];

         Print["σxx = ", stress@xx, " MPa"];
         Print["σyy = ", stress@yy, " MPa"];
         Print["τxy = ", stress@xy, " MPa"];

         Print[];
         Print[];

         Print["Strain Components at (x,y) = ", "(", x, ",", y, ")"];
```

```
        Print[];

        Print["εxx = ", strain@xx, " μ"];
        Print["εyy = ", strain@yy, " μ"];
        Print["γxy = ", strain@xy, " μ"];

        Print[];
        Print[];

        , {i, 1, Length[coordinatesrec]}

       ]

    ];

x =.; y =.;

Print["Boundary Region is x = ", minimum@x, " to ",
  maximum@x, " m and y = ", minimum@y, " to ", maximum@y, " m"];

Print[];
Print[];

setequations =
  {stress@xx, stress@yy, stress@xy, strain@xx, strain@yy, strain@xy};

xxmax = maximum@x;
xxmin = minimum@x;
yymax = maximum@y;
yymin = minimum@y;

pixelnumberxx = 64;
pixelnumberyy = 16;

  Do[

     equation = setequations[[k]];

     x = xxmin; y = yymin;

     maxvalue = equation;
     minvalue = equation;

     xxmaxvalue = x;
     xxminvalue = x;
     yymaxvalue = y;
     yyminvalue = y;

     Do[

        y = yymin + (yymax - yymin) j / pixelnumberyy;
```

```
Do[

    x = xxmin + (xxmax - xxmin) i / pixelnumberxx;

    If[

        equation > maxvalue,
maxvalue = equation; xxmaxvalue = x; yymaxvalue = y

      ];

    If[

        equation < minvalue,
minvalue = equation; xxminvalue = x; yyminvalue = y

      ],

    {i, 0, pixelnumberxx}

   ],

  {j, 0, pixelnumberyy}

 ];

If[

   xxmaxvalue =!= xxmax &&
xxmaxvalue =!= xxmin && yymaxvalue =!= yymax && yymaxvalue =!= yymin,

   xxmaxbox = xxmaxvalue - (xxmax - xxmin) / pixelnumberxx;
   yymaxbox = yymaxvalue - (yymax - yymin) / pixelnumberyy;

   Do[

     y = yymaxbox + (yymax - yymin) 2 j / pixelnumberyy^2;

     Do[

        x = xxmaxbox + (xxmax - xxmin) 2 i / pixelnumberxx^2;

        If[

           equation > maxvalue,

           maxvalue = equation

          ],
```

```
              {i, 0, pixelnumberxx}
            ],
          {j, 0, pixelnumberyy}
        ]
      ];
    If[
        xxminvalue =!= xxmax &&
  xxminvalue =!= xxmin && yyminvalue =!= yymax && yyminvalue =!= yymin,
        xxminbox = xxminvalue - (xxmax - xxmin) / pixelnumberxx;
        yyminbox = yyminvalue - (yymax - yymin) / pixelnumberyy;
        Do[
            y = yyminbox + (yymax - yymin) 2 j / pixelnumberyy^2;
            Do[
                x = xxminbox + (xxmax - xxmin) 2 i / pixelnumberxx^2;
                If[
                    equation < minvalue,
                    minvalue = equation
                  ],
                {i, 0, pixelnumberxx}
              ],
            {j, 0, pixelnumberyy}
          ]
      ];
    mav[k] = N[maxvalue];
    miv[k] = N[minvalue];
    x =.; y =.,
    {k, 1, Length[setequations]}
```

```
    ];

Print["Maximum and Minimum Stress"];

Print[];

Print["σxx max = ", mav[1], " MPa"];
Print["σyy max = ", mav[2], " MPa"];
Print["τxy max = ", mav[3], " MPa"];

Print[];

Print["σxx min = ", miv[1], " MPa"];
Print["σyy min = ", miv[2], " MPa"];
Print["τxy min = ", miv[3], " MPa"];

Print[];
Print[];

Print["Maximum and Minimum Strain"];

Print[];

Print["εxx max = ", mav[4], " μ"];
Print["εyy max = ", mav[5], " μ"];
Print["γxy max = ", mav[6], " μ"];

Print[];

Print["εxx min = ", miv[4], " μ"];
Print["εyy min = ", miv[5], " μ"];
Print["γxy min = ", miv[6], " μ"];

Print[];
Print[];

Show[Graphics[Text[
  StyleForm["Stress Distribution",
     FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
     FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

"Graphics of σxx distribution";

rectangularstress1 =
  Plot3D[stress@xx, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "σ xx (MPa)"},
```

```
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of σyy distribution";

rectangularstress2 =
  Plot3D[stress@yy, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "σ yy (MPa)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
```

```
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of τxy distribution";

rectangularstress3 =
  Plot3D[stress@xy, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "τ xy (MPa)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

Show[GraphicsArray[
   {{rectangularstress1}, {rectangularstress2}, {rectangularstress3}}],
  ImageSize → {650, 700}];

Print[];
Print[];
```

```
Show[Graphics[Text[
  StyleForm["Strain Distribution",
     FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
     FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

"Graphics of εxx distribution";

rectangularstrain1 =
  Plot3D[strain@xx, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "ε xx (μ)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of εyy distribution";

rectangularstrain2 =
  Plot3D[strain@yy, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "ε yy (μ)"},
```

```
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of γxy distribution";

rectangularstrain3 =
  Plot3D[strain@xy, {x, minimum@x, maximum@x}, {y, minimum@y, maximum@y},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.6,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "γ xy (μ)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {3, 1, 1},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorFunction → Automatic,
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  HiddenSurface → True,
  Lighting → True,
  LightSources → {},
  Mesh → True,
  MeshStyle → {GrayLevel[0]},
```

```
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

Show[GraphicsArray[
   {{rectangularstrain1}, {rectangularstrain2}, {rectangularstrain3}}],
  ImageSize → {650, 700}];

Print[];
Print[];

Show[Graphics[Text[
  StyleForm["Boundary Conditions",
     FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
     FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

Print[];
Print[];

x = minimum@x;

Print["Stress at the Boundary x = ", x, " m" ];

Print[];

Print["σxx at x = ", x , " m is ", stress@xx, " MPa"];
Print["τxy at x = ", x , " m is ", stress@xy, " MPa"];

"Graphics of σxx at x = minimum@x";

rectangularboundary1 =
  Plot[If[-10^-6 < stress@xx < 10^-6, 0, stress@xx], {y, minimum@y, maximum@y
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"y (m)", "normal stress (MPa)"},
```

```
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

"Graphics of τxy at x = minimum@x";

rectangularboundary2 =
  Plot[If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {y, minimum@y, maximum@y
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"y (m)", "shear stress (MPa)"},
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

Show[GraphicsArray[
   {rectangularboundary1, rectangularboundary2}], ImageSize → {500, 250}];

x =.;

Print[];
Print[];
```

```
x = maximum@x;

Print["Stress at the Boundary x = ", x , " m"];

Print[];

Print["σxx at x = ", x , " m is ", stress@xx, " MPa"];
Print["τxy at x = ", x , " m is ", stress@xy, " MPa"];

"Graphics of σxx at x = maximum@x";

rectangularboundary3 =
  Plot[If[-10^-6 < stress@xx < 10^-6, 0, stress@xx], {y, minimum@y, maximum@y
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"y (m)", "normal stress (MPa)"},
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

"Graphics of τxy at x = maximum@x";

rectangularboundary4 =
  Plot[If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {y, minimum@y, maximum@y
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
```

```
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"y (m)", "shear stress (MPa)"},
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

Show[GraphicsArray[
   {rectangularboundary3, rectangularboundary4}], ImageSize → {500, 250}];

x =.;

Print[];
Print[];

y = minimum@y;

Print["Stress at the Boundary y = ", y, " m"];

Print[];

Print["σyy at y = ", y, " m is ", stress@yy, " MPa"];
Print["τxy at y = ", y, " m is ", stress@xy, " MPa"];

"Graphics of σyy at y = minimum@y";

rectangularboundary5 =
  Plot[If[-10^-6 < stress@yy < 10^-6, 0, stress@yy], {x, minimum@x, maximum@x
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
```

```
 FrameLabel → {"x (m)", "normal stress (MPa)"},
 FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
 FrameTicks → Automatic,
 GridLines → Automatic,
 MaxBend → 10,
 PlotDivision → 20,
 PlotLabel → None,
 PlotPoints → 15,
 PlotRange → Automatic,
 PlotRegion → {{0, 0.95}, {0, 0.95}},
 PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
 RotateLabel → True,
 TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
 Ticks → Automatic];

"Graphics of τxy at y = minimum@y";

rectangularboundary6 =
  Plot[If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {x, minimum@x, maximum@x
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"x (m)", "shear stress (MPa)"},
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

Show[GraphicsArray[
   {rectangularboundary5, rectangularboundary6}], ImageSize → {500, 250}];

y =.;

Print[];
```

```
Print[];

y = maximum@y;

Print["Stress at the Boundary y = ", y , " m"];

Print[];

Print["σyy at y = ", y , " m is ", stress@yy, " MPa"];
Print["τxy at y = ", y , " m is ", stress@xy, " MPa"];

"Graphics of σyy at y = maximum@y";

rectangularboundary7 =
  Plot[If[-10^-6 < stress@yy < 10^-6, 0, stress@yy], {x, minimmm@x, maximum@x
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FormatType → $FormatType,
  Frame → True,
  FrameLabel → {"x (m)", "normal stress (MPa)"},
  FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
  FrameTicks → Automatic,
  GridLines → Automatic,
  MaxBend → 10,
  PlotDivision → 20,
  PlotLabel → None,
  PlotPoints → 15,
  PlotRange → Automatic,
  PlotRegion → {{0, 0.95}, {0, 0.95}},
  PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
  RotateLabel → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic];

"Graphics of τxy at y = maximum@y";

rectangularboundary8 =
  Plot[If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {x, minimmm@x, maximum@x
  AspectRatio → 0.8,
  Axes → False,
  AxesLabel → None,
  AxesOrigin → Automatic,
  AxesStyle → None,
  Background → RGBColor[0.8, 1, 1],
```

```
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"x (m)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

Show[GraphicsArray[
   {rectangularboundary7, rectangularboundary8}], ImageSize → {500, 250}];

y =.;

csx = Sort[Select[Select[csx, # > minimum@x&], # < maximum@x&], Less];

csy = Sort[Select[Select[csy, # > minimum@y&], # < maximum@y&], Less];

   If[csx =!= {} || csy =!= {},

      Print[];
      Print[];

Show[Graphics[Text[
  StyleForm["Considered Section",
      FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
      FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

      If[

         csx =!= {},

         Do[

            x = csx[[i]];

            Print[];
            Print[];
```

```
Print["Stress at the Section x = ", x, " m"];

Print[];

Print["σxx at x = ", x , " m is ", stress@xx, " MPa"];
Print["τxy at x = ", x , " m is ", stress@xy, " MPa"];

"Graphics of σxx at x = csx[[i]]";

rectangularsection1 = Plot[
If[-10^-6 < stress@xx < 10^-6, 0, stress@xx], {y, minimum@y, maximum@y}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"y (m)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τxy at x = csx[[i]]";

rectangularsection2 = Plot[
If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {y, minimum@y, maximum@y}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, O],
DisplayFunction → Identity,
```

```
FormatType → $FormatType,
Frame → True,
FrameLabel → {"y (m)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

Show[GraphicsArray[
{rectangularsection1, rectangularsection2}], ImageSize → {500, 250}];

setequations = {stress@xx, stress@xy};

yymax = maximum@y;
yymin = minimum@y;

pixelnumberyy = 32;

Do[

    equation = setequations[[k]];

    y = yymin;

    maxvalue = equation;
    minvalue = equation;

    yymaxvalue = y;
    yyminvalue = y;

    Do[

        y = yymin + (yymax - yymin) j / pixelnumberyy;

        If[

            equation > maxvalue,

            maxvalue = equation; yymaxvalue = y

        ];
```

```
    If[
      equation < minvalue,
      minvalue = equation; yyminvalue = y
     ],
    {j, 0, pixelnumberyy}
   ];
  If[
    yymaxvalue =!= yymax && yymaxvalue =!= yymin,
    yymaxbox = yymaxvalue - (yymax - yymin) / pixelnumberyy;
    Do[
      y = yymaxbox + (yymax - yymin) 2 j / pixelnumberyy^2;
      If[
        equation > maxvalue,
        maxvalue = equation
       ],
      {j, 0, pixelnumberyy}
     ]
   ];
  If[
    yyminvalue =!= yymax && yyminvalue =!= yymin,
    yyminbox = yyminvalue - (yymax - yymin) / pixelnumberyy;
    Do[
      y = yyminbox + (yymax - yymin) 2 j / pixelnumberyy^2;
      If[
        equation < minvalue,
```

```
                minvalue = equation

               ],

              {j, 0, pixelnumberyy}

             ]

           ];

          mav[k] = N[maxvalue];
          miv[k] = N[minvalue];

          y =.,

          {k, 1, Length[setequations]}

         ];

       Print["Maximum and Minimum Stress"];

       Print[];

       Print["σxx max = ", mav[1], " MPa"];
       Print["τxy max = ", mav[2], " MPa"];

       Print[];

       Print["σxx min = ", miv[1], " MPa"];
       Print["τxy min = ", miv[2], " MPa"];

       , {i, 1, Length[csx]}

      ];

    x =.

   ];

 If[

    csy =!= {},

    Do[

       y = csy[[i]];

       Print[];
       Print[];

       Print["Stress at the Section y = ", y, " m" ];
```

```
Print[];

Print["σyy at y = ", y, " m is ", stress@yy, " MPa"];
Print["τxy at y = ", y, " m is ", stress@xy, " MPa"];

"Graphics of σyy at y = csy[[i]]";

rectangularsection3 = Plot[
If[-10^-6 < stress@yy < 10^-6, 0, stress@yy], {x, minimum@x, maximum@x}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"x (m)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τxy at y = csy[[i]]";

rectangularsection4 = Plot[
If[-10^-6 < stress@xy < 10^-6, 0, stress@xy], {x, minimum@x, maximum@x}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
```

```
Frame → True,
FrameLabel → {"x (m)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

Show[GraphicsArray[
{rectangularsection3, rectangularsection4}], ImageSize → {500, 250}];

setequations = {stress@yy, stress@xy};

xxmax = maximum@x;
xxmin = minimum@x;

pixelnumberxx = 32;

Do[

  equation = setequations[[k]];

  x = xxmin;

  maxvalue = equation;
  minvalue = equation;

  xxmaxvalue = x;
  xxminvalue = x;

  Do[

    x = xxmin + (xxmax - xxmin) j / pixelnumberxx;

    If[

      equation > maxvalue,

      maxvalue = equation; xxmaxvalue = x

    ];
```

```
    If[
       equation < minvalue,
       minvalue = equation; xxminvalue = x
      ],
    {j, 0, pixelnumberxx}
   ];
 If[
    xxmaxvalue =!= xxmax && xxmaxvalue =!= xxmin,
    xxmaxbox = xxmaxvalue - (xxmax - xxmin) / pixelnumberxx;
    Do[
       x = xxmaxbox + (xxmax - xxmin) 2 j / pixelnumberxx^2;
       If[
          equation > maxvalue,
          maxvalue = equation
         ],
       {j, 0, pixelnumberxx}
      ]
    ];
 If[
    xxminvalue =!= xxmax && xxminvalue =!= xxmin,
    xxminbox = xxminvalue - (xxmax - xxmin) / pixelnumberxx;
    Do[
       x = xxminbox + (xxmax - xxmin) 2 j / pixelnumberxx^2;
       If[
          equation < minvalue,
          minvalue = equation
```

```
                    ],

                  {j, 0, pixelnumberxx}

                 ]

               ];

             mav[k] = N[maxvalue];
             miv[k] = N[minvalue];

             x =.,

             {k, 1, Length[setequations]}

            ];

          Print["Maximum and Minimum Stress"];

          Print[];

          Print["σyy max = ", mav[1], " MPa"];
          Print["τxy max = ", mav[2], " MPa"];

          Print[];

          Print["σyy min = ", miv[1], " MPa"];
          Print["τxy min = ", miv[2], " MPa"];

          , {i, 1, Length[csy]}

         ];

       y =.

      ]

    ],


r =.; theta =.;

stress@func =

   const@polar[1] * Log[r] +
   const@polar[2] * r^2 +
   const@polar[3] * r^2 * Log[r] +
   const@polar[4] * r^2 * theta +
   const@polar[5] * theta +
```

```
   const@polar[6] * r * theta * Cos[theta] +
  (const@polar[7] * r^3 +
   const@polar[8] * r^-1 +
   const@polar[9] * r * Log[r]) * Cos[theta] +
   const@polar[10] * r * theta * Sin[theta] +
  (const@polar[11] * r^3 +
   const@polar[12] * r^-1 +
   const@polar[13] * r * Log[r]) * Sin[theta] +

   Sum[

       (const@polar[14] * r^n +
        const@polar[15] * r^(n + 2) +
        const@polar[16] * r^-n +
        const@polar[17] * r^(-n + 2)) * Cos[n * theta] +

       (const@polar[18] * r^n +
        const@polar[19] * r^(n + 2) +
        const@polar[20] * r^-n +
        const@polar[21] * r^(-n + 2)) * Sin[n * theta],

       {n, minimum@summation@pol, maximum@summation@pol}];

stress@func = Collect[N[stress@func], {r, theta}];

Show[Graphics[Text[
  StyleForm["Program Stress Function",
     FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
     FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

Print[];
Print[];

Print["Stress Function = ", stress@func, " kN"];

Print[];
Print[];

Print["Equations of Stress"];

Print[];

stress@rr = Collect[D[ stress@func * 10^-3, r] / r +
    D[ stress@func * 10^-3, {theta, 2}] / r^2, {r, theta}];

Print["σrr = ", stress@rr, " MPa"];

stress@θθ = Collect[D[ stress@func * 10^-3, {r, 2}], {r, theta}];

Print["σθθ = ", stress@θθ, " MPa"];
```

```
stress@rθ = Collect[-D[D[stress@func * 10^-3, theta] / r, r], {r, theta}];

Print["σrθ = ", stress@rθ, " MPa"];

Print[];
Print[];

   If[

      type@of@problem@pol < 2,

      strain@rr = Collect[Expand[1000 * (1 - poisson@ratio@pol^2) *
      (stress@rr - poisson@ratio@pol / (1 - poisson@ratio@pol) * stress@θθ) /
       young@modu@pol],
    {r, theta}];
      strain@θθ = Collect[Expand[1000 * (1 - poisson@ratio@pol^2) *
      (stress@θθ - poisson@ratio@pol / (1 - poisson@ratio@pol) * stress@rr) /
       young@modu@pol],
    {r, theta}];
      strain@rθ =
   Collect[Expand[1000 * stress@rθ / shear@modu@pol], {r, theta}],

      strain@rr = Collect[Expand[
     1000 * (stress@rr - (poisson@ratio@pol * stress@θθ)) / young@modu@pol],
    {r, theta}];
      strain@θθ = Collect[Expand[
     1000 * (stress@θθ - (poisson@ratio@pol * stress@rr)) / young@modu@pol],
    {r, theta}];
      strain@rθ =
   Collect[Expand[1000 * stress@rθ / shear@modu@pol], {r, theta}]

   ];

Print["Equations of Strain"];

Print[];

Print["εrr = ", strain@rr, " μ"];
Print["εθθ = ", strain@θθ, " μ"];
Print["γrθ = ", strain@rθ, " μ"];

Print[];
Print[];

   If[

      coordinatespol =!= {},

      innercoordinatespol = {};
```

```
Do[

   If[

      minimum@r <= coordinatespol[[i, 1]] <= maximum@r &&
minimum@θ <= coordinatespol[[i, 2]] <= maximum@θ,

      innercoordinatespol =
Append[innercoordinatespol, coordinatespol[[i]]]

     ],

   {i, 1, Length[coordinatespol]}

  ];

coordinatespol = innercoordinatespol;

Do[

   r = coordinatespol[[i, 1]];
   theta = coordinatespol[[i, 2]];

   Print["Stress Components at (r,θ) = ", "(", r, ",", theta, ")"];

   Print[];

   Print["σrr = ", stress@rr, " MPa"];
   Print["σθθ = ", stress@θθ, " MPa"];
   Print["τrθ = ", stress@rθ, " MPa"];

   Print[];
   Print[];

   Print["Strain Components at (r,θ) = ", "(", r, ",", theta, ")"];

   Print[];

   Print["εrr = ", strain@rr, " μ"];
   Print["εθθ = ", strain@θθ, " μ"];
   Print["γrθ = ", strain@rθ, " μ"];

   Print[];
   Print[];

   , {i, 1, Length[coordinatespol]}

  ]

];
```

```
r =.; theta =.;

Print["Boundary Region is r = ", minimum@r, " to ",
  maximum@r, " m and θ = ", minimum@θ, " to ", maximum@θ, " rad"];

Print[];
Print[];

setequations =
  {stress@rr, stress@θθ, stress@rθ, strain@rr, strain@θθ, strain@rθ};

rrmax = maximum@r;
rrmin = minimum@r;
θθmax = maximum@θ;
θθmin = minimum@θ;

pixelnumberrr = 120;
pixelnumberθθ = 8;

   Do[

      equation = setequations[[k]];

      r = rrmin; theta = θθmin;

      maxvalue = equation;
      minvalue = equation;

      rrmaxvalue = r;
      rrminvalue = r;
      θθmaxvalue = theta;
      θθminvalue = theta;

      Do[

         theta = θθmin + (θθmax - θθmin) j / pixelnumberθθ;

         Do[

            r = rrmin + (rrmax - rrmin) i / pixelnumberrr;

            If[

               equation > maxvalue,

               maxvalue = equation; rrmaxvalue = r; θθmaxvalue = theta

              ];

            If[
```

```
          equation < minvalue,

          minvalue = equation; rrminvalue = r; θθminvalue = theta

         ],

        {i, 0, pixelnumberrr}

       ],

      {j, 0, pixelnumberθθ}

     ];

    If[

      rrmaxvalue =!= rrmax &&
rrmaxvalue =!= rrmin && θθmaxvalue =!= θθmax && θθmaxvalue =!= θθmin,

      rrmaxbox = rrmaxvalue - (rrmax - rrmin) / pixelnumberrr;
      θθmaxbox = θθmaxvalue - (θθmax - θθmin) / pixelnumberθθ;

      Do[

        theta = θθmaxbox + (θθmax - θθmin) 2 j / pixelnumberθθ^2;

        Do[

          r = rrmaxbox + (rrmax - rrmin) 2 i / pixelnumberrr^2;

          If[

            equation > maxvalue,

            maxvalue = equation

           ],

          {i, 0, pixelnumberrr}

         ],

        {j, 0, pixelnumberθθ}

       ]

     ];

    If[

      rrminvalue =!= rrmax &&
```

```
rrminvalue =!= rrmin && yyminvalue =!= θθmax && yyminvalue =!= θθmin,

    rrminbox = rrminvalue - (rrmax - rrmin) / pixelnumberrr;
    θθminbox = θθminvalue - (θθmax - θθmin) / pixelnumberθθ;

    Do[

      theta = θθminbox + (θθmax - θθmin) 2 j / pixelnumberθθ^2;

      Do[

        r = rrminbox + (rrmax - rrmin) 2 i / pixelnumberrr^2;

        If[

          equation < minvalue,

          minvalue = equation

         ],

        {i, 0, pixelnumberrr}

       ],

      {j, 0, pixelnumberθθ}

     ]

    ];

   mav[k] = N[maxvalue];
   miv[k] = N[minvalue];

   r =.; theta =.,

   {k, 1, Length[setequations]}

  ];

Print["Maximum and Minimum Stress"];

Print[];

Print["σrr max = ", mav[1], " MPa"];
Print["σθθ max = ", mav[2], " MPa"];
Print["τrθ max = ", mav[3], " MPa"];

Print[];

Print["σrr min = ", miv[1], " MPa"];
```

```
Print["σθθ min = ", miv[2], " MPa"];
Print["τrθ min = ", miv[3], " MPa"];

Print[];
Print[];

Print["Maximum and Minimum Strain"];

Print[];

Print["εrr max = ", mav[4], " μ"];
Print["εθθ max = ", mav[5], " μ"];
Print["γrθ max = ", mav[6], " μ"];

Print[];

Print["εrr min = ", miv[4], " μ"];
Print["εθθ min = ", miv[5], " μ"];
Print["γrθ min = ", miv[6], " μ"];

Print[];
Print[];

Show[Graphics[Text[
  StyleForm["Stress Distribution",
     FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
     FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

"Graphics of σrr distribution";

polarstress1 = CylindricalPlot3D[ stress@rr,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "σ rr (MPa)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
```

```
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of σθθ distribution";

polarstress2 = CylindricalPlot3D[ stress@θθ,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "σ θθ (MPa)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of τrθ distribution";

polarstress3 = CylindricalPlot3D[ stress@rθ,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "τ rθ (MPa)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
```

```
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

Show[GraphicsArray[{{polarstress1}, {polarstress2}, {polarstress3}}],
  ImageSize → {650, 700}];

Print[];
Print[];

Show[Graphics[Text[
 StyleForm["Strain Distribution",
    FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
    FontColor → RGBColor[Random[], Random[], Random[]]],
   {0, 0}]], ImageSize → {650, 100}];

"Graphics of εrr distribution";

polarstrain1 = CylindricalPlot3D[ strain@rr,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "ε rr (μ)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
```

```
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of εθθ distribution";

polarstrain2 = CylindricalPlot3D[ strain@θθ,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
  AxesLabel → {"x (m)", "y (m)", "ε θθ (μ)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

"Graphics of γrθ distribution";

polarstrain3 = CylindricalPlot3D[ strain@rθ,
      {r, minimum@r, maximum@r}, {theta, minimum@θ, maximum@θ},
  AmbientLight → RGBColor[1, 1, 0],
  AspectRatio → 0.7,
  Axes → True,
  AxesEdge → {{-1, -1}, {1, -1}, {-1, -1}},
```

```
  AxesLabel → {"x (m)", "y (m)", "γ rθ (μ)"},
  AxesStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  Background → RGBColor[0.8, 1, 1],
  Boxed → True,
  BoxRatios → {1, 1, 0.7},
  BoxStyle → {AbsoluteThickness[1], RGBColor[0, 1, 0]},
  ColorOutput → Automatic,
  DefaultColor → RGBColor[0, 0, 0],
  DisplayFunction → Identity,
  FaceGrids → None,
  FormatType → $FormatType,
  Lighting → True,
  LightSources → {},
  PlotLabel → None,
  PlotPoints → 20,
  PlotRange → Automatic,
  PlotRegion → {{0.05, 0.95}, {0, 1}},
  Shading → True,
  TextStyle → {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
  Ticks → Automatic,
  ViewCenter → Automatic,
  ViewPoint → {1.3, -2.4, 2}];

Show[GraphicsArray[{{polarstrain1}, {polarstrain2}, {polarstrain3}}],
  ImageSize → {650, 700}];

Print[];
Print[];

Show[Graphics[Text[
 StyleForm["Boundary Conditions",
    FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
    FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
   {0, 0}]], ImageSize → {650, 100}];

Print[];
Print[];

   Which[

         maximum@θ - minimum@θ == 2 Pi,

         r = minimum@r;

         Print["Stress at the Boundary r = ", r, " m"];

         Print[];

         Print["σrr at r = ", r , " m is ", stress@rr, " MPa"];
         Print["τrθ at r = ", r , " m is ", stress@rθ, " MPa"];
```

```
"Graphics of σrr at r = minimum@r";

circulardiskboundary1 = Plot[
If[-10^-6 < stress@rr < 10^-6, 0, stress@rr], {theta, minimum@θ, maximum@
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"θ (rad)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic, None, None},
GridLines → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic},
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τrθ at r = minimum@r";

circulardiskboundary2 = Plot[
If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {theta, minimum@θ, maximum@
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"θ (rad)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic, None, None},
GridLines → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic},
MaxBend → 10,
```

```
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
 {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    Show[GraphicsArray[
{circulardiskboundary1, circulardiskboundary2}], ImageSize → {500, 250}];

    r =.;

    Print[];
    Print[];

    r = maximum@r;

    Print["Stress at the Boundary r = ", r , " m"];

    Print[];

    Print["σrr at r = ", r , " m is ", stress@rr, " MPa"];
    Print["τrθ at r = ", r , " m is ", stress@rθ, " MPa"];

    "Graphics of σrr at r = maximum@r";

    circulardiskboundary3 = Plot[
If[-10^-6 < stress@rr < 10^-6, 0, stress@rr], {theta, minimum@θ, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"θ (rad)", "normal stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic, None, None},
    GridLines → {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
```

```
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
 {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    "Graphics of τrθ at r = maximum@r";

    circulardiskboundary4 = Plot[
If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {theta, minimum@θ, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"θ (rad)", "shear stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic, None, None},
    GridLines → {{0, Pi/2, Pi, 3Pi/2, 2Pi}, Automatic},
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
 {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    Show[GraphicsArray[
{circulardiskboundary3, circulardiskboundary4}], ImageSize → {500, 250}];

    r =.;,

    maximum@θ - minimum@θ == Pi,

    theta = maximum@θ;
    stressθθ1 = stress@θθ;
    stressrθ1 = stress@rθ;

    theta = minimum@θ;
```

```
stressθθ2 = stress@θθ /. r → -r;
stressrθ2 = stress@rθ /. r → -r;

maxr = maximum@r;
minr = minimum@r;

Print["Stress at the Straight Boundary"];

Print[];

Print["σθθ at y > 0 is ", stressθθ1, " MPa"];
Print["σθθ at y < 0 is ", stressθθ2, " MPa"];
Print["τrθ at y > 0 is ", stressrθ1, " MPa"];
Print["τrθ at y < 0 is ", stressrθ2, " MPa"];

"Graphics of σθθ at halfplane boundary";

halfplaneboundary1 = Plot[Which[r < -minr,
If[-10^-6 < stressθθ2 < 10^-6, 0, stressθθ2], -minr ≤ r ≤ minr, 0, minr < r,
If[-10^-6 < stressθθ1 < 10^-6, 0, stressθθ1]], {r, -maximum@r, maximum@r}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"y (m)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τrθ at halfplane boundary";

halfplaneboundary2 = Plot[Which[r < -minr,
If[-10^-6 < stressrθ2 < 10^-6, 0, stressrθ2], -minr ≤ r ≤ minr, 0, minr < r,
```

```
If[-10^-6 < stressrθ1 < 10^-6, 0, stressrθ1]], {r, -maximum@r, maximum@r}
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"y (m)", "shear stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → Automatic,
    GridLines → Automatic,
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
 {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    Show[GraphicsArray[
{halfplaneboundary1, halfplaneboundary2}], ImageSize → {500, 250}];

    theta =.;,

    maximum@θ - minimum@θ > 0,

    r = minimum@r;

    Print["Stress at the Boundary r = ", r, " m"];

    Print[];

    Print["σrr at r = ", r, " m is ", stress@rr, " MPa"];
    Print["τrθ at r = ", r, " m is ", stress@rθ, " MPa"];

    "Graphics of σrr at r = minimum@r";

    sectorboundary1 = Plot[
If[-10^-6 < stress@rr < 10^-6, 0, stress@rr], {theta, minimum@θ, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
```

```
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"θ (rad)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τrθ at r = minimum@r";

sectorboundary2 = Plot[
If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {theta, minimum@θ, maximum@
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"θ (rad)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
```

```
    TextStyle →
  {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    Show[GraphicsArray[
{sectorboundary1, sectorboundary2}], ImageSize → {500, 250}];

    r =.;

    Print[];
    Print[];

    r = maximum@r;

    Print["Stress at the Boundary r = ", r, " m"];

    Print[];

    Print["σrr at r = ", r, " m is ", stress@rr, " MPa"];
    Print["τrθ at r = ", r, " m is ", stress@rθ, " MPa"];

    "Graphics of σrr at r = maximum@r";

    sectorboundary3 = Plot[
If[-10^-6 < stress@rr < 10^-6, 0, stress@rr], {theta, minimum@θ, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"θ (rad)", "normal stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → Automatic,
    GridLines → Automatic,
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
  {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];
```

```
"Graphics of τrθ at r = maximum@r";

sectorboundary4 = Plot[
If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {theta, minimum@θ, maximum@
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"θ (rad)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

Show[GraphicsArray[
{sectorboundary3, sectorboundary4}], ImageSize → {500, 250}];

r =.;

Print[];
Print[];

theta = minimum@θ;

Print["Stress at the Boundary θ = ", theta, " rad" ];

Print[];

Print["σθθ at θ = ", theta , " rad is ", stress@θθ, " MPa"];
Print["τrθ at θ = ", theta , " rad is ", stress@rθ, " MPa"];

"Graphics of σθθ at theta = minimum@θ";
```

```
    sectorboundary5 =
Plot[If[-10^-6 < stress@θθ < 10^-6, 0, stress@θθ], {r, minimum@r, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"r (m)", "normal stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → Automatic,
    GridLines → Automatic,
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
  {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    "Graphics of τrθ at theta = minimum@θ";

    sectorboundary6 =
Plot[If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {r, minimum@r, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"r (m)", "shear stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → Automatic,
    GridLines → Automatic,
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
```

```
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
    PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
    RotateLabel → True,
    TextStyle →
 {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
    Ticks → Automatic];

    Show[GraphicsArray[
 {sectorboundary5, sectorboundary6}], ImageSize → {500, 250}];

    theta =.;

    Print[];
    Print[];

    theta = maximum@θ;

    Print["Stress at the Boundary θ = ", theta, " rad" ];

    Print[];

    Print["σθθ at θ = ", theta , " rad is ", stress@θθ, " MPa"];
    Print["τrθ at θ = ", theta , " rad is ", stress@rθ, " MPa"];

    "Graphics of σθθ at theta = maximum@θ";

    sectorboundary7 =
Plot[If[-10^-6 < stress@θθ < 10^-6, 0, stress@θθ], {r, minimum@r, maximum@
    AspectRatio → 0.8,
    Axes → False,
    AxesLabel → None,
    AxesOrigin → Automatic,
    AxesStyle → None,
    Background → RGBColor[0.8, 1, 1],
    ColorOutput → Automatic,
    DefaultColor → RGBColor[0, 0, 0],
    DisplayFunction → Identity,
    FormatType → $FormatType,
    Frame → True,
    FrameLabel → {"r (m)", "normal stress (MPa)"},
    FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
    FrameTicks → Automatic,
    GridLines → Automatic,
    MaxBend → 10,
    PlotDivision → 20,
    PlotLabel → None,
    PlotPoints → 15,
    PlotRange → Automatic,
    PlotRegion → {{0, 0.95}, {0, 0.95}},
```

```
        PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
        RotateLabel → True,
        TextStyle →
    {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
        Ticks → Automatic];

        "Graphics of τrθ at theta = maximum@θ";

        sectorboundary8 =
  Plot[If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {r, minimum@r, maximum@
        AspectRatio → 0.8,
        Axes → False,
        AxesLabel → None,
        AxesOrigin → Automatic,
        AxesStyle → None,
        Background → RGBColor[0.8, 1, 1],
        ColorOutput → Automatic,
        DefaultColor → RGBColor[0, 0, 0],
        DisplayFunction → Identity,
        FormatType → $FormatType,
        Frame → True,
        FrameLabel → {"r (m)", "shear stress (MPa)"},
        FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
        FrameTicks → Automatic,
        GridLines → Automatic,
        MaxBend → 10,
        PlotDivision → 20,
        PlotLabel → None,
        PlotPoints → 15,
        PlotRange → Automatic,
        PlotRegion → {{0, 0.95}, {0, 0.95}},
        PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
        RotateLabel → True,
        TextStyle →
    {FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
        Ticks → Automatic];

        Show[GraphicsArray[
    {sectorboundary7, sectorboundary8}], ImageSize → {500, 250}];

        theta =.

       ];

csr = Sort[Select[Select[csr, # > minimum@r&], # < maximum@r&], Less];

csθ = Sort[Select[Select[csθ, # > minimum@θ&], # < maximum@θ&], Less];

   If[

     csr =!= {} || csθ =!= {},
```

```
      Print[];
      Print[];

      Show[Graphics[Text[
  StyleForm["Considered Section",
      FontSize → 30, FontWeight → "Bold", FontFamily → "Times",
      FontColor → RGBColor[Random[ ], Random[ ], Random[ ]]],
    {0, 0}]], ImageSize → {650, 100}];

      If[

         csr =!= {},

         Do[

            r = csr[[i]];

            Print[];
            Print[];

            Print["Stress at the Section r = ", r, " m" ];

            Print[];

            Print["σrr at r = ", r, " m is ", stress@rr, " MPa"];
            Print["τrθ at r = ", r, " m is ", stress@rθ, " MPa"];

            "Graphics of σrr at r = csr[[i]]";

            polarsection1 = Plot[If[-10^-6 < stress@rr < 10^-6, 0, stress@rr],
      {theta, minimum@θ, maximum@θ},
            AspectRatio → 0.8,
            Axes → False,
            AxesLabel → None,
            AxesOrigin → Automatic,
            AxesStyle → None,
            Background → RGBColor[0.8, 1, 1],
            ColorOutput → Automatic,
            DefaultColor → RGBColor[0, 0, 0],
            DisplayFunction → Identity,
            FormatType → $FormatType,
            Frame → True,
            FrameLabel → {"θ (rad)", "normal stress (MPa)"},
            FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
            FrameTicks → Which[maximum@θ - minimum@θ == 2 Pi, {{0, Pi / 2, Pi,
         3 Pi / 2, 2 Pi}, Automatic, None, None}, maximum@θ - minimum@θ == Pi,
       {{0, Pi / 4, Pi / 2, 3 Pi / 4, Pi}, Automatic, None, None},
       maximum@θ - minimum@θ > 0, Automatic],
            GridLines → Which[
       maximum@θ - minimum@θ == 2 Pi, {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
```

```
 maximum@θ - minimum@θ == Pi, {{0, Pi / 4, Pi / 2, 3 Pi / 4, Pi}, Automatic},
 maximum@θ - minimum@θ > 0, Automatic],
     MaxBend → 10,
     PlotDivision → 20,
     PlotLabel → None,
     PlotPoints → 15,
     PlotRange → Automatic,
     PlotRegion → {{0, 0.95}, {0, 0.95}},
     PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
     RotateLabel → True,
     TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
     Ticks → Automatic];

     "Graphics of τrθ at r = csr[[i]]";

     polarsection2 = Plot[If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ],
{theta, minimum@θ, maximum@θ},
     AspectRatio → 0.8,
     Axes → False,
     AxesLabel → None,
     AxesOrigin → Automatic,
     AxesStyle → None,
     Background → RGBColor[0.8, 1, 1],
     ColorOutput → Automatic,
     DefaultColor → RGBColor[0, 0, 0],
     DisplayFunction → Identity,
     FormatType → $FormatType,
     Frame → True,
     FrameLabel → {"θ (rad)", "shear stress (MPa)"},
     FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
     FrameTicks → Which[maximum@θ - minimum@θ == 2 Pi, {{0, Pi / 2, Pi,
   3 Pi / 2, 2 Pi}, Automatic, None, None}, maximum@θ - minimum@θ == Pi,
 {{0, Pi / 4, Pi / 2, 3 Pi / 4, Pi}, Automatic, None, None},
 maximum@θ - minimum@θ > 0, Automatic],
     GridLines → Which[
 maximum@θ - minimum@θ == 2 Pi, {{0, Pi / 2, Pi, 3 Pi / 2, 2 Pi}, Automatic},
 maximum@θ - minimum@θ == Pi, {{0, Pi / 4, Pi / 2, 3 Pi / 4, Pi}, Automatic},
 maximum@θ - minimum@θ > 0, Automatic],
     MaxBend → 10,
     PlotDivision → 20,
     PlotLabel → None,
     PlotPoints → 15,
     PlotRange → Automatic,
     PlotRegion → {{0, 0.95}, {0, 0.95}},
     PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
     RotateLabel → True,
     TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
     Ticks → Automatic];
```

```
Show[
GraphicsArray[{polarsection1, polarsection2}], ImageSize → {500, 250}];

setequations = {stress@rr, stress@rθ};

θθmax = maximum@θ;
θθmin = minimum@θ;

pixelnumberθθ = 32;

Do[

  equation = setequations[[k]] ;

  theta = θθmin;

  maxvalue = equation;
  minvalue = equation;

  θθmaxvalue = theta;
  θθminvalue = theta;

  Do[

    theta = θθmin + (θθmax - θθmin) j / pixelnumberθθ;

    If[

      equation > maxvalue,

      maxvalue = equation; θθmaxvalue = theta

     ];

    If[

      equation < minvalue,

      minvalue = equation; θθminvalue = theta

     ],

    {j, 0, pixelnumberθθ}

   ];

  If[

    θθmaxvalue =!= θθmax && θθmaxvalue =!= θθmin,

    θθmaxbox = θθmaxvalue - (θθmax - θθmin) / pixelnumberθθ;
```

```
  Do[
      theta = θθmaxbox + (θθmax - θθmin) 2 j / pixelnumberθθ^2;
      If[
         equation > maxvalue,
         maxvalue = equation
        ],
      {j, 0, pixelnumberθθ}
     ]
  ];
If[
  θθminvalue =!= θθmax && θθminvalue =!= θθmin,
  θθminbox = θθminvalue - (θθmax - θθmin) / pixelnumberθθ;
  Do[
      theta = θθminbox + (θθmax - θθmin) 2 j / pixelnumberθθ^2;
      If[
         equation < minvalue,
         minvalue = equation
        ],
      {j, 0, pixelnumberθθ}
     ]
  ];
mav[k] = N[maxvalue];
miv[k] = N[minvalue];

theta =.,

{k, 1, Length[setequations]}

];
```

```
      Print["Maximum and Minimum Stress"];

      Print[];

      Print["σrr max = ", mav[1], " MPa"];
      Print["τrθ max = ", mav[2], " MPa"];

      Print[];

      Print["σrr min = ", miv[1], " MPa"];
      Print["τrθ min = ", miv[2], " MPa"];

      , {i, 1, Length[csr]}

     ];

   r =.

  ];

If[

   csθ =!= {},

   Do[

      theta = csθ[[i]];

      Print[];
      Print[];

      Print["Stress at the Section θ = ", theta, " rad" ];

      Print[];

      Print["σθθ at θ = ", theta , " rad is ", stress@θθ, " MPa"];
      Print["τrθ at θ = ", theta , " rad is ", stress@rθ, " MPa"];

      "Graphics of σθθ at θ = csθ[[i]]";

      polarsection3 = Plot[
If[-10^-6 < stress@θθ < 10^-6, 0, stress@θθ], {r, minimum@r, maximum@r}
      AspectRatio → 0.8,
      Axes → False,
      AxesLabel → None,
      AxesOrigin → Automatic,
      AxesStyle → None,
      Background → RGBColor[0.8, 1, 1],
      ColorOutput → Automatic,
      DefaultColor → RGBColor[0, 0, 0],
```

```
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"r (m)", "normal stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

"Graphics of τrθ at θ = csθ[[i]]";

polarsection4 = Plot[
If[-10^-6 < stress@rθ < 10^-6, 0, stress@rθ], {r, minimum@r, maximum@r}
AspectRatio → 0.8,
Axes → False,
AxesLabel → None,
AxesOrigin → Automatic,
AxesStyle → None,
Background → RGBColor[0.8, 1, 1],
ColorOutput → Automatic,
DefaultColor → RGBColor[0, 0, 0],
DisplayFunction → Identity,
FormatType → $FormatType,
Frame → True,
FrameLabel → {"r (m)", "shear stress (MPa)"},
FrameStyle → {AbsoluteThickness[1], RGBColor[0, 0, 0]},
FrameTicks → Automatic,
GridLines → Automatic,
MaxBend → 10,
PlotDivision → 20,
PlotLabel → None,
PlotPoints → 15,
PlotRange → Automatic,
PlotRegion → {{0, 0.95}, {0, 0.95}},
PlotStyle → {AbsoluteThickness[3], RGBColor[1, 0, 0]},
RotateLabel → True,
TextStyle →
{FontFamily → "Symbol", FontSlant → "Italic", FontSize → 10},
Ticks → Automatic];

Show[
```

```
GraphicsArray[{polarsection3, polarsection4}], ImageSize → {500, 250}];

        setequations = {stress@θθ, stress@rθ};

        rrmax = maximum@r;
        rrmin = minimum@r;

        pixelnumberrr = 32;

        Do[

            equation = setequations[[k]];

            r = rrmin;

            maxvalue = equation;
            minvalue = equation;

            rrmaxvalue = r;
            rrminvalue = r;

            Do[

                r = rrmin + (rrmax - rrmin) j / pixelnumberrr;

                If[

                    equation > maxvalue,

                    maxvalue = equation; rrmaxvalue = r

                  ];

                If[

                    equation < minvalue,

                    minvalue = equation; rrminvalue = r

                  ],

                {j, 0, pixelnumberrr}

              ];

            If[

                rrmaxvalue =!= rrmax && rrmaxvalue =!= rrmin,

                rrmaxbox = rrmaxvalue - (rrmax - rrmin) / pixelnumberrr;
```

```
  Do[
      r = rrmaxbox + (rrmax - rrmin) 2 j / pixelnumberrr^2;
      If[
         equation > maxvalue,
         maxvalue = equation
        ],
      {j, 0, pixelnumberrr}
     ]
   ];
If[
   rrminvalue =!= rrmax && rrminvalue =!= rrmin,
   rrminbox = rrminvalue - (rrmax - rrmin) / pixelnumberrr;
   Do[
      r = rrminbox + (rrmax - rrmin) 2 j / pixelnumberrr^2;
      If[
         equation < minvalue,
         minvalue = equation
        ],
      {j, 0, pixelnumberrr}
     ]
   ];
mav[k] = N[maxvalue];
miv[k] = N[minvalue];
r =.,
{k, 1, Length[setequations]}
];
```

```
        Print["Maximum and Minimum Stress"];

        Print[];

        Print["σθθ max = ", mav[1], " MPa"];
        Print["τrθ max = ", mav[2], " MPa"];

        Print[];

        Print["σθθ min = ", miv[1], " MPa"];
        Print["τrθ min = ", miv[2], " MPa"];

        , {i, 1, Length[csθ]}

       ];

     theta =.

   ]

  ]

];
```

ภาคผนวก ข

คู่มือการใช้โปรแกรม

ข.1 วัตถุประสงค์

โปรแกรมนี้เป็นโปรแกรมที่ใช้ในการวิเคราะห์ความเค้นและความเครียด ซึ่งคำนวณจากฟังก์ชันความเค้นของแอรี โดยที่ฟังก์ชันความเค้นนี้สอดคล้องกับสมการสมดุล สมการความสัมพันธ์ระหว่างองค์ประกอบความเค้น สมการความสัมพันธ์ระหว่างความเค้นและความเครียด และเงื่อนไขที่ขอบ การแสดงผลของโปรแกรมแบ่งออกเป็นสองส่วนคือ ส่วนแรกจะทำการคำนวณสมการขององค์ประกอบความเค้นและองค์ประกอบความเครียด ค่าองค์ประกอบความเค้นและองค์ประกอบความเครียด ณ จุดที่ต้องการทราบค่า ค่าสูงสุดและค่าต่ำสุดขององค์ประกอบความเค้นและองค์ประกอบความเครียดของวัสดุ สมการของความเค้นที่ขอบและที่หน้าตัดที่ต้องการพิจารณา และค่าสูงสุดและค่าต่ำสุดของความเค้นในแนวตั้งฉากและในแนวสำผัสกับหน้าตัดที่ต้องการพิจารณา ในส่วนที่สองจะทำการแสดงภาพการกระจายของความเค้นและความเครียด ความเค้นที่ขอบของวัสดุ และความเค้นที่หน้าตัดที่ต้องการพิจารณา

ข.2 ขอบเขต

โปรแกรมนี้ถูกออกแบบให้ใช้กับวัสดุที่มีคุณสมบัติเอกพันธ์ ไม่ขึ้นอยู่กับทิศทาง และยืดหยุ่นเชิงเส้นเท่านั้น โดยไม่คิดผลของความเค้นเนื่องจากอุณหภูมิ ลักษณะของปัญหาที่ทำการวิเคราะห์คือปัญหาแบบสองมิติในระบบพิกัดคาร์ทีเซียนและระบบพิกัดเชิงขั้ว

ข.3 ส่วนประกอบของโปรแกรม

ส่วนประกอบของโปรแกรมแบ่งออกเป็น 5 ส่วนคือ

ส่วนที่ 1. แนะนำโปรแกรม

ส่วนที่ 2. การป้อนข้อมูลชนิดของระบบพิกัดที่พิจารณา

ส่วนที่ 3. การป้อนข้อมูลค่าคงที่ของระบบพิกัดคาร์ทีเซียน

ส่วนที่ 4. การป้อนข้อมูลค่าคงที่ของระบบพิกัดเชิงขั้ว

ส่วนที่ 5. โปรแกรม

ข.4 วิธีการใช้งาน

เริ่มต้นการใช้งานโปรแกรมโดยการอ่านคำอธิบายในแต่ละส่วนของโปรแกรม แล้วจึงทำการป้อนข้อมูลโดยการใส่ค่าของตัวแปรทางด้านขวาของตัวแปรที่ขีดเส้นใต้ไว้หลังเครื่องหมาย = และลงท้ายด้วยเครื่องหมาย ; ทุกครั้ง โปรแกรมจะทำการคำนวณและแสดงผลภายหลังจากป้อนข้อมูลจนครบแล้วกด Enter การป้อนข้อมูลเริ่มต้นจากการเลือกชนิดของระบบพิกัดที่พิจารณาระหว่างพิกัดคาร์ทีเซียนหรือพิกัดเชิงขั้วในส่วนที่ 2 ของโปรแกรม ถ้าเลือกวิเคราะห์ความเค้นและความเครียดโดยใช้พิกัดคาร์ทีเซียนให้ทำการป้อนข้อมูลลงในส่วนที่ 3 ของโปรแกรม เมื่อป้อนข้อมูลในส่วนที่ 3 จนครบแล้วจึงสั่งให้โปรแกรมทำงานโดยกด Enter หรือถ้าเลือกวิเคราะห์ความเค้นและความเครียดโดยใช้พิกัดเชิงขั้วให้ทำการป้อนข้อมูลลงในส่วนที่ 4 ของโปรแกรมแทน แล้วทำตามขั้นตอนเหมือนกับแบบที่ 3 ข้อมูลที่ต้องป้อนในส่วนที่ 3 และ 4 ประกอบด้วย สมการของฟังก์ชันความเค้น ค่าคงที่ต่าง ๆ ที่ใช้ในการคำนวณ ลักษณะรูปร่างของวัสดุ และขอบเขตของวัสดุที่พิจารณา เมื่อโปรแกรมทำการแสดงผลจบลงแล้ว สามารถทำการวิเคราะห์ครั้งต่อไปโดยการเริ่มต้นป้อนข้อมูลใหม่ แล้วทำตามขั้นตอนข้างต้นอีกครั้งหนึ่ง

ข.5 การป้อนข้อมูลในระบบพิกัดคาร์ทีเซียน

ถ้าเลือกวิเคราะห์ความเค้นและความเครียดในระบบพิกัดคาร์ทีเซียน ให้เริ่มต้นโดยการป้อนข้อมูล type@of@coordinates = 1; ในส่วนที่ 2 ของโปรแกรม จากนั้นจึงป้อนข้อมูลในส่วนที่ 3 ของโปรแกรมซึ่งมีรายละเอียดดังนี้

degpolynomial

คืออันดับของฟังก์ชันความเค้นในรูปโพลีโนเมียลทุก ๆ อันดับที่มีเช่น สมการของฟังก์ชันความเค้นคือ $2xy + 3xy^2 + x^5$ จะได้ว่า degpolynomial = {2, 3, 5}; ถ้าในสมการของฟังก์ชันความเค้นไม่มีเทอมของโพลิโนเมียลให้กำหนด degpolynomial = { };

const@poly[1]-const@poly[4]

คือค่าสัมประสิทธิ์ของสมการโพลีโนเมียลดีกรีใด ๆ 4 ตัวแรกเรียงตามลำดับของอันดับของตัวแปร x จากมากไปหาน้อยตามสมการ

$$
\begin{aligned}
\Phi = \sum_{n=2}^{\infty}\Big(& \text{const@poly[1]}\,x^{n} + \\
& \text{const@poly[2]}\,x^{(n-1)}y + \\
& \text{const@poly[3]}\,x^{(n-2)}y^{2} + \\
& \text{const@poly[4]}\,x^{(n-3)}y^{3} + \\
& \text{const@poly[5]}\,x^{(n-4)}y^{4} + \\
& \cdots\cdots\cdots\cdots + \\
& \text{const@poly[n]}\,x\,y^{(n-1)} + \\
& \text{const@poly[n+1]}\,y^{n} \Big)\ \text{kN}
\end{aligned}
$$

สำหรับค่าสัมประสิทธิ์ของสมการโพลีโนเมียลในเทอมที่มากกว่า 4 โปรแกรมจะทำการคำนวณค่าสัมประสิทธิ์เหล่านั้นเองโดยอัตโนมัติ ในการป้อนค่าของ const@poly[1]-const@poly[4] ในสมการโพลีโนเมียลแต่ละอันดับ จะต้องป้อนหลักของค่าสัมประสิทธิ์ให้ตรงกับหลักของอันดับในตัวแปร degpolynomial ด้วยเช่น สมการของฟังก์ชันความเค้นคือ $2x^2+x^2y+3y^3+5x^2y^3-0.5y^5$

degpolynomial = {2, 3, 5};

const@poly[1] = {2, 0, 0};

const@poly[2] = {0, 1, 0};

const@poly[3] = {0, 0, 0};

const@poly[4] = {0, 3, 5};

const@four[1]-const@four[8]

คือค่าสัมประสิทธิ์ในสมการของฟังก์ชันความเค้นในรูปของอนุกรมฟูเรียร์ตามสมการ

$$\Phi = \sum_{n=1}^{\infty}\Bigg(\Big(\underline{\text{const@four[1]}}\cosh[\alpha y] + \underline{\text{const@four[2]}}\sinh[\alpha y] + \underline{\text{const@four[3]}}\alpha y\cosh[\alpha y] + \underline{\text{const@four[4]}}\alpha y\sinh[\alpha y]\Big)\cos[\alpha x] + \Big(\underline{\text{const@four[5]}}\cosh[\alpha y] + \underline{\text{const@four[6]}}\sinh[\alpha y] + \underline{\text{const@four[7]}}\alpha y\cosh[\alpha y] + \underline{\text{const@four[8]}}\alpha y\sinh[\alpha y]\Big)\sin[\alpha x]\Bigg)\ \text{kN}$$

ในการป้อนค่าสัมประสิทธิ์แต่ละเทอมสามารถเขียนให้อยู่ในรูปของตัวแปร n ได้เช่น const@four[1] = 1 / n; โดยที่ค่า $\alpha = n\pi$ / ความยาวในทิศทาง x

minimum@summation@rec

maximum@summation@rec

คือค่าเริ่มต้นและค่าสุดท้ายของตัวแปร n ในสมการของฟังก์ชันความเค้นในรูปของอนุกรมฟูเรียร์

type@of@problem@rec

คือรูปแบบชนิดของปัญหาในระบบสองมิติแบ่งออกเป็น

ปัญหาแบบความเครียดระนาบ

type@of@problem@rec = 1;

ปัญหาแบบความเค้นระนาบ

type@of@problem@rec = 2;

young@modu@rec

คือค่ายังก์โมดูลัสในหน่วย Gpa

shear@modu@rec

คือค่าเชียร์โมดูลัสในหน่วย Gpa

poisson@ratio@rec

คืออัตราส่วนของปัวซอง

minimum@x
maximum@x
minimum@y
maximum@y

คือขอบเขตของวัสดุที่พิจารณาในหน่วยเมตร ซึ่งในที่นี้จะกำหนดให้ x>>y

coordinaterec

คือพิกัดของจุดที่ต้องการทราบค่าองค์ประกอบความเค้นและองค์ประกอบความเครียดในหน่วยเมตรเช่น พิกัดที่ต้องการพิจารณาคือ (x, y) = (0, 0), (1, 1) จะได้ว่า coordinaterec = {(0, 0), (1, 1)}; ถ้าไม่ต้องการทราบค่าองค์ประกอบความเค้นและองค์ประกอบความเครียดที่จุดใด ๆ ให้กำหนด coordinaterec = { };

csx
csy

คือพิกัดที่ระนาบของหน้าตัดที่ต้องการพิจารณาตัดกับแกน x และแกน y ตามลำดับเช่น ต้องการพิจารณาความเค้นที่หน้าตัด x = 5,10 เมตร (เวคเตอร์ที่ตั้งฉากกับระนาบขนานกับแกน x) จะได้ว่า csx = {5,10}; ถ้าไม่ต้องการพิจารณาความเค้นที่หน้าตัดใด ๆ ให้กำหนด csx = { };

ข.6 การป้อนข้อมูลในระบบพิกัดเชิงขั้ว

ถ้าเลือกวิเคราะห์ความเค้นและความเครียดในระบบพิกัดเชิงขั้ว ให้เริ่มต้นโดยการป้อนข้อมูล type@of@coordinates = 2; ในส่วนที่ 2 ของโปรแกรม จากนั้นจึงป้อนข้อมูลในส่วนที่ 4 ของโปรแกรมซึ่งมีรายละเอียดดังนี้

const@polar[1]-const@polar[13]

คือค่าสัมประสิทธิ์ในสมการของฟังก์ชันความเค้นในรูปของพิกัดเชิงขั้วตามสมการ

$$\begin{aligned}
\Phi = {} & \text{const@polar[1]}\ln[r] + \\
& \text{const@polar[2]}r^2 + \\
& \text{const@polar[3]}r^2\ln[r] + \\
& \text{const@polar[4]}\, r^2\theta + \\
& \text{const@polar[5]}\theta + \\
& \text{const@polar[6]}r\theta\cos\theta + \\
& (\text{const@polar[7]}r^3 + \\
& \text{const@polar[8]}/r + \\
& \text{const@polar[9]}r\ln[r])\ \cos[\theta] + \\
& \text{const@polar[10]}r\theta\sin\theta + \\
& (\text{const@polar[11]}r^3 + \\
& \text{const@polar[12}/r + \\
& \text{const@polar[13]}r\ln[r])\ \sin[\theta]\ \text{kN}
\end{aligned}$$

const@polar[14]-const@polar[21]

คือค่าสัมประสิทธิ์ในสมการของฟังก์ชันความเค้นในรูปของอนุกรมฟูเรียร์ตามสมการ

$$\Phi = \sum_{n=2}^{\infty}\Bigg(\Big(\underline{\text{const@polar[14]}}r^{n} + \underline{\text{const@polar[15]}}r^{(n+2)} + \underline{\text{const@polar[16]}}r^{(-n)} + \underline{\text{const@polar[17]}}\,r^{(-n+2)}\Big)\cos n\theta + \Big(\underline{\text{const@polar[18]}}r^{n} + \underline{\text{const@polar[19]}}r^{(n+2)} + \underline{\text{const@polar[20]}}r^{(-n)} + \underline{\text{const@polar[21]}}\,r^{(-n+2)}\Big)\sin n\theta\Bigg)\ \text{kN}$$

ในการป้อนค่าสัมประสิทธิ์แต่ละเทอมสามารถเขียนให้อยู่ในรูปของตัวแปร n ได้เช่น const@polar[14] = 1 / n;

minimum@summation@pol
maximum@summation@pol

คือค่าเริ่มต้นและค่าสุดท้ายของตัวแปร n ในสมการของฟังก์ชันความเค้นในรูปของอนุกรมฟูเรียร์

type@of@problem@pol

คือรูปแบบชนิดของปัญหาในระบบสองมิติแบ่งออกเป็น
ปัญหาแบบความเครียดระนาบ
type@of@problem@pol = 1;
ปัญหาแบบความเค้นระนาบ
type@of@problem@pol = 2;

young@modu@pol

คือค่ายังก์โมดูลัสในหน่วย Gpa

shear@modu@pol

คือค่าเชียร์โมดูลัสในหน่วย Gpa

poisson@ratio@pol

คืออัตราส่วนของปัวซอง

minimum@r
maximum@r
minimum@θ
maximum@θ

คือขอบเขตของวัสดุที่พิจารณาในหน่วยเมตรและ rad ตามลำดับ โดยแบ่งออกเป็น 3 กรณีคือ
วัสดุเป็นแบบจานวงกลม
minimum@θ = 0;
maximum@θ = 2Pi;
วัสดุเป็นแบบครึ่งระนาบ
minimum@θ = -Pi/2;
maximum@θ = Pi/2;
วัสดุเป็นแบบส่วนตัดของวงกลม
minimum@θ และ maximum@θ อยู่ระหว่าง 0-2Pi โดยที่ maximum@θ > minimum@θ

coordinatepol

คือพิกัดของจุดที่ต้องการทราบค่าองค์ประกอบความเค้นและองค์ประกอบความเครียดในหน่วยเมตรและ rad ตามลำดับเช่น พิกัดที่ต้องการพิจารณาคือ $(r, \theta) = (1, \pi/2), (2, 2\pi)$ จะได้ว่า coordinatepol = {(1, Pi/2), (2, 2Pi)}; ถ้าไม่ต้องการทราบค่าองค์ประกอบความเค้นและองค์ประกอบความเครียดที่จุดใด ๆ ให้กำหนด coordinatepol = { };

csr
csθ

คือพิกัดที่ระนาบของหน้าตัดที่ต้องการพิจารณาตัดกับแกน r และแกน θ ตามลำดับเช่น ต้องการพิจารณาความเค้นที่หน้าตัด r = 5,10 เมตร (เวคเตอร์ที่ตั้งฉากกับระนาบขนานกับแกน r) จะได้ว่า csr = {5,10}; ถ้าไม่ต้องการพิจารณาความเค้นที่หน้าตัดใด ๆ ให้กำหนด csr = { };

ภาคผนวก ค

ตัวอย่างการใช้โปรแกรม

ตัวอย่างการใช้โปรแกรมในระบบพิกัดคาร์ทีเซียนกับปัญหาจากรูปที่ 4.13 โดยการกำหนดค่าตัวแปรต่าง ๆ ดังต่อไปนี้

```
type@of@coordinates = 1;
degpolynomial = {2, 3, 5};
const@poly[1] = {-25, 0, 0};
const@poly[2] = {0, -37.5, 0};
const@poly[3] = {0, 0, 0};
const@poly[4] = {0, -1245, 12.5};
const@four[1]- const@four[8] = 0;
minimum@summation@rec = 1;
maximum@summation@rec = 100;
type@of@problem@rec = 2;
young@modu@rec = 200;
shear@modu@rec = 77;
poisson@ratio@rec = 0.3;
minimum@x = -10;
maximum@x = -10;
minimum@y = -1;
maximum@y = 1;
coordinatereo = {{0, 1}, {5, 1}, {10, 0}};
csx = {0, 5};
csy = {0};
```

Program Stress Function

Stress Function = $-1245.\,y^3 + 0.\,x\,y^4 - 2.5\,y^5 + x^2\,(-25. - 37.5\,y + 12.5\,y^3)$ kN

Equations of Stress

$\sigma xx = -7.47\,y + 0.075\,x^2\,y + 0.\,x\,y^2 - 0.05\,y^3$ MPa

$\sigma yy = -0.05 - 0.075\,y + 0.025\,y^3$ MPa

$\tau xy = 0.\,y^3 + x\,(0.075 - 0.075\,y^2)$ MPa

Equations of Strain

$\epsilon xx = 0.075 - 37.2375\,y + 0.375\,x^2\,y + 0.\,x\,y^2 - 0.2875\,y^3$ μ

$\epsilon yy = -0.25 + 10.83\,y - 0.1125\,x^2\,y + 0.\,x\,y^2 + 0.2\,y^3$ μ

$\gamma xy = 0.\,y^3 + x\,(0.974026 - 0.974026\,y^2)$ μ

Stress Components at (x,y) = (5,0)

$\sigma xx = 0$ MPa

$\sigma yy = -0.05$ MPa

$\tau xy = 0.375$ MPa

Strain Components at (x,y) = (5,0)

$\epsilon xx = 0.075$ μ

$\epsilon yy = -0.25$ μ

$\gamma xy = 4.87013\ \mu$

Stress Components at (x,y) = (5,1)

$\sigma xx = -5.645$ MPa

$\sigma yy = -0.1$ MPa

$\tau xy = 0.$ MPa

Strain Components at (x,y) = (5,1)

$\epsilon xx = -28.075\ \mu$

$\epsilon yy = 7.9675\ \mu$

$\gamma xy = 0.\ \mu$

Boundary Region is x = -10 to 10 m and y = -1 to 1 m

Maximum and Minimum Stress

σxx max = 7.52 MPa

σyy max = -6.93889×10^{-18} MPa

τxy max = 0.75 MPa

σxx min = -7.52 MPa

σyy min = -0.1 MPa

τxy min = -0.75 MPa

Maximum and Minimum Strain

ϵxx max = 37.6 μ

ϵyy max = 10.78 μ

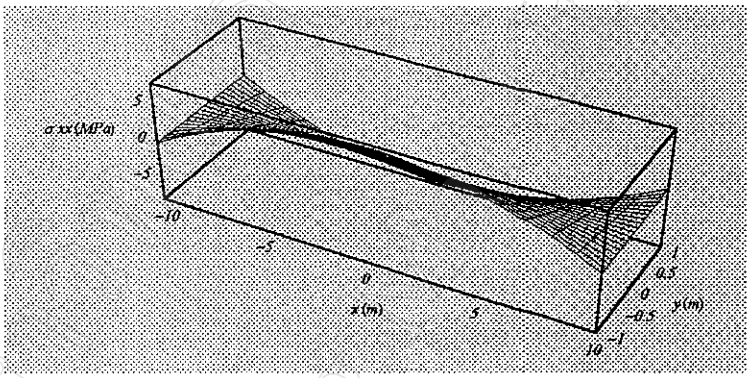
γxy max = 9.74026 μ

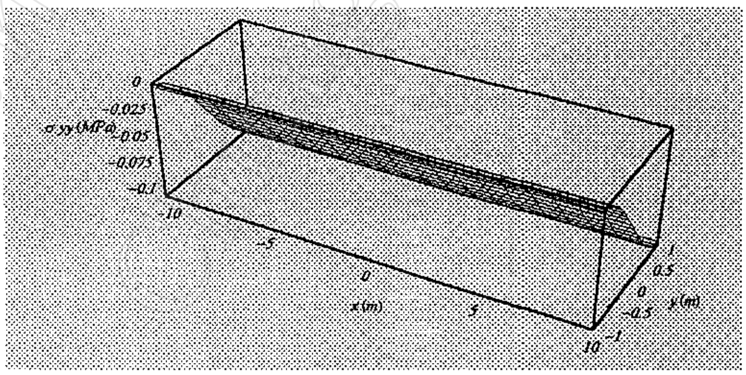
ϵxx min = -37.45 μ

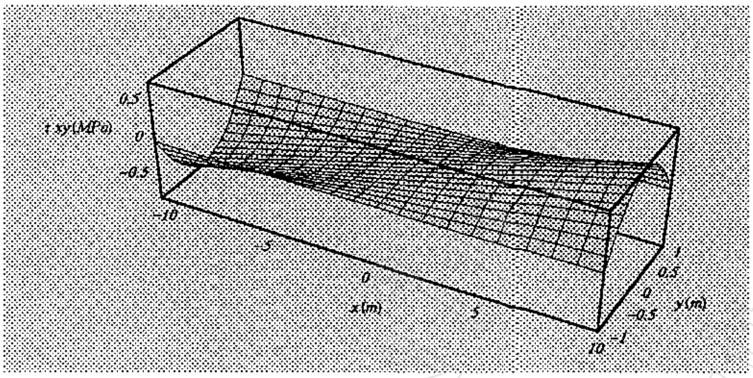
ϵyy min = -11.28 μ

γxy min = -9.74026 μ

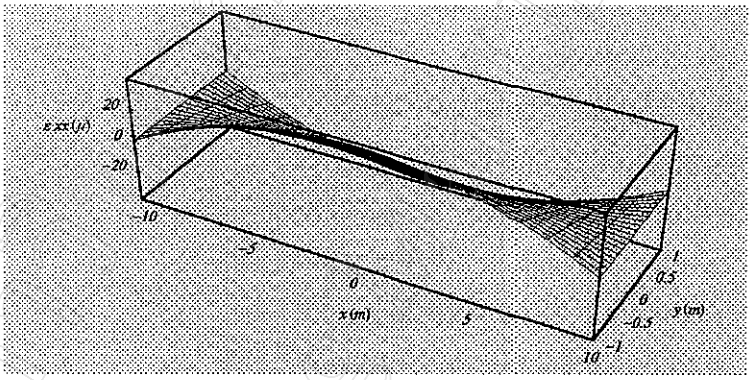
Stress Distribution

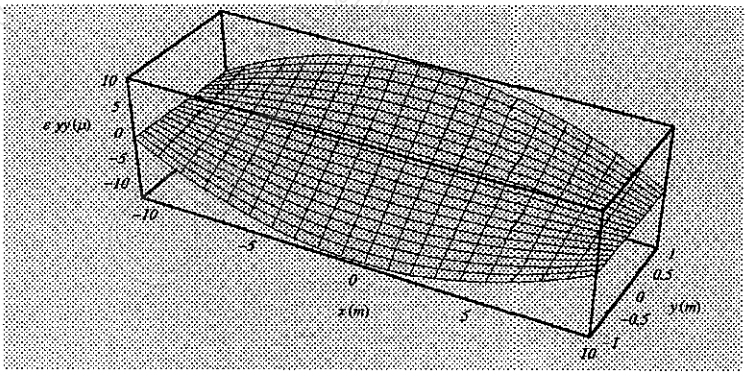


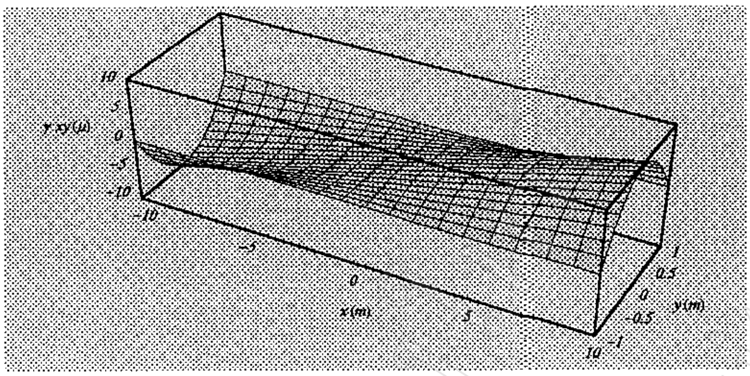




Strain Distribution





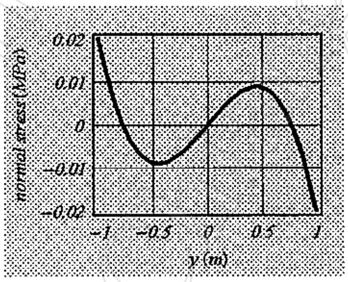


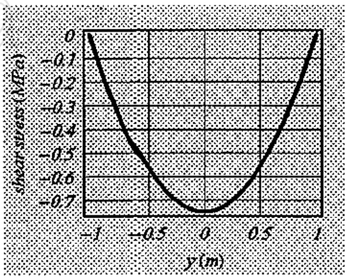
Boundary Conditions

Stress at the Boundary x = -10 m

σxx at x = -10 m is $0.03\,y + 0.\,y^2 - 0.05\,y^3$ MPa

τxy at x = -10 m is $0.\,y^3 - 10\,(0.075 - 0.075\,y^2)$ MPa

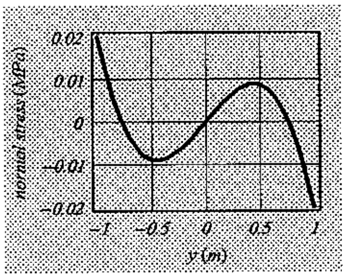


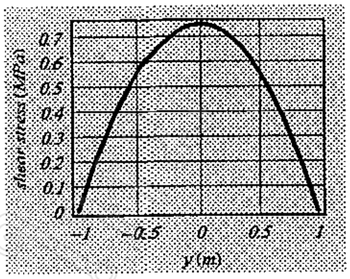


Stress at the Boundary x = 10 m

σxx at x = 10 m is $0.03\,y + 0.\,y^2 - 0.05\,y^3$ MPa

τxy at x = 10 m is $0.\,y^3 + 10\,(0.075 - 0.075\,y^2)$ MPa

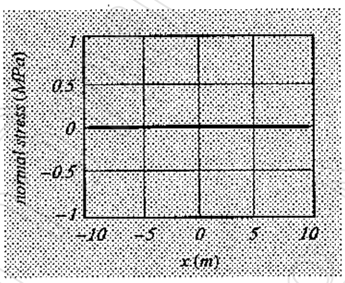


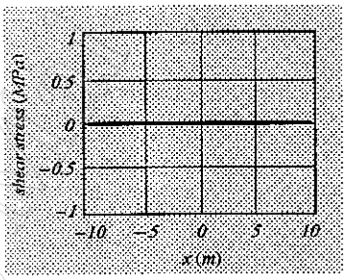


Stress at the Boundary y = -1 m

σyy at y = -1 m is -6.93889×10^{-18} MPa

τxy at y = -1 m is 0. + 0. x MPa

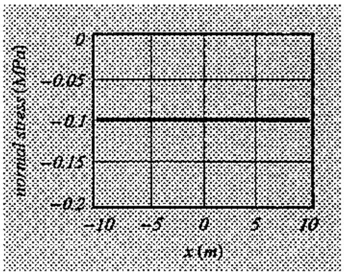


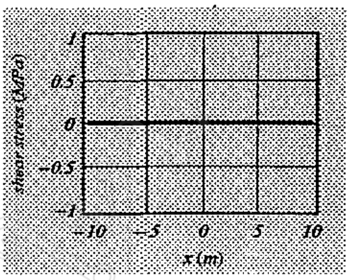


Stress at the Boundary y = 1 m

σyy at y = 1 m is -0.1 MPa

τzy at y = 1 m is 0. + 0. x MPa



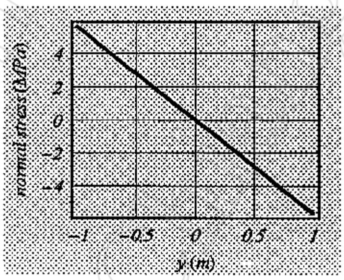


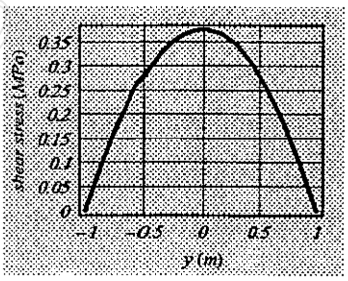
Considered Section

Stress at the Section x = 5 m

σxx at x = 5 m is $-5.595\,y + 0.\,y^2 - 0.05\,y^3$ MPa

τxy at x = 5 m is $0.\,y^3 + 5\,(0.075 - 0.075\,y^2)$ MPa





Maximum and Minimum Stress

σxx max = 5.645 MPa

τxy max = 0.375 MPa

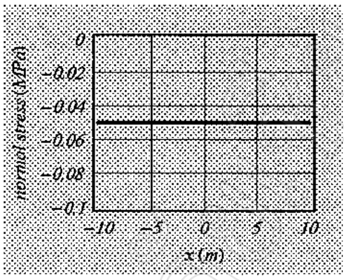
σxx min = -5.645 MPa

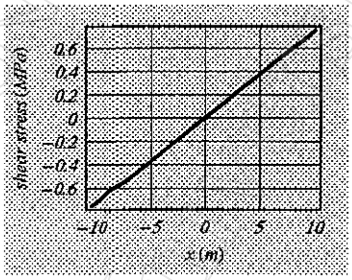
τxy min = 0. MPa

Stress at the Section y = 0 m

σyy at y = 0 m is -0.05 MPa

τxy at y = 0 m is 0.075 x MPa





Maximum and Minimum Stress

σyy max = -0.05 MPa

τxy max = 0.75 MPa

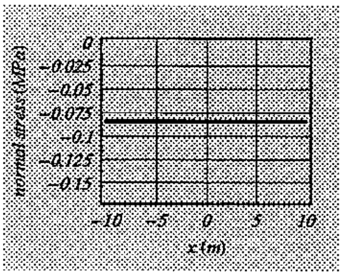
σyy min = -0.05 MPa

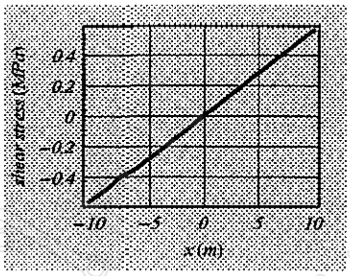
τxy min = -0.75 MPa

Stress at the Section y = 0.5 m

σyy at y = 0.5 m is -0.084375 MPa

τxy at y = 0.5 m is 0. + 0.05625 x MPa





Maximum and Minimum Stress

σyy max = -0.084375 MPa

τxy max = 0.5625 MPa

σyy min = -0.084375 MPa

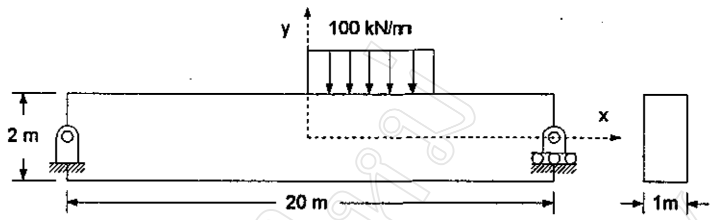
τxy min = -0.5625 MPa

ภาคผนวก ง

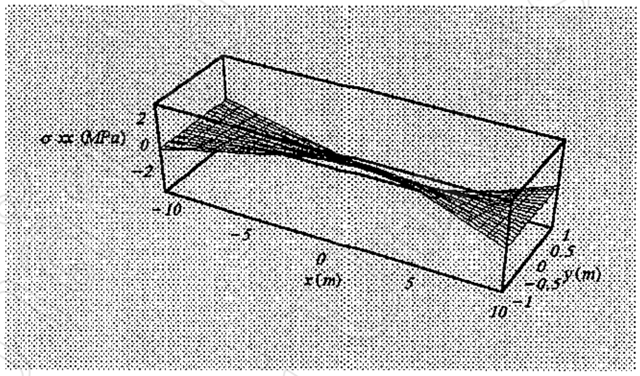
ตัวอย่างการประยุกต์ใช้งาน

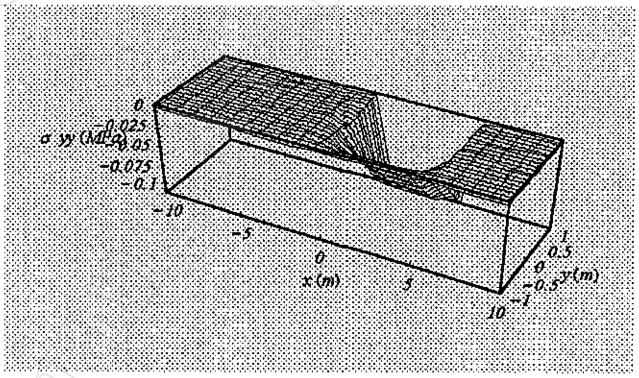
ตัวอย่างต่อไปนี้เป็นการนำโปรแกรมไปประยุกต์ใช้วิเคราะห์การรับภาระในรูปแบบอื่น ๆ ที่น่าสนใจ โดยค่าสัมประสิทธิ์ในสมการของฟังก์ชันความเค้นเหล่านี้ได้มาจากเงื่อนไขที่ขอบ ซึ่งในแต่ละตัวอย่างจะแสดงเฉพาะภาพการกระจายความเค้นเท่านั้น

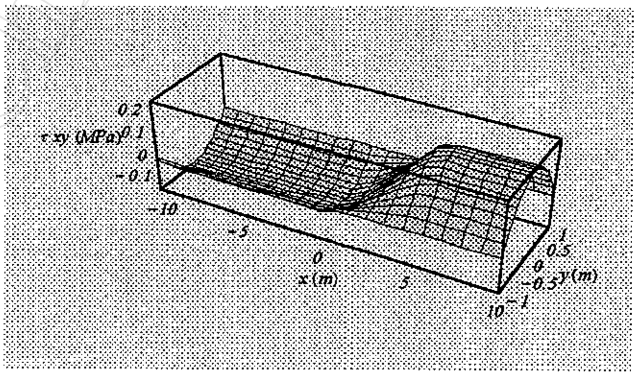
ตัวอย่างที่ 1



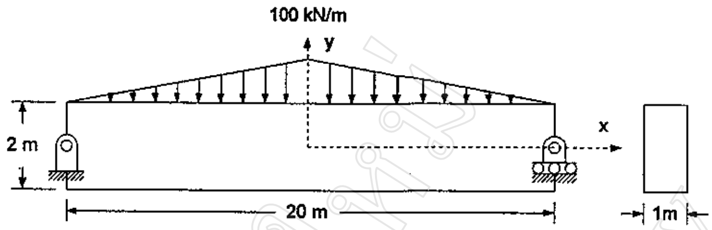
ภาพการกระจายความเค้น σ_{xx} , σ_{yy} และ τ_{xy} ตามลำดับ



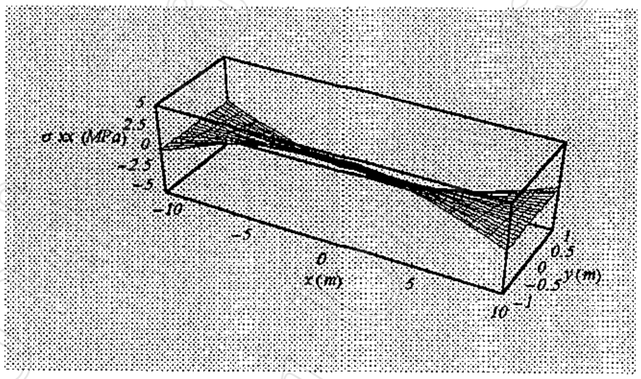


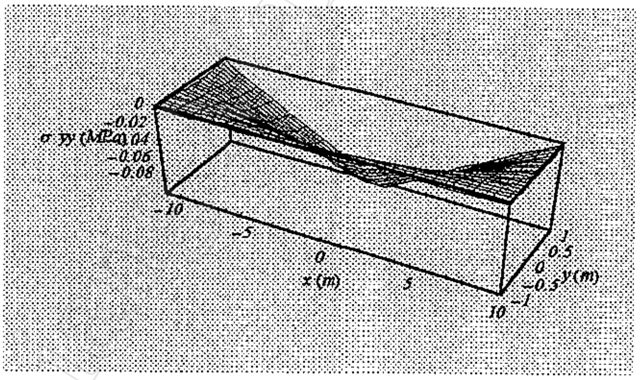


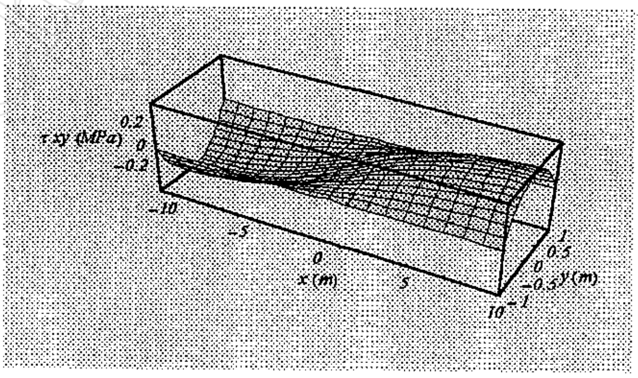
ตัวอย่างที่ 2



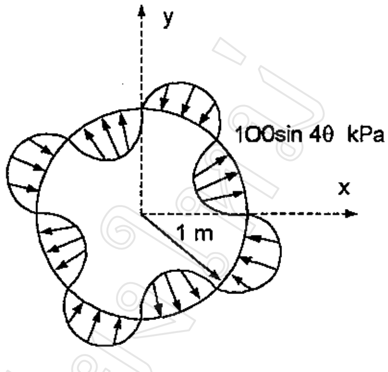
ภาพการกระจายความเค้น σ_{xx} , σ_{yy} และ τ_{xy} ตามลำดับ



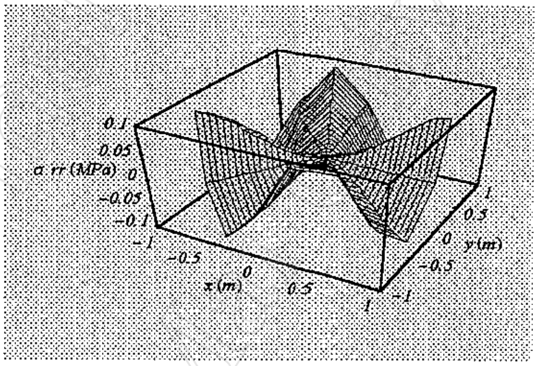


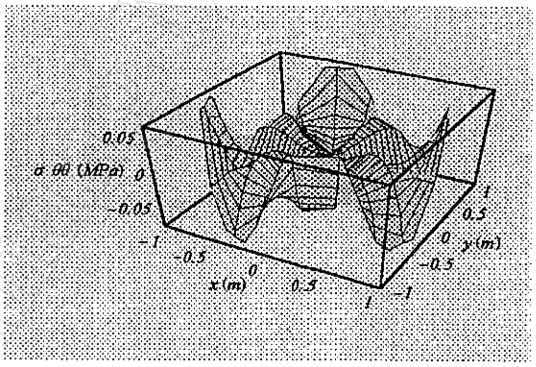


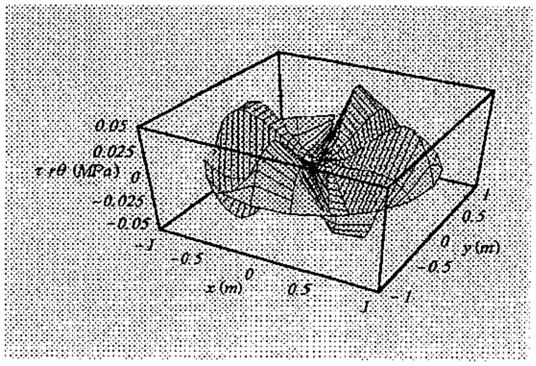
ตัวอย่างที่ 3



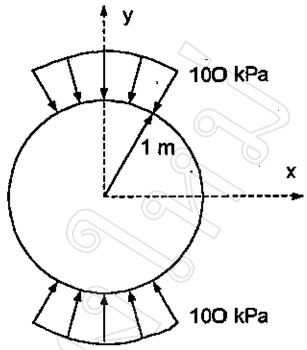
ภาพการกระจายความเค้น σ_{rr} , $\sigma_{\theta\theta}$ และ $\tau_{r\theta}$ ตามลำดับ



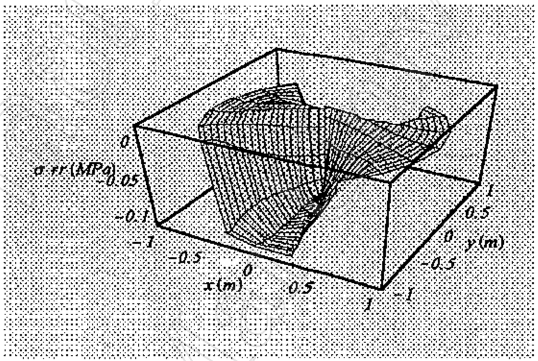


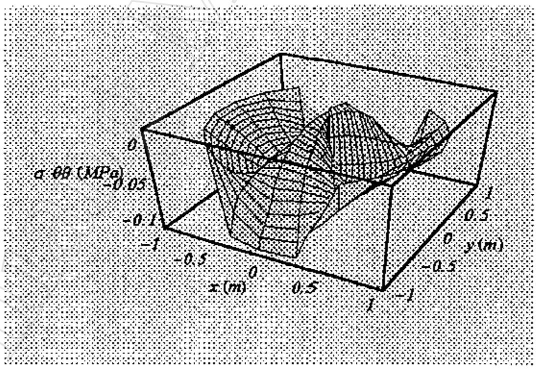


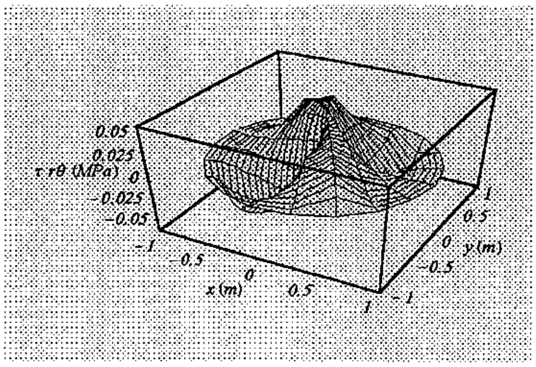
ตัวอย่างที่ 4



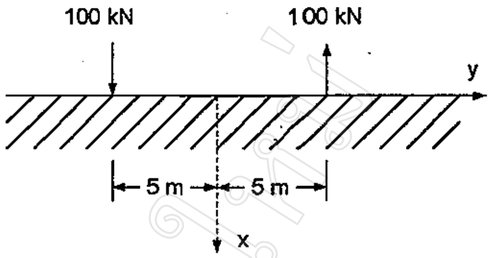
ภาพการกระจายความเค้น σ_{rr} , $\sigma_{\theta\theta}$ และ $\tau_{r\theta}$ ตามลำดับ



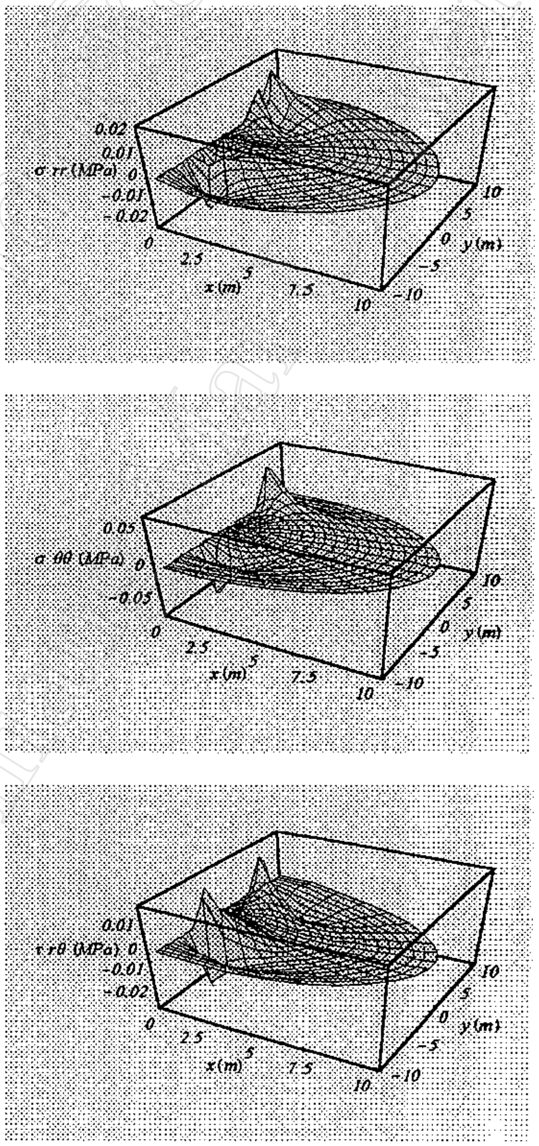




ตัวอย่างที่ 5



ภาพการกระจายความเค้น σ_{rr} , $\sigma_{\theta\theta}$ และ $\tau_{r\theta}$ ตามลำดับ



ประวัติผู้เขียน

ชื่อ	นายณัฏฐ์ กาศยปนันทน์
วัน เดือน ปี เกิด	6 กรกฎาคม 2519
ประวัติการศึกษา	สำเร็จการศึกษามัธยมศึกษาตอนต้น โรงเรียนเตรียมอุดมศึกษาพัฒนาการ ปีการศึกษา 2533 สำเร็จการศึกษามัธยมศึกษาตอนปลาย กศน. ปีการศึกษา 2534 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล มหาวิทยาลัยเชียงใหม่ ปีการศึกษา 2539