In[235]:=
```
Off[rmnsm];
Remove["Global`*"]
Off[General::spell1]
Off[Solve::"ifun"]
```

1. Introducción de los datos iníciales

Unidades

bi	Ancho de los elementos (cm)
t	Espesor de los elementos (cm)
v	Coeficiente de Poisson
e	Modulo de elasticidad en kN/cm2
uni	Valor del desplazamiento asociado a cada nodo (cm)
L	Longitud del perfil (cm)
n	Indice de precisión de los gráficos

Notaciones

Se nota por Xinj, el valor Xi al mover unitariamente el nudo j.

In[239]:=
```
b1 = 1.31;
b2 = 6.88;
b3 = 8.97;
t = 0.167;
v = 0.3;
e = 20000;
un1 = un2 = un3 = un4 = un5 = un6 = 1;
L = 200;
n = 100;
```

2. Movimiento de solido rígido de la sección

Coordenadas iníciales de los nodos del perfil

In[248]:=
```
P1 = {b1, 0};
P2 = {0, 0};
P3 = {0, b2};
P4 = {b3, b2};
P5 = {b3, 0};
P6 = {b3 - b1, 0};
```

Angulos iníciales entre los elementos del perfil

In[254]:=
```
T2 = T3 = T4 = T5 = Pi / 2;
T02 = T05 = Pi / 2;
```

■ **Estado de la seccion antes de la deformación**

In[256]:=
```
EI = ListPlot[{P1, P2, P3, P4, P5, P6}, PlotStyle → {AbsolutePointSize[10], RGBColor[1, 0, 0],
     AbsoluteThickness[3]}, Axes → False, PlotJoined → True, ImageSize → 600, AspectRatio → Automatic];
```

2.1. Movimiento del nudo 3

■ **Determinación de la posición del nodo 2 al mover el nodo 3: P2n3'**

In[257]:=
```
P2n3' = {0, -un3 / b2}
```
Out[257]=
```
{0, -0.145349}
```

■ **Determinación de la posición del nodo 3 al mover el nodo 3: P3n3'**

In[258]:=
```
P3n3' = {un3 / b3, b2 - un3 / b2}
```
Out[258]=
```
{0.111483, 6.73465}
```

■ **Determinación de la posición del nodo 4 al mover el nodo 3: P4n3'**

In[259]:=
```
P4n3' = {b3 + un3 / b3, b2}
```
Out[259]=
```
{9.08148, 6.88}
```

■ **Determinación de la posición del nodo 5 al mover el nodo 3: P5n3'**

In[260]:=
```
P5n3' = P5
```
Out[260]=
```
{8.97, 0}
```

■ **Determinación de la posición del nodo 6 al mover el nodo 3: P6n3'**

In[261]:=
```
N[s1n3 = {X6n3', Y6n3'} /. Solve[{(X6n3' - P5n3'[[1]])^2 + (Y6n3' - P5n3'[[2]])^2 == b1^2, (P5n3'[[1]] - X6n3') *
        (P4n3'[[1]] - P5n3'[[1]]) + (P5n3'[[2]] - Y6n3') * (P4n3'[[2]] - P5n3'[[2]]) == 0}, {X6n3', Y6n3'}]];
N[P6n3' = If[s1n3[[2, 1]] <= P5[[1]], s1n3[[2]], s1n3[[1]]]]
```
Out[262]=
```
{7.66017, 0.0212243}
```

- **Determinación de la posición del nodo 1 al mover el nodo 3: P1n3'**

In[263]:=
```
N[s2n3 = {X1n3', Y1n3'} /. Solve[{(X1n3' - P2n3'[[1]])^2 + (Y1n3' - P2n3'[[2]])^2 == b1^2, (X1n3' - P2n3'[[1]]) *
        (P3n3'[[1]] - P2n3'[[1]]) + (Y1n3' - P2n3'[[2]]) * (P3n3'[[2]] - P2n3'[[2]]) == 0}, {X1n3', Y1n3'}]];
N[P1n3' = If[s2n3[[2, 1]] >= P2n3'[[1]], s2n3[[2]], s2n3[[1]]]]
```
Out[264]=
```
{1.30983, -0.166573}
```

- **Deformada de la sección como un sólido rígido**

In[265]:=
```
CGn3 = ListPlot[{P1n3', P2n3', P3n3', P4n3', P5n3', P6n3'}, PlotStyle → {AbsolutePointSize[10],
     RGBColor[0, 0, 1], AbsoluteThickness[3]}, Axes → False, PlotJoined → True, ImageSize → 600,
   AspectRatio → Automatic];
```

- **Superposición de la deformada al estado inicial**

In[266]:=
```
Show[EI, CGn3, ImageSize → 600, AspectRatio → Automatic];
```

- **Determinación del ángulo θ3' al mover el nodo 3: T03n3**

In[267]:=
```
N[S3n3 = T03n3 /. Solve[((P3n3'[[1]] - P2n3'[[1]]) * (P3n3'[[1]] - P4n3'[[1]]) + (P3n3'[[2]] - P2n3'[[2]])
          * (P3n3'[[2]] - P4n3'[[2]])) /
       ( ((P3n3'[[1]] - P2n3'[[1]])^2 + (P3n3'[[2]] - P2n3'[[2]])^2)      *
          ((P3n3'[[1]] - P4n3'[[1]])^2 + (P3n3'[[2]] - P4n3'[[2]])^2) )^(1/2) == Cos[Pi - T03n3], T03n3]];
N[T03n3 = If[S3n3[[1]] ≥ 0, S3n3[[1]], S3n3[[2]]]];
N[T03n3] * 180 / (Pi)
```

Out[269]=
```
88.1433
```

- **Determinación del ángulo θ4' al mover el nodo 3: T04n3**

In[270]:=
```
N[S4n3 = T04n3 /. Solve[((P4n3'[[1]] - P3n3'[[1]]) * (P4n3'[[1]] - P5n3'[[1]]) + (P4n3'[[2]] - P3n3'[[2]]) *
          (P4n3'[[2]] - P5n3'[[2]])) /
       ( ((P4n3'[[1]] - P3n3'[[1]])^2 + (P4n3'[[1]] - P5n3'[[1]])^2) *
          ((P4n3'[[2]] - P3n3'[[2]])^2 + (P4n3'[[2]] - P5n3'[[2]])^2))^(1/2) == Cos[Pi - T04n3], T04n3]];
N[T04n3 = If[S4n3[[1]] ≥ 0, S4n3[[1]], S4n3[[2]]]];
N[T04n3] * 180 / (Pi)
```

Out[272]=
```
91.8566
```

- **Calculo de los momentos hiperestáticos (M32, M43) al mover el nodo 3**

In[273]:=
```
i = t^3 / (12 * (1 - v^2)^2);
(T = {T3, T4}) // MatrixForm;
(T0n3 = {N[T03n3], N[T04n3]}) // MatrixForm;
(Mn3 = {M32n3, M43n3}) // MatrixForm;
(An3 = (1 / (e * i)) * {
      {-(b2 / 3 + b3 / 3), -b3 / 6},
      {-b3 / 6, -(b3 / 3 + b2 / 3)}
     }) // MatrixForm;
(S5n3 = {M32n3, M43n3} /. Solve[T + An3.Mn3 == T0n3, Mn3]) // MatrixForm
```

```
( 0.0801816  -0.0801799 )
```

- **Determinación del giro de los ángulos (θ2, θ3, θ4, θ5) al mover el nodo 3**

In[279]:=
```
M32n3 = S5n3[[1, 1]];
M43n3 = S5n3[[1, 2]];
θ2n3 = -L / Pi * (1 / 2 * M32n3 / (3 * e * i / b2) + ArcTan[1 / (b3 * b2)]);
θ3n3 = L / Pi * (M32n3 / (3 * e * i / b2) - ArcTan[1 / (b3 * b2)]);
θ4n3 = L / Pi * (M43n3 / (3 * e * i / b3) + 1 / 2 * M32n3 / (3 * e * i / b3) + ArcTan[1 / (b3 * b2)]);
θ5n3 = L / Pi * (1 / 2 * M43n3 / (3 * e * i / b2) - ArcTan[1 / (b3 * b2)]);
(θn3 = {θ2n3, θ2n3, θ3n3, θ4n3, θ5n3, θ5n3}) // MatrixForm
```

```
( -1.6559  )
( -1.6559  )
( 0.217358 )
( 0.21741  )
( -1.65589 )
( -1.65589 )
```

2.2. Movimiento del nodo 1

- **Determinación de la posición del nodo 3 al mover el nodo 1: P3n1'**

In[286]:=
```
P3n1' = P3
```

Out[286]=
```
{0, 6.88}
```

- **Determinación de la posición del nodo 4 al mover el nodo 1: P4n1'**

In[287]:=
```
P4n1' = P4
```

Out[287]=
```
{8.97, 6.88}
```

■ **Determinación de la posición del nodo 5 al mover el nodo 1: P5n1'**

In[288]:=
```
P5n1' = P5
```
Out[288]=
```
{8.97, 0}
```

■ **Determinación de la posición del nodo 6 al mover el nodo 1: P6n1'**

In[289]:=
```
P6n1' = P6
```
Out[289]=
```
{7.66, 0}
```

■ **Determinación de la posición del nodo 2 al mover el nodo 1: P2n1'**

In[290]:=
```
P2n1' = {-1 / b1, 0}
```
Out[290]=
```
{-0.763359, 0}
```

■ **Determinación de la posición del nodo 1 al mover el nodo 1: P1n1'**

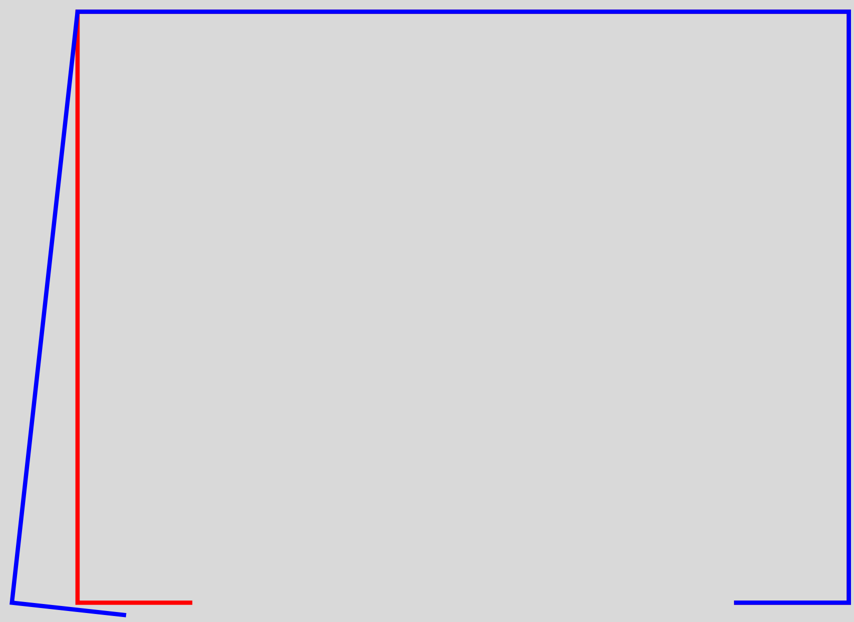
In[291]:=
```
N[s1n1 = {X1n1', Y1n1'} /. Solve[{(X1n1' - P2n1'[[1]])^2 + (Y1n1' - P2n1'[[2]])^2 == b1^2,
      (X1n1' - P2n1'[[1]]) * (P3n1'[[1]] - P2n1'[[1]]) + (Y1n1' - P2n1'[[2]]) * (P3n1'[[2]] - P2n1'[[2]]) == 0},
     {X1n1', Y1n1'}]];
N[P1n1' = If[s1n1[[2, 1]] >= P2n1'[[1]], s1n1[[2]], s1n1[[1]]]]
```
Out[292]=
```
{0.538651, -0.144462}
```

■ **Deformada de la sección como un solido rígido al mover el nodo 1**

In[293]:=
```
CGn1 = ListPlot[{P1n1', P2n1', P3n1', P4n1', P5n1', P6n1'}, PlotStyle → {AbsolutePointSize[10],
     RGBColor[0, 0, 1], AbsoluteThickness[3]}, Axes → False, PlotJoined → True, ImageSize → 600,
   AspectRatio → Automatic];
```

■ Superposición de la deformada con la geometría inicial

In[294]:=
```
Show[EI, CGn1, ImageSize → 600, AspectRatio → Automatic];
```



■ Determinación del ángulo θ4' al mover el nodo 1: T04n1

In[295]:=
```
T04n1 = Pi / 2
```

Out[295]= $\frac{\pi}{2}$

■ Determinación del ángulo θ3' al mover el nodo 1: T03n1

In[296]:=
```
N[S2n1 = T03n1 /. Solve[((P3n1'[[1]] - P2n1'[[1]]) * (P3n1'[[1]] - P4n1'[[1]]) + (P3n1'[[2]] - P2n1'[[2]])
          * (P3n1'[[2]] - P4n1'[[2]])) /
       ( ((P3n1'[[1]] - P2n1'[[1]])^2 + (P3n1'[[2]] - P2n1'[[2]])^2)      *
          ((P3n1'[[1]] - P4n1'[[1]])^2 + (P3n1'[[2]] - P4n1'[[2]])^2) )^(1 / 2) == Cos[Pi - T03n1], T03n1]];
N[T03n1 = If[S2n1[[1]] ≥ 0, S2n1[[1]], S2n1[[2]]]];
N[T03n1] * 180 / (Pi)
```

Out[298]= 83.6687

■ Calculo de los momentos hiperestáticos (M32, M43) al mover el nodo 1

In[299]:=
```
(T = {T3, T4}) // MatrixForm;
(T0n1 = {N[T03n1], N[T04n1]}) // MatrixForm;
(Mn1 = {M32n1, M43n1}) // MatrixForm;
(An1 = (1 / (e * i)) * {
      {-(b2 / 3 + b3 / 3), -b3 / 6},
      {-b3 / 6, -(b3 / 3 + b2 / 3)}
     }) // MatrixForm;
(S3n1 = {M32n1, M43n1} /. Solve[T + An1.Mn1 == T0n1, Mn1]) // MatrixForm
```

(0.213118 -0.060305)

- **Determinación del giro de los ángulos (θ2, θ3, θ4, θ5) al mover el nodo 1**

In[304]:=
```
M32n1 = S3n1[[1, 1]];
M43n1 = S3n1[[1, 2]];
θ2n1 = L / Pi * (-1 / 2 * M32n1 / (3 * e * i / b2) - ArcTan[1 / (b1 * b2)]);
θ3n1 = L / Pi * (M32n1 / (3 * e * i / b2) - ArcTan[1 / (b1 * b2)]);
θ4n1 = L / Pi * (-M43n1 / (3 * e * i / b2));
θ5n1 = L / Pi * (1 / 2 * M43n1 / (3 * e * i / b2));
(θn1 = {θ2n1, θ2n1, θ3n1, θ4n1, θ5n1, θ5n1}) // MatrixForm
```

```
( -8.6944   )
( -8.6944   )
( -3.71539  )
( 0.939258  )
( -0.469629 )
( -0.469629 )
```

2.3. Movimiento del nodo 2

- **Determinación de la posición del nodo 4 al mover el nodo 2: P4n2'**

In[311]:=
```
P4n2' = P4
```
Out[311]=
```
{8.97, 6.88}
```

- **Determinación de la posición del nodo 5 al mover el nodo 2: P5n2'**

In[312]:=
```
P5n2' = P5
```
Out[312]=
```
{8.97, 0}
```

- **Determinación de la posición del nodo 6 al mover el nodo 2: P6n2'**

In[313]:=
```
P6n2' = P6
```
Out[313]=
```
{7.66, 0}
```

- **Determinación de la posición del nodo 3 al mover el nodo 2: P3n2'**

In[314]:=
```
P3n2' = {0, b2 + 1 / b2}
```
Out[314]=
```
{0, 7.02535}
```

- **Determinación de la posición del nodo 2 al mover el nodo 2: P2n2'**

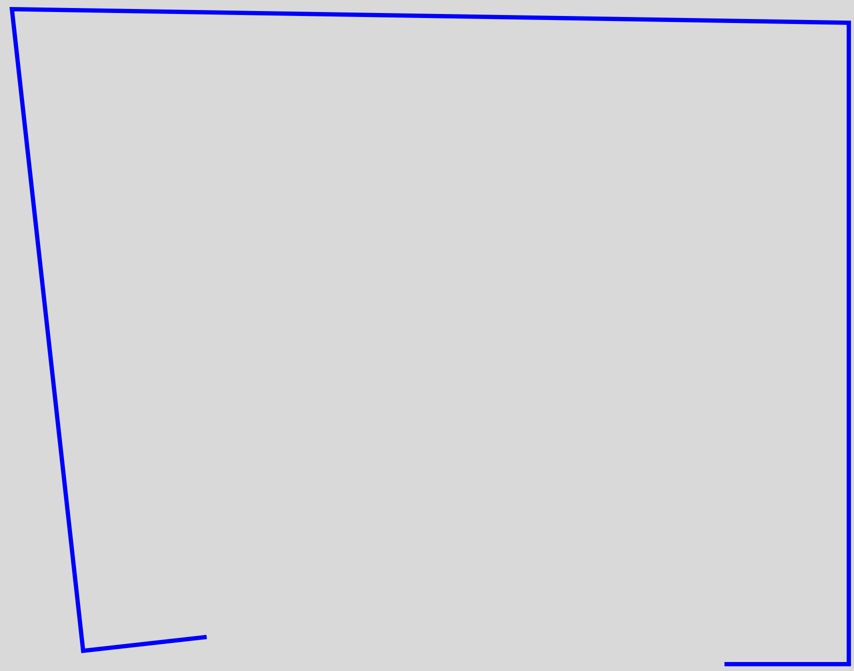
In[315]:=
```
P2n2' = {1 / b1, 1 / b2}
```
Out[315]=
```
{0.763359, 0.145349}
```

- **Determinación de la posición del nodo 1 al mover el nodo 2: P1n2'**

In[316]:=
```
N[s1n2 = {X1n2', Y1n2'} /. Solve[{(X1n2' - P2n2'[[1]])^2 + (Y1n2' - P2n2'[[2]])^2 == b1^2,
      (X1n2' - P2n2'[[1]]) * (P3n2'[[1]] - P2n2'[[1]]) + (Y1n2' - P2n2'[[2]]) * (P3n2'[[2]] - P2n2'[[2]]) == 0},
     {X1n2', Y1n2'}]];
N[P1n2' = If[s1n2[[2, 1]] >= P2n2'[[1]], s1n2[[2]], s1n2[[1]]]]
```
Out[317]=
```
{2.06537, 0.289811}
```

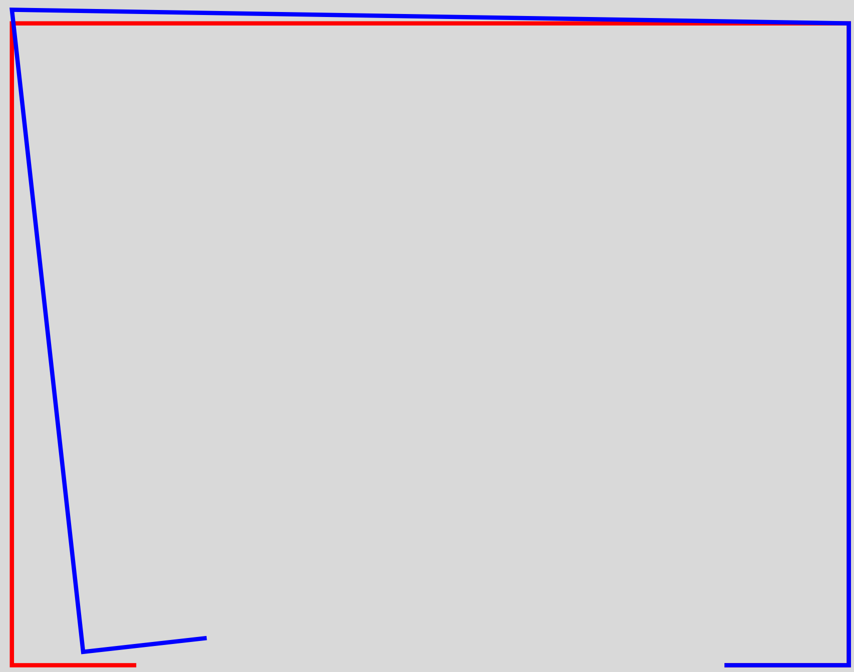
■ **Deformada de la sección como un solido rígido al mover el nodo 2**

In[318]:=
```
CGn2 = ListPlot[{P1n2', P2n2', P3n2', P4n2', P5n2', P6n2'}, PlotStyle → {AbsolutePointSize[10],
     RGBColor[0, 0, 1], AbsoluteThickness[3]}, Axes → False, PlotJoined → True, ImageSize → 600,
   AspectRatio → Automatic];
```



■ **Superposición de la deformada con la geometría inicial**

In[319]:=
```
Show[EI, CGn2, ImageSize → 600, AspectRatio → Automatic];
```



- **Determinación del ángulo θ3' al mover el nodo 2: T03n2**

In[320]:=
```
N[S2n2 = T03n2 /. Solve[((P3n2'[[1]] - P2n2'[[1]]) * (P3n2'[[1]] - P4n2'[[1]]) + (P3n2'[[2]] - P2n2'[[2]])
          * (P3n2'[[2]] - P4n2'[[2]])) /
       ( ((P3n2'[[1]] - P2n2'[[1]])^2 + (P3n2'[[2]] - P2n2'[[2]])^2)      *
          ((P3n2'[[1]] - P4n2'[[1]])^2 + (P3n2'[[2]] - P4n2'[[2]])^2) )^(1/2) == Cos[Pi - T03n2], T03n2]];
N[T03n2 = If[S2n2[[1]] ≥ 0, S2n2[[1]], S2n2[[2]]]];
N[T03n2] * 180 / (Pi)
```

Out[322]=
```
97.2596
```

- **Determinación del ángulo θ4' al mover el nodo 2: T04n2**

In[323]:=
```
N[S3n2 = T04n2 /. Solve[((P4n2'[[1]] - P3n2'[[1]]) * (P4n2'[[1]] - P5n2'[[1]]) + (P4n2'[[2]] - P3n2'[[2]]) *
          (P4n2'[[2]] - P5n2'[[2]])) /
       ( ((P4n2'[[1]] - P3n2'[[1]])^2 + (P4n2'[[1]] - P5n2'[[1]])^2) *
          ((P4n2'[[2]] - P3n2'[[2]])^2 + (P4n2'[[2]] - P5n2'[[2]])^2))^(1/2) == Cos[Pi - T04n2], T04n2]];
N[T04n2 = If[S3n2[[1]] ≥ 0, S3n2[[1]], S3n2[[2]]]];
N[T04n2] * 180 / (Pi)
```

Out[325]=
```
89.0718
```

- **Calculo de los momentos hiperestáticos (M32, M43) al mover el nodo 2**

In[326]:=
```
(T = {T3, T4}) // MatrixForm;
(T0n2 = {N[T03n2], N[T04n2]}) // MatrixForm;
(Mn2 = {M32n2, M43n2}) // MatrixForm;
(An2 = (1 / (e * i)) * {
      {-(b2 / 3 + b3 / 3), -b3 / 6},
      {-b3 / 6, -(b3 / 3 + b2 / 3)}
     }) // MatrixForm;
(S4n2 = {M32n2, M43n2} /. Solve[T + An2.Mn2 == T0n2, Mn2]) // MatrixForm
```

```
( -0.253208  0.100393 )
```

- **Determinación del giro de los ángulos (θ2, θ3, θ4, θ5) al mover el nodo 2**

In[331]:=
```
M32n2 = S4n2[[1, 1]];
M43n2 = S4n2[[1, 2]];
θ2n2 = L / Pi * (-1 / 2 * M32n2 / (3 * e * i / b2) + ArcTan[1 / (b1 * b2)]);
θ3n2 = L / Pi * (M32n2 / (3 * e * i / b2) + ArcTan[1 / (b1 * b2)]);
θ4n2 = L / Pi * (-M43n2 / (3 * e * i / b2));
θ5n2 = L / Pi * (1 / 2 * M43n2 / (3 * e * i / b2));
(θn2 = {θ2n2, θ2n2, θ3n2, θ4n2, θ5n2, θ5n2}) // MatrixForm
```

```
9.00661
9.00661
3.09098
-1.56364
0.781819
0.781819
```

- **Matriz de giro de los nodos al mover los nodos 1, 2 y 3**

In[338]:=
```
(GA = -Transpose[{θn1, θn2, θn3}]) // MatrixForm
```

8.6944	-9.00661	1.6559
8.6944	-9.00661	1.6559
3.71539	-3.09098	-0.217358
-0.939258	1.56364	-0.21741
0.469629	-0.781819	1.65589
0.469629	-0.781819	1.65589

3.Obtención de la deformación de la sección

- **Obtención de la función Viq(s) que representa el desplazamiento nodal según el eje local s**

In[339]:=
```
Viq[1] = {1 / b1, 0, 0, 0, 0};
Viq[2] = {-1 / b1, 1 / b2, 0, 0, 0};
Viq[3] = {0, -1 / b2, 1 / b3, 0, 0};
Viq[3] = {0, -1 / b2, 1 / b3, 0, 0};
Viq[4] = Reverse[-Viq[3]];
Viq[5] = Reverse[-Viq[2]];
Viq[6] = Reverse[-Viq[1]];
V[i_, q_] := Viq[i][[q]];
```

- **Definición de funciones necesarias para realizar los cálculos sucesivos**

In[347]:=
```
b = {b1, b2, b3, b4 = b2, b5 = b1};
s[q_] := Sum[(b[[j]]), {j, 1, q-1}];
Table[b[[i]], {i, 1, 5}];
Table[s[i], {i, 1, 6}];
ξ[q_] := (s - s[q]) / b[[q]];
φ1 = 1;
φ2[q_] := 2 * ξ[q] - 1;
φ3[q_] := -ξ[q] + 3 / 2 * ξ[q]^2 - 1 / 2 * ξ[q]^3;
φ4[q_] := -1 / 2 * ξ[q] + 1 / 2 * ξ[q]^3;
Table[ξ[i], {i, 1, 5}];
Kq = e * t^3 / (12 * (1 - v^2));
```

- **Obtención de la función Wiq(s) que representa el desplazamiento nodal según el eje local z**

In[358]:=
```
W[i_, q_] := Wq[i][[q]] * φ1 + b[[q]] / 2 * φ2[q] * θq[i][[q]] * Pi / L -
  b[[q]]^2 / (3 * Kq) * (Mq[i][[2 * q - 1]] * φ3[q] - Mq[i][[2 * q]] * φ4[q])
```

- **Obtención de la función W1q(s) que representa el desplazamiento nodal según el eje local z al mover el nodo 1**

Wq[1 = {W11,W12,W13,W14,W15}

θq[1] = {θ11,θ12,θ13,θ14,θ15}

Mq[1] = {M12n1,M21n1,M23n1,M32n1,M34n1,M43n1,M45n1,M54n1,M56n1,M65n1}

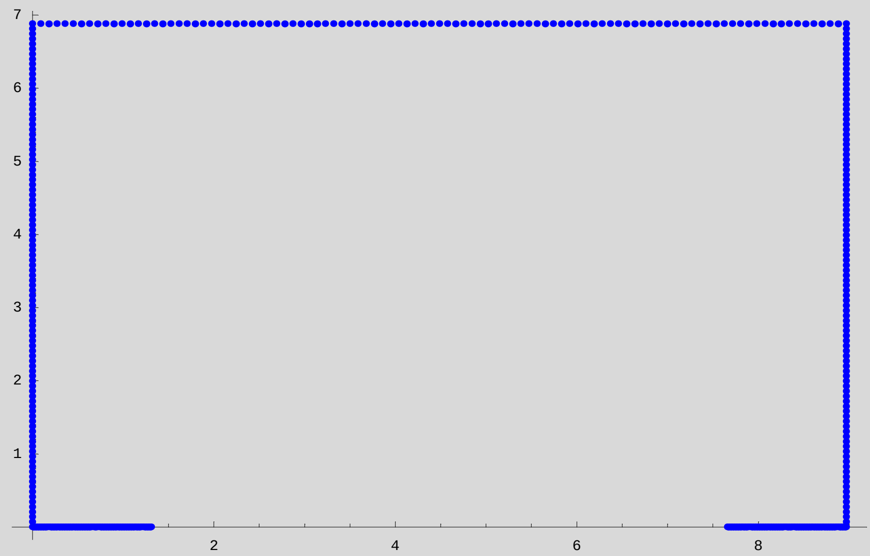
In[359]:=
```
Wq[1] = {-(P1n1'[[2]] + P2n1'[[2]]) / 2, -(P2n1'[[1]] + P3n1'[[1]]) / 2, 0, 0, 0};
θq[1] = {-L / Pi * ArcTan[1 / (b1 * b2)], -L / Pi * ArcTan[1 / (b1 * b2)], 0, 0, 0};
Mq[1] = {0, 0, 0, M32n1, -M32n1, M43n1, -M43n1, 0, 0, 0};
```

P[i,q]= coordenadas del elemento q al mover el nudo i

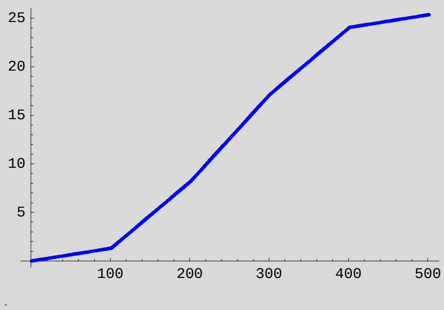
In[362]:=
```
ψ[1] = Pi;
ψ[2] = Pi / 2;
ψ[3] = 0;
ψ[4] = -Pi / 2;
ψ[5] = -Pi;
(Pass[q_] = {{Cos[ψ[q]], -Sin[ψ[q]]}, {Sin[ψ[q]], Cos[ψ[q]]}}) // MatrixForm;
Pass[1] // MatrixForm;
Pt[1] = Table[{b1 - k * b1 / n, 0}, {k, 0, n}];
Pt[2] = Table[{0, k * b2 / n}, {k, 1, n}];
Pt[3] = Table[{k * b3 / n, b2}, {k, 1, n}];
Pt[4] = Table[{b3, b2 - k * b2 / n}, {k, 1, n}];
Pt[5] = Table[{b3 - k * b1 / n, 0}, {k, 1, n}];
Pts = Join[Pt[1], Pt[2], Pt[3], Pt[4], Pt[5]];
Length[Pts];
```

■ **Representación discreta de la geometría inicial**

In[376]:=
```
Fig = ListPlot[Pts, PlotStyle → {RGBColor[0, 0, 1], AbsoluteThickness[3]}, Axes → True,
   ImageSize → 600, PlotJoined → False];
```

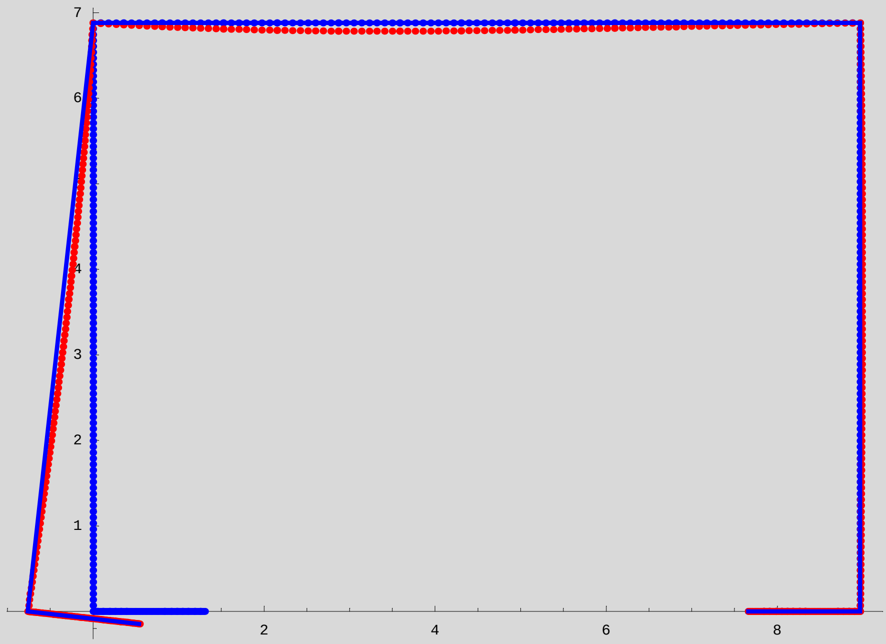


In[377]:=
```
CoordPoint[p_] := Pts[[p]]
CoordPoint[1];
ss[p_] =
  If[1 <= p ≤ n + 1, (p - 1) * b1 / n,
   If[n + 2 <= p ≤ 2 * n + 1, b1 + (p - n - 1) * b2 / n,
    If[2 * n + 2 <= p ≤ 3 * n + 1, b1 + b2 + (p - 2 n - 1) * b3 / n,
     If[3 * n + 2 <= p ≤ 4 * n + 1, b1 + b2 + b3 + (p - 3 n - 1) * b2 / n,
      If[4 * n + 2 <= p ≤ 5 * n + 1, b1 + b2 + b3 + b2 + (p - 4 n - 1) * b1 / n, "ssp pas compris"]]]]];
tab1 = Table[ss[i], {i, 1, 5 * n + 1}];
ListPlot[tab1, PlotStyle → {RGBColor[0, 0, 1], AbsoluteThickness[3]}, Axes → True,
  ImageSize → 300, PlotJoined → False];
Length[tab1];
El[p_] =
  If[1 <= p ≤ n + 1, 1,
   If[n + 2 <= p ≤ 2 * n + 1, 2,
    If[2 * n + 2 <= p ≤ 3 * n + 1, 3,
     If[3 * n + 2 <= p ≤ 4 * n + 1, 4,
      If[4 * n + 2 <= p ≤ 5 * n + 1, 5, "segment non trouvé"]]]]];
CoordPointdef[i_, p_] := (CoordPoint[p] + Pass[El[p]].{V[i, El[p]], ReplaceAll[W[i, El[p]], s → ss[p]]});
L1 = ListPlot[Table[CoordPointdef[1, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0], AbsoluteThickness[3]},
   Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
```



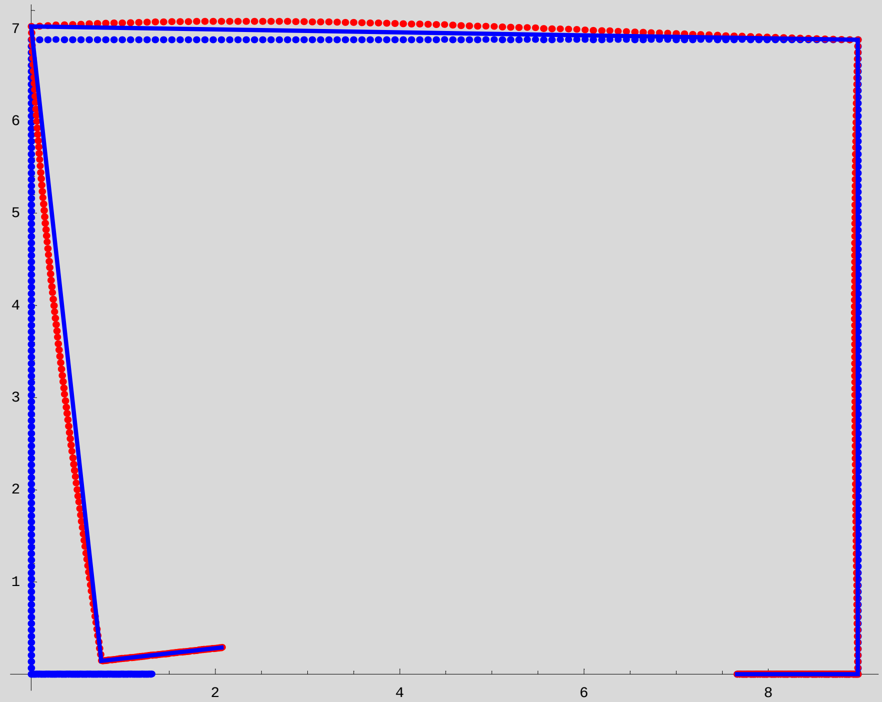
■ **Deformada de la sección al mover el nodo 1**

In[386]:=
```
Show[Fig, L1, CGn1, ImageSize → 600, AspectRatio → Automatic];
```



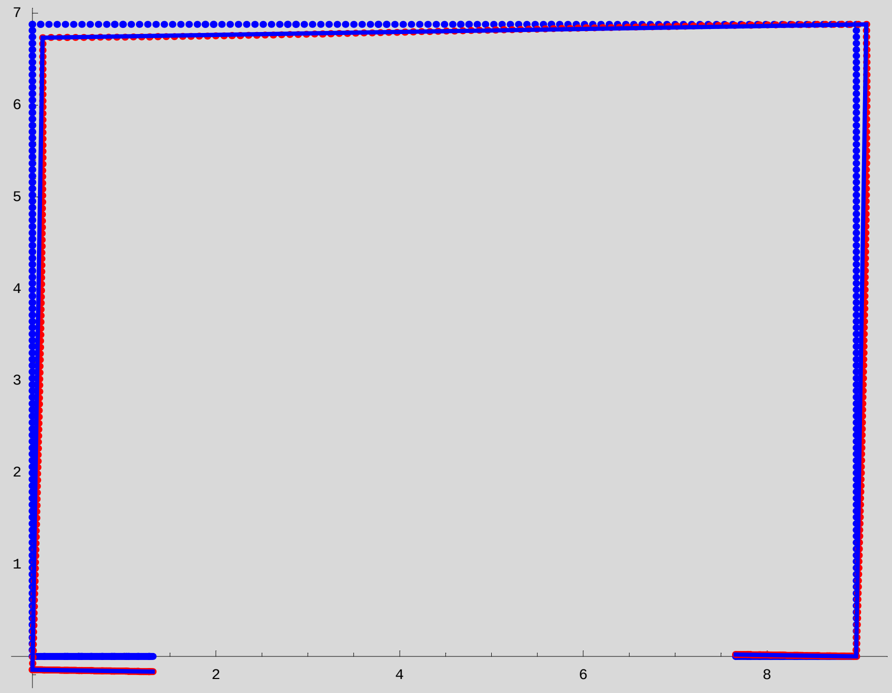
■ **Deformada de la sección al mover el nodo 2**

In[387]:=
```
Wq[2] = {-(P1n2'[[2]] + P2n2'[[2]]) / 2, -(P2n2'[[1]] + P3n2'[[1]]) / 2, (P3n2'[[2]] - P4n2'[[2]]) / 2, 0, 0};
N[Wq[2]];
θq[2] = {L / Pi * ArcTan[1 / (b1 * b2)], L / Pi * ArcTan[1 / (b1 * b2)], -L / Pi * ArcTan[1 / (b3 * b2)], 0, 0};
Mq[2] = {0, 0, 0, M32n2, -M32n2, M43n2, -M43n2, 0, 0, 0};
L2 = ListPlot[Table[CoordPointdef[2, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0], AbsoluteThickness[3]},
   Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
Show[Fig, L2, CGn2, ImageSize → 600, AspectRatio → Automatic];
```



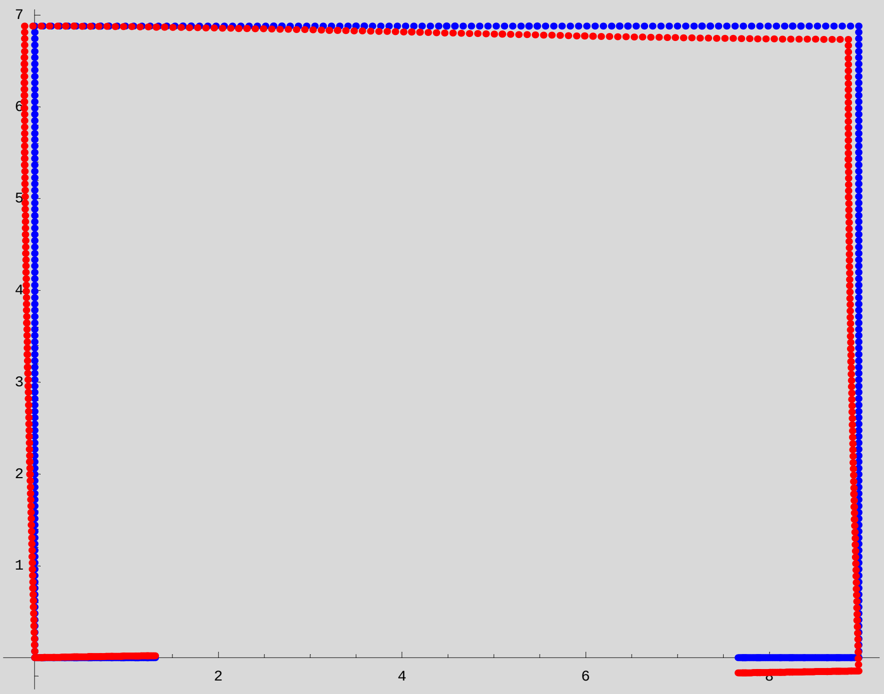
■ **Deformada de la sección al mover el nodo 3**

In[393]:=
```
Wq[3] = {-(P1n3'[[2]] + P2n3'[[2]]) / 2, -(P3n3'[[1]] + P2n3'[[1]]) / 2, -(P4n3'[[2]] - P3n3'[[2]]) / 2,
   (P4n3'[[1]] - P5n3'[[1]]) / 2, -(P6n3'[[2]] - P5n3'[[2]]) / 2};
θq[3] = {-L / Pi * ArcTan[1 / (b2 * b3)], -L / Pi * ArcTan[1 / (b2 * b3)], L / Pi * ArcTan[1 / (b2 * b3)], -L / Pi *
    ArcTan[1 / (b2 * b3)], -L / Pi * ArcTan[1 / (b2 * b3)]};
Mq[3] = {0, 0, 0, M32n3, -M32n3, M43n3, -M43n3, 0, 0, 0};
L3 = ListPlot[Table[CoordPointdef[3, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0], AbsoluteThickness[3]},
   Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
Show[Fig, L3, CGn3, ImageSize → 600, AspectRatio → Automatic];
```



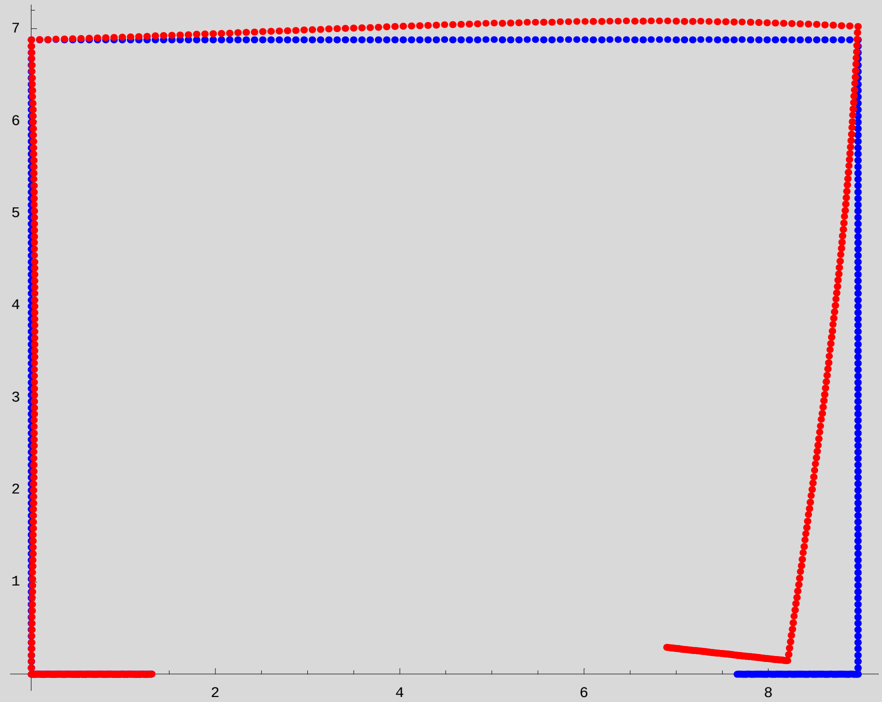
■ **Deformada de la sección al mover el nodo 4 (se utilizan las propiedades de simetría del perfil)**

In[398]:=
```
Wq[4] = Reverse[Wq[3]];
θq[4] = Reverse[-θq[3]];
Mq[4] = Reverse[-Mq[3]];
L4 = ListPlot[Table[CoordPointdef[4, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0],
     AbsoluteThickness[3]}, Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
Show[Fig, L4, ImageSize → 600, AspectRatio → Automatic];
```



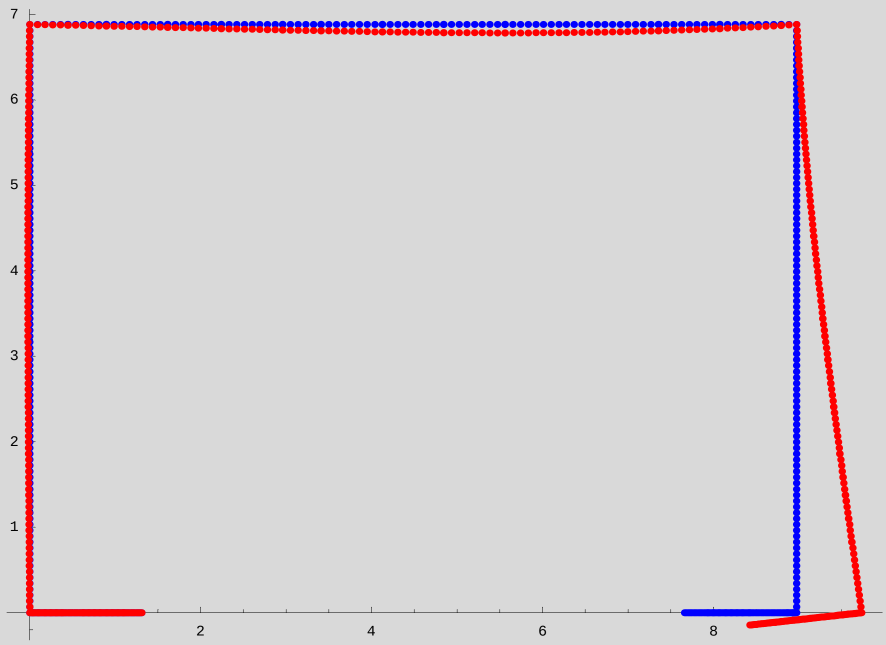
■ **Deformada de la sección al mover el nodo 5 (se utilizan las propiedades de simetría del perfil)**

In[403]:=
```
Wq[5] = Reverse[Wq[2]];
θq[5] = Reverse[-θq[2]];
Mq[5] = Reverse[-Mq[2]];
L5 = ListPlot[Table[CoordPointdef[5, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0],
     AbsoluteThickness[3]}, Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
Show[Fig, L5, ImageSize → 600, AspectRatio → Automatic];
```



■ **Deformada de la sección al mover el nodo 6 (se utilizan las propiedades de simetría del perfil)**

In[408]:=
```
Wq[6] = Reverse[Wq[1]];
θq[6] = Reverse[-θq[1]];
Mq[6] = Reverse[-Mq[1]];
L6 = ListPlot[Table[CoordPointdef[6, p], {p, 1, 5 * n + 1}], PlotStyle → {RGBColor[1, 0, 0],
     AbsoluteThickness[3]}, Axes → True, ImageSize → 1, AspectRatio → Automatic, PlotJoined → False];
Show[Fig, L6, ImageSize → 600, AspectRatio → Automatic];
```



4. Determinación de las matrices de la T.V.G. no diagonalizadas

Propiedades del material utilizado

In[413]:=
```
{e, t, v}
```

Out[413]=
```
{20000, 0.167, 0.3}
```

Calculo de la matriz Cik1

In[414]:=
```
(Cik1 = t * {{b1 / 3, b1 / 6, 0, 0, 0, 0}, {b1 / 6, b2 / 3 + b1 / 3, b2 / 6, 0, 0, 0},
    {0, b2 / 6, b2 / 3 + b3 / 3, b3 / 6, 0, 0}, {0, 0, b3 / 6, b3 / 3 + b4 / 3, b4 / 6, 0},
    {0, 0, 0, b4 / 6, b4 / 3 + b5 / 3, b5 / 6}, {0, 0, 0, 0, b5 / 6, b5 / 3}}) // MatrixForm
```

0.0729233	0.0364617	0	0	0	0
0.0364617	0.45591	0.191493	0	0	0
0	0.191493	0.882317	0.249665	0	0
0	0	0.249665	0.882317	0.191493	0
0	0	0	0.191493	0.45591	0.0364617
0	0	0	0	0.0364617	0.0729233

Calculo de la matriz Cik2

In[415]:=

$$\text{Cik2}[i\_, j\_] := (t^3/12) \sum_{q=1}^{5} \left(\int_{s[q]}^{s[q+1]} W[i, q] * W[j, q] \, ds \right);$$

```
(Cik2 = Table[Table[Cik2[i, j], {j, 1, 6}], {i, 1, 6}]) // MatrixForm
```

0.000451757	−0.000467416	−0.0000184256	0.0000541549	−0.000056436	0.0000363651
−0.000467416	0.00052002	−0.0000234042	−0.0000598423	0.0000870786	−0.000056436
−0.0000184256	−0.0000234042	0.0000755028	−0.0000279856	−0.0000598423	0.0000541549
0.0000541549	−0.0000598423	−0.0000279856	0.0000755028	−0.0000234042	−0.0000184256
−0.000056436	0.0000870786	−0.0000598423	−0.0000234042	0.00052002	−0.000467416
0.0000363651	−0.000056436	0.0000541549	−0.0000184256	−0.000467416	0.000451757

Calculo de la matriz Cik=Cik1+Cik2

In[417]:=
```
(Cik = Cik1 + Cik2) // MatrixForm
```

0.0733751	0.0359943	−0.0000184256	0.0000541549	−0.000056436	0.0000363651
0.0359943	0.45643	0.19147	−0.0000598423	0.0000870786	−0.000056436
−0.0000184256	0.19147	0.882392	0.249637	−0.0000598423	0.0000541549
0.0000541549	−0.0000598423	0.249637	0.882392	0.19147	−0.0000184256
−0.000056436	0.0000870786	−0.0000598423	0.19147	0.45643	0.0359943
0.0000363651	−0.000056436	0.0000541549	−0.0000184256	0.0359943	0.0733751

Calculo de la matriz Bik

In[452]:=

$$\left(\text{Bik} = \text{Table}\left[\text{Table}\left[e * t^3 / (12 * (1 - v^2)) \sum_{q=1}^{5} \left(\int_{s[q]}^{s[q+1]} \partial_s \partial_s W[i, q] * \partial_s \partial_s W[j, q] \, ds \right), \{j, 1, 6\}\right], \{i, 1, 6\}\right]\right) // \text{MatrixForm}$$

0.0258789	−0.0307471	0.00973645	−0.00973625	0.0121908	−0.00732283
−0.0307471	0.0370428	−0.0125915	0.0125913	−0.0184862	0.0121908
0.00973645	−0.0125915	0.00571034	−0.00571034	0.0125913	−0.00973625
−0.00973625	0.0125913	−0.00571034	0.00571034	−0.0125915	0.00973645
0.0121908	−0.0184862	0.0125913	−0.0125915	0.0370428	−0.0307471
−0.00732283	0.0121908	−0.00973625	0.00973645	−0.0307471	0.0258789

- **Calculo de la matriz Dik**

In[419]:=

$$\left(\texttt{Dik} = \texttt{Table}\Big[\texttt{Table}\Big[\; t^3/3 \sum_{q=1}^{5}\left(\int_{s[q]}^{s[q+1]} \partial_s W[i,q] * \partial_s W[j,q]\, ds\right) + (2*v*e*t^3/(12*(1-v^2)))/(e/(2*(1+v))) \sum_{q=1}^{5}\left(\int_{s[q]}^{s[q+1]} (W[i,q]*\partial_s\partial_s W[j,q] + W[j,q]*\partial_s\partial_s W[i,q])\, ds\right), \{j,1,6\}\Big], \{i,1,6\}\Big]\right) \text{ // MatrixForm}$$

0.000186797	−0.000183646	0.0000172132	−0.0000249832	0.0000125961	-7.97658×10^{-6}
−0.000183646	0.000182883	−0.0000174129	0.000025183	−0.0000196031	0.0000125961
0.0000172132	−0.0000174129	1.58025×10^{-6}	-1.58042×10^{-6}	0.000025183	−0.0000249832
−0.0000249832	0.000025183	-1.58042×10^{-6}	1.58025×10^{-6}	−0.0000174129	0.0000172132
0.0000125961	−0.0000196031	0.000025183	−0.0000174129	0.000182883	−0.000183646
-7.97658×10^{-6}	0.0000125961	−0.0000249832	0.0000172132	−0.000183646	0.000186797

5. Determinación de la matriz de cambio de base

- **Determinación de los vectores propios del sistema (Bik-λ.Cic)**

In[420]:=

```
(Vp = Eigenvectors[{Bik, Cik}]) // MatrixForm
```

0.666562	−0.212525	0.102609	−0.102609	0.212525	−0.666562
−0.687421	0.163373	−0.0276029	−0.0276029	0.163373	−0.687421
0.604825	0.425653	−0.0931041	0.122583	−0.36498	−0.544152
−0.271715	−0.426273	−0.337623	0.169721	−0.465727	−0.620295
−0.531607	−0.321513	0.621641	−0.139084	−0.410142	−0.200063
−0.0980455	0.143252	−0.0502517	−0.960313	−0.0311421	0.210168

- **Coordenadas sectoriales (calculadas con el programa CUFSM)**

In[421]:=

```
sect1 = {1, 1, 1, 1, 1, 1};
sect2 = {31.75, 44.85, 44.85, -44.85, -44.85, -31.75} / 10;
sect3 = {43.017, 43.017, -25.783, -25.783, 43.017, 43.017} / 10;
sect4 = {-2884.71, -1528.65, 1557.02, -1557.02, 1528.65, 2884.71} / 100;
```

- **Obtención de la matriz de cambio de base**

In[425]:=

```
(Vp1 = {Take[Vp, {1}][[1]] / Norm[Take[Vp, {1}]] * 1 / Vp[[1, 1]], Take[Vp, {2}][[1]] /
    Norm[Take[Vp, {2}]] / Vp[[2, 1]], sect1, sect2, sect3, sect4}) // MatrixForm
```

1.	−0.318838	0.153937	−0.153937	0.318838	−1.
1.	−0.237661	0.0401542	0.0401542	−0.237661	1.
1	1	1	1	1	1
3.175	4.485	4.485	−4.485	−4.485	−3.175
4.3017	4.3017	−2.5783	−2.5783	4.3017	4.3017
−28.8471	−15.2865	15.5702	−15.5702	15.2865	28.8471

6. Determinación de las matrices de la T.V.G. diagonalizadas

- **Calculo de la matriz Cik diagonalizada**

In[426]:=

```
Vp1.Cik.Transpose[Vp1] // MatrixForm
```

0.185823	-5.34613×10^{-17}	-3.05311×10^{-16}	0.000649702	-1.74302×10^{-15}	0.00666356
-5.56331×10^{-17}	0.160579	-3.08426×10^{-7}	2.1515×10^{-15}	−0.0000112263	-1.04608×10^{-14}
-2.848×10^{-16}	-3.08426×10^{-7}	4.23345	-4.44572×10^{-15}	0.000015866	3.19139×10^{-14}
0.000649702	2.18488×10^{-15}	-4.71845×10^{-15}	62.7535	-1.72419×10^{-14}	0.036906
-1.78012×10^{-15}	−0.0000112263	0.000015866	-1.74759×10^{-14}	28.8296	8.28877×10^{-14}
0.00666356	-1.07104×10^{-14}	3.10862×10^{-14}	0.036906	9.09446×10^{-14}	523.504

- **Calculo de algunas propiedades del perfil que se obtiene en la matriz Cik diagonalizada**

Calculo del area del perfil:

In[427]:=
$$\text{Area} = t * \sum_{q=1}^{5} b[[q]]$$

Out[427]=
```
4.23345
```

Calculo de la inercia a flexion segun el eje debil:

In[428]:=
```
Zg = Zg /. Solve[Zg * (2 * t * b1 + t * b3 + 2 * t * (b2 - 2 * t)) == 2 * b1 * t * t / 2 + 2 * (b2 - 2 * t) * b2 / 2 * t +
      t * b3 * (b2 - t / 2), Zg][[1]];
IdebilEexacta = (2 * (b1 * t^3 / 12 + (Zg - t / 2)^2 * b1 * t) + 2 * (t * (b2 - 2 * t)^3 / 12 + (b2 / 2 - Zg)^2 *
        (b2 - 2 * t) * t) + b3 * t^3 / 12 + (b2 - t / 2 - Zg)^2 * b3 * t) * e;
Idebilaprox = (2 * (b1 * t^3 / 12 + (Zg)^2 * b1 * t) + 2 * (t * (b2)^3 / 12 + (b2 / 2 - Zg)^2 * (b2) * t) +
   b3 * t^3 / 12 + (b2 - Zg)^2 * b3 * t)
```

Out[430]=
```
28.8296
```

Calculo de la inercia a flexion segun el eje fuerte:

In[431]:=
```
Yg = Yg /. Solve[Yg * (2 * t * b1 + t * b3 + 2 * t * (b2 - 2 * t)) == b1 * t * b1 / 2 + t * (b2 - 2 * t) * t / 2 + b3 * t * b3 / 2 +
      t * (b2 - 2 * t) * (b3 - t / 2) + b1 * t * (b3 - b1 / 2), Yg][[1]];
Ifortaprox = (t * b3^3 / 12 + 2 * (b2 * t^3 / 12 + b2 * t * (b3 / 2)^2 + t * b1^3 / 12 + b1 * t * (b3 / 2 - b1 / 2)^2))
```

Out[432]=
```
62.7535
```

- **Calculo de la matriz Bik diagonalizada**

In[433]:=
```
Vp1.Bik.Transpose[Vp1] // MatrixForm
```

0.14993	-5.36375×10^{-17}	-3.22659×10^{-16}	0.000524206	-1.50735×10^{-15}	0.00537643
-5.32191×10^{-17}	0.0568489	-1.0919×10^{-7}	6.96037×10^{-16}	-3.97438×10^{-6}	-3.51347×10^{-15}
-3.15195×10^{-16}	-1.0919×10^{-7}	2.10711×10^{-13}	6.46961×10^{-16}	7.63561×10^{-12}	9.1326×10^{-15}
0.000524206	7.07761×10^{-16}	6.46957×10^{-16}	1.8328×10^{-6}	6.39136×10^{-15}	0.0000187978
-1.47253×10^{-15}	-3.97438×10^{-6}	7.63561×10^{-12}	6.39138×10^{-15}	2.77854×10^{-10}	1.97399×10^{-14}
0.00537643	-3.54187×10^{-15}	9.13289×10^{-15}	0.0000187978	1.97406×10^{-14}	0.000192797

- **Calculo de la matriz Dik diagonalizada**

In[434]:=
```
Vp1.Dik.Transpose[Vp1] // MatrixForm
```

0.000715488	3.18828×10^{-17}	-1.46977×10^{-17}	-0.000218153	-7.10108×10^{-17}	-0.00529587
3.18384×10^{-17}	0.000537149	-6.68808×10^{-11}	-7.10689×10^{-17}	0.000132325	2.28139×10^{-16}
-1.47725×10^{-17}	-6.68808×10^{-11}	5.37324×10^{-17}	-4.86768×10^{-17}	-2.54162×10^{-10}	2.85445×10^{-16}
-0.000218153	-7.10412×10^{-17}	-4.86739×10^{-17}	-1.53411×10^{-6}	-2.28355×10^{-16}	-0.0000285054
-7.13539×10^{-17}	0.000132325	-2.54162×10^{-10}	-2.28389×10^{-16}	-1.85023×10^{-8}	1.53706×10^{-15}
-0.00529587	2.2744×10^{-16}	2.85145×10^{-16}	-0.0000285054	1.53542×10^{-15}	0.0389223

7. Determinación de los desplazamientos nodales

- **Calculo de la matriz [$\overline{\overline{v}}$]**

In[435]:=
```
Vdesp1={{P1n1'-P1}[[1,1]],{P2n1'-P2}[[1,1]],{P3n1'-P3}[[1,1]],{P4n1'-P4}[[1,1]],{P5n1'-P5}[[1,1]],{P6n1'-P6}[[1,1]]};
Vdesp2={{P1n2'-P1}[[1,1]],{P2n2'-P2}[[1,1]],{P3n2'-P3}[[1,1]],{P4n2'-P4}[[1,1]],{P5n2'-P5}[[1,1]],{P6n2'-P6}[[1,1]]};
Vdesp3={{P1n3'-P1}[[1,1]],{P2n3'-P2}[[1,1]],{P3n3'-P3}[[1,1]],{P4n3'-P4}[[1,1]],{P5n3'-P5}[[1,1]],{P6n3'-P6}[[1,1]]};
Vdesp4=-Reverse[Vdesp3];
Vdesp5=-Reverse[Vdesp2];
Vdesp6=-Reverse[Vdesp1];
(DespX=Transpose[{Vdesp1,Vdesp2,Vdesp3,Vdesp4,Vdesp5,Vdesp6}])//MatrixForm
```

-0.771349	0.755369	-0.000171947	-0.000171947	0.	0.
-0.763359	0.763359	0	0.	0.	0.
0	0	0.111483	-0.111483	0.	0.
0.	0.	0.111483	-0.111483	0	0
0.	0.	0.	0	-0.763359	0.763359
0.	0.	0.000171947	0.000171947	-0.755369	0.771349

▪ Calculo de la matriz [$\overline{\overline{w}}$]

In[442]:=

```
Wdesp1={{P1n1'-P1}[[1,2]],{P2n1'-P2}[[1,2]],{P3n1'-P3}[[1,2]],{P4n1'-P4}[[1,2]],{P5n1'-P5}[[1,2]],{P6n1'-P6}[[1,2]]};
Wdesp2={{P1n2'-P1}[[1,2]],{P2n2'-P2}[[1,2]],{P3n2'-P3}[[1,2]],{P4n2'-P4}[[1,2]],{P5n2'-P5}[[1,2]],{P6n2'-P6}[[1,2]]};
Wdesp3={{P1n3'-P1}[[1,2]],{P2n3'-P2}[[1,2]],{P3n3'-P3}[[1,2]],{P4n3'-P4}[[1,2]],{P5n3'-P5}[[1,2]],{P6n3'-P6}[[1,2]]};
Wdesp4=Reverse[Wdesp3];
Wdesp5=Reverse[Wdesp2];
Wdesp6=Reverse[Wdesp1];
(DespY=Transpose[{Wdesp1,Wdesp2,Wdesp3,Wdesp4,Wdesp5,Wdesp6}])//MatrixForm
```

$$\begin{pmatrix} -0.144462 & 0.289811 & -0.166573 & 0.0212243 & 0 & 0 \\ 0 & 0.145349 & -0.145349 & 0 & 0 & 0 \\ 0. & 0.145349 & -0.145349 & 0. & 0. & 0. \\ 0. & 0. & 0. & -0.145349 & 0.145349 & 0. \\ 0 & 0 & 0 & -0.145349 & 0.145349 & 0 \\ 0 & 0 & 0.0212243 & -0.166573 & 0.289811 & -0.144462 \end{pmatrix}$$

▪ Calculo de la matriz de vectores propios [$\tilde{u}_i$]

In[449]:=

```
(Vp1desp=Transpose[{sect1,-sect2,sect3, sect4,-Take[Vp,{1}][[1]]/Norm[Take[Vp,{1}]]*1/Vp[[1,1]],
-Take[Vp,{2}][[1]]/Norm[Take[Vp,{2}]]/Vp[[2,1]]}])//MatrixForm
```

$$\begin{pmatrix} 1 & -3.175 & 4.3017 & -28.8471 & -1. & -1. \\ 1 & -4.485 & 4.3017 & -15.2865 & 0.318838 & 0.237661 \\ 1 & -4.485 & -2.5783 & 15.5702 & -0.153937 & -0.0401542 \\ 1 & 4.485 & -2.5783 & -15.5702 & 0.153937 & -0.0401542 \\ 1 & 4.485 & 4.3017 & 15.2865 & -0.318838 & 0.237661 \\ 1 & 3.175 & 4.3017 & 28.8471 & 1. & -1. \end{pmatrix}$$

▪ Calculo de la matriz de desplazamiento horizontal nodal para cada modo i [$\overline{\overline{v}}_i^r$]

In[450]:=

```
DespX.Vp1desp//MatrixForm
```

$$\begin{pmatrix} -0.0163234 & -0.938798 & -0.0678526 & 10.7042 & 1.01219 & 0.950884 \\ 0. & -1. & 0. & 10.3516 & 1.00675 & 0.94478 \\ -2.35922\times10^{-16} & -1. & 6.08292\times10^{-16} & 3.47162 & -0.0343227 & 4.82123\times10^{-16} \\ 2.35922\times10^{-16} & -1. & -6.08292\times10^{-16} & 3.47162 & -0.0343227 & 4.63176\times10^{-16} \\ 0. & -1. & 0. & 10.3516 & 1.00675 & -0.94478 \\ 0.0163234 & -0.938798 & 0.0678526 & 10.7042 & 1.01219 & -0.950884 \end{pmatrix}$$

▪ Calculo de la matriz de desplazamiento vertical nodal para cada modo i [$\overline{\overline{w}}_i^r$]

In[451]:=

```
DespY.Vp1desp//MatrixForm
```

$$\begin{pmatrix} -1.73472\times10^{-17} & 0.00113631 & 1. & -3.18692 & 0.265774 & 0.219176 \\ 0. & 0. & 1. & -4.48499 & 0.0687173 & 0.0403802 \\ 0. & 0. & 1. & -4.48499 & 0.0687173 & 0.0403802 \\ 0. & 0. & 1. & 4.48499 & -0.0687173 & 0.0403802 \\ 0. & 0. & 1. & 4.48499 & -0.0687173 & 0.0403802 \\ -1.73472\times10^{-17} & -0.00113631 & 1. & 3.18692 & -0.265774 & 0.219176 \end{pmatrix}$$