```
(* APPENDIX 4 *)
(* Select all and copy of this paper,
and paste on the MATHEMATICA in TEXT type,
then MATHEMATICA program will be run. *)

(* A rolling position of the Two-Circle-Roller by Moving-Frame *)

Off[General::spell]

r = 1.0;
(* "r" is a radius of the T.C.R.
     and any value is permitted, 1.0 is an example. *)

α = (Pi/2 + ArcSin[1/(Sqrt[2] + 1)])*0.7;
(* the "α" is a Moving-Frame parameter, and it can be changed in between
    -(Pi/2 + ArcSin[1/(Sqrt[2] + 1)]) and
    +(Pi/2 + ArcSin[1/(Sqrt[2] + 1)]),
  however both ends are singular points *)

(* the "θ[t]" is a inclination angle of a tangential line
    of the curve Cad based on (36) *)

a = Sqrt[2] + 1;
b = Sqrt[2] - 1;
k = -1;
ϕ[α_] := ArcSin[Sqrt[b]*Tan[α/2]]

θ[α_] := -2*Sqrt[2]*
    (EllipticF[φ[α], k]
        - EllipticPi[-a, φ[α], k]
        - EllipticPi[-b, φ[α], k])

(* A contact point between T.C.R.and curve Cad on the XY - plane *)
Xa = r* NIntegrate[Cos[θ[τ]], {τ, 0, α}];
Ya = r* NIntegrate[Sin[θ[τ]], {τ, 0, α}];

(* A Moving - Frame of the T.C.R. in vector representation *)

c = Cos[α];
s = Sin[α];
```

```
{X, Y, Z} = {Xa, Ya, 0} + (r *Sqrt[2]/2)*{s*Cos[θ[α]] -
           Sin[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1],
         s*Sin[θ[α]] + Cos[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1], 1} +
{{c* Cos[θ[α]] + s*Sin[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1]/(Sqrt[2] + c),
  s *Cos[θ[α]] - c *Sin[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1]/(Sqrt[2] + c),
  Sin[θ[α]]/(Sqrt[2] + c)},
 {c* Sin[θ[α]] - s* Cos[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1]/(Sqrt[2] + c),
  s *Sin[θ[α]] + c *Cos[θ[α]]*Sqrt[(Sqrt[2] + c)^2 - 1]/(Sqrt[2] + c),
  -Cos[θ[α]]/(Sqrt[2] + c)}, {-s/(Sqrt[2] + c),  c/(Sqrt[2] + c),
   Sqrt[(Sqrt[2] + c)^2 - 1]/(Sqrt[2] + c)}}.{x, y, z} // N;

δo = ArcSin[1/a];
tend = Pi/2 + δo;
(* "tend" is a terminal value of the parameter "t" *)

(* The moved segment circle Ca based on (3). *)
caX[t_] := X /. {x -> r*Sin[t], y -> -r(Sqrt[2]/2 + Cos[t]), z -> 0.}
caY[t_] := Y /. {x -> r*Sin[t], y -> -r(Sqrt[2]/2 + Cos[t]), z -> 0.}
caZ[t_] := Z /. {x -> r*Sin[t], y -> -r(Sqrt[2]/2 + Cos[t]), z -> 0.}

(* The moved segment circle Ca based on (71). *)
caXτ[τ_] :=
  X /. {x -> r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
caYτ[τ_] :=
  Y /. {x -> r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
caZτ[τ_] :=
  Z /. {x -> r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
(* The moved segment circle Ca based on (71), but x <= 0 .*)
caXτm[τ_] :=
  X /. {x -> -r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
caYτm[τ_] :=
  Y /. {x -> -r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
caZτm[τ_] :=
  Z /. {x -> -r*Sqrt[(Sqrt[2] + Cos[τ])^2 - 1]/(1 + Sqrt[2]*Cos[τ]),
     y -> -r/(Sqrt[2] + 2*Cos[τ]), z -> 0}
```

```
(* The moved segment circle Cb based on (70). *)
cbXτ[τ_] := X /. {x -> 0, y -> r*(Sqrt[2] + 2*Cos[τ])/2, z -> r *Sin[τ]}
cbYτ[τ_] := Y /. {x -> 0, y -> r*(Sqrt[2] + 2*Cos[τ])/2, z -> r *Sin[τ]}
cbZτ[τ_] := Z /. {x -> 0, y -> r*(Sqrt[2] + 2*Cos[τ])/2, z -> r *Sin[τ]}

(* The moved segment circle Cb based on (6). *)
cbX[t_] :=
  X /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> -r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}
cbY[t_] :=
  Y /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> -r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}
cbZ[t_] :=
  Z /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> -r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}

(* The moved segment circle Cb based on (6), but z < 0 . *)
cbXp[t_] :=
  X /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}
cbYp[t_] :=
  Y /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}
cbZp[t_] :=
  Z /. {x -> 0, y -> r/(Sqrt[2] + 2*Cos[t]),
     z -> r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(1 + Sqrt[2]*Cos[t])}

movCadCbd =
  ParametricPlot3D[{{caX[t], caY[t], caZ[t], AbsoluteThickness[2]},
{cbXτ[t],
       cbYτ[t], cbZτ[t], AbsoluteThickness[2]}}, {t, -tend, tend},
   ViewPoint -> {3.929, -2.674, 2.231}, DisplayFunction -> Identity]

(* Moved x axis *)
xofx[x_] := X /. {x -> x, y -> 0, z -> 0.}
yofx[x_] := Y /. {x -> x, y -> 0, z -> 0.}
zofx[x_] := Z /. {x -> x, y -> 0, z -> 0.}
xaxis = ParametricPlot3D[{xofx[x], yofx[x], zofx[x],
     AbsoluteThickness[1.0]}, {x, -r*(Sqrt[2]/2)*1.5, r*(Sqrt[2]/2)*1.5},
   DisplayFunction -> Identity]
```

```
(* Moved y axis *)
xofy[y_] := X /. {x -> 0, y -> y, z -> 0.}
yofy[y_] := Y /. {x -> 0, y -> y, z -> 0.}
zofy[y_] := Z /. {x -> 0, y -> y, z -> 0.}
yaxis = ParametricPlot3D[{xofy[y], yofy[y], zofy[y],
      AbsoluteThickness[1.0]}, {y, -r*(1 + Sqrt[2]/2), r*(1 + Sqrt[2]/2)},
    DisplayFunction -> Identity]

(* Moved z axis *)
xofz[z_] := X /. {x -> 0, y -> 0, z -> z}
yofz[z_] := Y /. {x -> 0, y -> 0, z -> z}
zofz[z_] := Z /. {x -> 0, y -> 0, z -> z}
zaxis = ParametricPlot3D[{xofz[z], yofz[z], zofz[z],
      AbsoluteThickness[1.0]}, {z, -r*(Sqrt[2]/2)*1.5, r*(Sqrt[2]/2)*1.5},
    DisplayFunction -> Identity]

(* Moved O - xyz coordinate axes *)
movOxyz = Show[xaxis, yaxis, zaxis]

(* Moved centerline of the Ca *)
xclCa[x_] := X /. {x -> x, y -> -Sqrt[2]/2, z -> 0.}
yclCa[x_] := Y /. {x -> x, y -> -Sqrt[2]/2, z -> 0.}
zclCa[x_] := Z /. {x -> x, y -> -Sqrt[2]/2, z -> 0.}
movcenterlineCa =
  ParametricPlot3D[{xclCa[x], yclCa[x], zclCa[x]}, {x, -r, r},
    DisplayFunction -> Identity]

(* Moved centerline of the Cb *)
xclCb[z_] := X /. {x -> 0, y -> Sqrt[2]/2, z -> z}
yclCb[z_] := Y /. {x -> 0, y -> Sqrt[2]/2, z -> z}
zclCb[z_] := Z /. {x -> 0, y -> Sqrt[2]/2, z -> z}
movcenterlineCb =
  ParametricPlot3D[{xclCb[z], yclCb[z], zclCb[z]}, {z, -r, r},
    DisplayFunction -> Identity]
(* Moved gravity center O *)
xO = X /. {x -> 0, y -> 0, z -> 0};
yO = Y /. {x -> 0, y -> 0, z -> 0};
zO = Z /. {x -> 0, y -> 0, z -> 0};
movO = Show[Graphics3D[{PointSize[0.015], Hue[0.0], Point[{xO, yO, zO}]}],
    DisplayFunction -> Identity]
```

```
(* Moved contact point Ad based on (37), (38) *)
movAd = Show[Graphics3D[{PointSize[0.015],
      Point[{Xa, Ya, 0}]}], DisplayFunction -> Identity]

(* Moved contact point Bd based on (39), (40).*)
Xb = Xa +
    r*(2 + Sqrt[2]*c)*
      Cos[θ[α] + ArcCos[s/Sqrt[2 + 2*Sqrt[2]*c]]]/Sqrt[1 + Sqrt[2]*c];
Yb = Ya +
    r*(2 + Sqrt[2]*c)*
      Sin[θ[α] + ArcCos[s/Sqrt[2 + 2*Sqrt[2]*c]]]/Sqrt[1 + Sqrt[2]*c];

movBd = Show[Graphics3D[{PointSize[0.015], Point[{Xb,
          Yb, 0}]}], DisplayFunction -> Identity]



(* A point Sd *)
Xsd = X /. {x -> r*s/(2 + Sqrt[2]*c), y -> -r*c/(2 + Sqrt[2]*c),
      z -> -r*Sqrt[(Sqrt[2] + c)^2 - 1]/(2 + Sqrt[2]*c)};

Ysd = Y /. {x -> r*s/(2 + Sqrt[2]*c), y -> -r*c/(2 + Sqrt[2]*c),
      z -> -r*Sqrt[(Sqrt[2] + c)^2 - 1]/(2 + Sqrt[2]*c)};

Zsd = Z /. {x -> r*s/(2 + Sqrt[2]*c), y -> -r*c/(2 + Sqrt[2]*c),
      z -> -r*Sqrt[(Sqrt[2] + c)^2 - 1]/(2 + Sqrt[2]*c)};

movSd = Show[
   Graphics3D[{PointSize[0.015], Hue[0.66], Point[{Xsd, Ysd, Zsd}]}],
   ViewPoint -> {4.766, -6.834, 3.405},
   DisplayFunction -> Identity]



(* A moved line AeBo *)
xl1[x_] := X /. {x -> x, y -> -Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
yl1[x_] := Y /. {x -> x, y -> -Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
zl1[x_] := Z /. {x -> x, y -> -Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
line1 = ParametricPlot3D[{xl1[x], yl1[x], zl1[x]}, {x, 0, r*Cos[δo]},
   DisplayFunction -> Identity]
```

```
(* A moved line - AeBo *)
xl2[x_] := X /. {x -> x, y -> Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
yl2[x_] := Y /. {x -> x, y -> Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
zl2[x_] := Z /. {x -> x, y -> Tan[Pi/2 - δo]x + r(1. + Sqrt[2]/2), z -> 0}
line2 = ParametricPlot3D[{xl2[x], yl2[x], zl2[x]}, {x, -r*Cos[δo], 0},
   DisplayFunction -> Identity]

(* A moved line AoBe *)
xl3[y_] := X /. {x -> 0, y -> y, z -> Tan[δo](y + r(1 + Sqrt[2]/2))}
yl3[y_] := Y /. {x -> 0, y -> y, z -> Tan[δo](y + r(1 + Sqrt[2]/2))}
zl3[y_] := Z /. {x -> 0, y -> y, z -> Tan[δo](y + r(1 + Sqrt[2]/2))}
line3 = ParametricPlot3D[{xl3[y], yl3[y], zl3[y]}, {y, -r(1 + Sqrt[2]/2),
     r(Sqrt[2]/2 - Sin[δo])},
   DisplayFunction -> Identity]

(* A moved line Ao - Be *)
xl4[y_] := X /. {x -> 0, y -> y, z -> -Tan[δo](y + r(1 + Sqrt[2]/2))}
yl4[y_] := Y /. {x -> 0, y -> y, z -> -Tan[δo](y + r(1 + Sqrt[2]/2))}
zl4[y_] := Z /. {x -> 0, y -> y, z -> -Tan[δo](y + r(1 + Sqrt[2]/2))}
line4 = ParametricPlot3D[{xl4[y], yl4[y], zl4[y]}, {y, -r(1 + Sqrt[2]/2),
     r(Sqrt[2]/2 - Sin[δo])},
   DisplayFunction -> Identity]

(*  The axes of cartesian coordinate Aod - XYZ.  *)
jX = Show[Graphics3D[Line[{{-2.5*r, 0, 0}, {2.5*r, 0, 0}}]],
   DisplayFunction -> Identity]
jY1 = Show[Graphics3D[Line[{{0, 0, 0}, {0, 0.5*r, 0}}]],
   DisplayFunction -> Identity]
jY2 = Show[Graphics3D[Line[{{0, 3.5*r, 0}, {0, 4.2*r, 0}}]],
   DisplayFunction -> Identity]
jZ = Show[Graphics3D[Line[{{0, 0, 0}, {0, 0, 2*r}}]],
   DisplayFunction -> Identity]
AodXYZ = Show[jX, jY1, jY2, jZ, DisplayFunction -> Identity]
```

(* Appendix 4. H.Ira : The Development of the Two Circle Roller in a Numerical Way *)

```
inc = (Pi/2)/9;
listmCat = Table[{caX[t], caY[t], caZ[t]},
     {t, -Pi/2, Pi/2, inc}];
listmCbtm = Table[{cbX[t], cbY[t], cbZ[t]},
     {t, -Pi/2, Pi/2, inc}];
listmCbtp = Table[{cbXp[t], cbYp[t], cbZp[t]},
     {t, -Pi/2, Pi/2, inc}];

listmCbτ = Table[{cbXτ[t], cbYτ[t], cbZτ[t]},
     {t, -Pi/2, Pi/2, inc}];
listmCaτp = Table[{caXτ[t], caYτ[t], caZτ[t]},
     {t, -Pi/2, Pi/2, inc}];
listmCaτm = Table[{caXτm[t], caYτm[t], caZτm[t]},
     {t, -Pi/2, Pi/2, inc}];
num2 = Length[listmCat];


(* The moved rulled surface Ω *)
Ω1 = Show[
   Graphics3D[Table[Line[{listmCat[[i]], listmCbtm[[i]]}], {i, 1, num2}]],
   DisplayFunction -> Identity]
Ω2 = Show[
   Graphics3D[Table[Line[{listmCat[[i]], listmCbtp[[i]]}], {i, 1, num2}]],
   DisplayFunction -> Identity]
Ω3 = Show[
   Graphics3D[Table[Line[{listmCbτ[[i]], listmCaτp[[i]]}], {i, 1, num2}]],
   DisplayFunction -> Identity]
Ω4 = Show[
   Graphics3D[Table[Line[{listmCbτ[[i]], listmCaτm[[i]]}], {i, 1, num2}]],
   DisplayFunction -> Identity]

movΩ = Show[Ω1, Ω2, Ω3, Ω4, DisplayFunction -> Identity,
   ViewPoint -> {7.969, -3.966, 1.327}]
```

```
(* Moved sphere G *)
(* Euler - angles *)
      (* Let vecx, vecy and vecz are position vector
   of a tip of the x, y and z axis
     on the Aod - XYZ coordinates *)
vecx = {X /. {x -> 1, y -> 0, z -> 0}, Y /. {x -> 1, y -> 0, z -> 0},
     Z /. {x -> 1, y -> 0, z -> 0}};
vecy = {X /. {x -> 0, y -> 1, z -> 0}, Y /. {x -> 0, y -> 1, z -> 0},
     Z /. {x -> 0, y -> 1, z -> 0}};
vecz = {X /. {x -> 0, y -> 0, z -> 1}, Y /. {x -> 0, y -> 0, z -> 1},
     Z /. {x -> 0, y -> 0, z -> 1}};

   (* Let vecO is a position - vector
     of a origine of the O - xyz coordinates system
     by the Aod - XYZ coordinate system. *)
  vecO = {xO, yO, zO};



      (* Let eex, eey and eez are unit vector
   of the x, y and z axis
     by the Aod - XYZ coordinate system. *)
eex = vecx - vecO;
eey = vecy - vecO;
eez = vecz - vecO;

(* Let eeX, eeY and eeZ are unit vector
   of the X, Y and Z axis
     by the Aod - XYZ coordinate system. *)
eeX = {1, 0, 0}; eeY = {0, 1, 0}; eeZ = {0, 0, 1};

(* A direction cosine of the x - y plane,
  described by the Aod - XYZ coordinates system *)
λxy = eez.eeX;
μxy = eez.eeY;
νxy = eez.eeZ;

(* The euler - angles (ψ, θ, φ), z - x - z system *)
eulerψ = ArcTan[-λxy/μxy];
eulerθ = ArcCos[νxy];
eulerφ = -ArcCos[eex.{Cos[eulerψ], Sin[eulerψ], 0}] ;
```

```
(* Moved sphere G *)
<< Graphics`Shapes`
g = Show[Graphics3D[Sphere[r/Sqrt[2], 25, 25]],
   Shading -> False, DisplayFunction -> Identity]
gg = Show[RotateShape[g, -eulerφ, -eulerθ, -eulerψ],
   DisplayFunction -> Identity]
movGw = WireFrame[Show[TranslateShape[gg, {xO, yO, zO}],
     DisplayFunction -> Identity]]



(* Moved Mc curve *)
xsd = r*Sin[t]/(2 + Sqrt[2]*Cos[t]);
ysd = -r*Cos[t]/(2 + Sqrt[2]*Cos[t]);
zsd = r*Sqrt[(Sqrt[2] + Cos[t])^2 - 1]/(2 + Sqrt[2]*Cos[t]);

xsdX1 = X /. {x -> xsd, y -> ysd, z -> zsd};
ysdY1 = Y /. {x -> xsd, y -> ysd, z -> zsd};
zsdZ1 = Z /. {x -> xsd, y -> ysd, z -> zsd};
xsdX2 = X /. {x -> xsd, y -> ysd, z -> -zsd};
ysdY2 = Y /. {x -> xsd, y -> ysd, z -> -zsd};
zsdZ2 = Z /. {x -> xsd, y -> ysd, z -> -zsd};
movMc = ParametricPlot3D[
   {{xsdX1, ysdY1, zsdZ1, {AbsoluteThickness[2], Hue[0.66]}},
 {xsdX2, ysdY2, zsdZ2, {AbsoluteThickness[2], Hue[0.66]}}},
   {t, -tend, tend}, AspectRatio -> Automatic,
   PlotPoints -> 200, DisplayFunction -> Identity]

(* The development of the Two - Circle - Roller
     with a numerical integration *)
j = 100;
(* "j" is a discrete rate of a numerical integration *)

(* the inclination angle θ[t]
   of a tangenntial line of the curve Cad based on (36) *)
ϕ[t_] := ArcSin[Sqrt[Sqrt[2] - 1]*Tan[t/2]]
θ[t_] := -2 *Sqrt[2]*
     (EllipticF[φ[t], k]
        - EllipticPi[-a, φ[t], k]
        - EllipticPi[-b, φ[t], k])
```

```
(* Sd(π/2)is a center of point - reflection,
  and let coordinates be(Xc[π/2], Yc[π/2]) *)

Xc[π/2] =
    r*(NIntegrate[Cos[θ[τ]], {τ, 0, Pi/2}] + Sqrt[1 + Sqrt[2]*Cos[t]]*
            (Cos[θ[t]]*cosβ[t] - Sin[θ[t]]*sinβ[t])) /. t -> Pi/2 // N;
Yc[π/2] =
    r*(NIntegrate[Sin[θ[τ]], {τ, 0, Pi/2}] + Sqrt[1 + Sqrt[2]*Cos[t]]*
            (Sin[θ[t]]*cosβ[t] + Cos[θ[t]]*sinβ[t])) /. t -> Pi/2 // N;

(*  The list of the curve Cad
    with a numerical integration (37)and(38),
  however 0 <= t <= (π/2) *)
 listXa1 = Table[r*NIntegrate[Cos[θ[t]],
         {t, 0, i*Pi/(2*j)}], {i, 0, j}];
listYa1 = Table[r* NIntegrate[Sin[θ[t]],
         {t, 0, i*Pi/(2*j)}], {i, 0, j}];
listZa1 = Table[0*i, {i, 0, j}];

(* The list of the curve Cbd
    with a numerical integration (39)a and(40)a,
  however 0 <= t <= (π/2)  *)

sinβ[t_] := Sqrt[((Sqrt[2] + Cos[t])^2 - 1)/(2 + 2*Sqrt[2]*Cos[t])]
cosβ[t_] := Sin[t]/Sqrt[2 + 2*Sqrt[2]*Cos[t]];

(* The "ab[t]" is an absolute value of the vector QA *)
ab[t_] := r*Sqrt[2]*(Sqrt[2] + Cos[t])/Sqrt[1 + Sqrt[2]*Cos[t]];

abcosθβ[t_] := ab[t]*
    (Cos[θ[t]]*cosβ[t] - Sin[θ[t]]*sinβ[t])
absinθβ[t_] := ab[t]*
    (Sin[θ[t]]*cosβ[t] + Cos[θ[t]]*sinβ[t])

listabcosθβ = Table[abcosθβ[t] /. t -> i*Pi/(2*j) // N, {i, 0, j}];
listabsinθβ = Table[absinθβ[t] /. t -> i*Pi/(2*j) // N, {i, 0, j}];

listXb1 = listXa1 + listabcosθβ;
listYb1 = listYa1 + listabsinθβ;
listZb1 = listZa1;
```

```
(* Cmad4, Cmad3, Cmad2, Cmad1, Cad1, Cad2, Cad3, Cad4
   are segments of curve Cad *)

listXa2 = Table[2*Xc[π/2] - listXb1[[i]], {i, 1, j + 1}];
listYa2 = Table[2*Yc[π/2] - listYb1[[i]], {i, 1, j + 1}];
listZa2 = listZa1;

listXa3 = Table[4*Xc[π/2] - listXa2[[i]], {i, 1, j + 1}];
listYa3 = listYa2;
listZa3 = listZa1;

listXa4 = Table[4*Xc[π/2] - listXa1[[i]], {i, 1, j + 1}];
listYa4 = listYa1;
listZa4 = listZa1;

listCad1 = Transpose[{listXa1, listYa1, listZa1}];
listCad2 = Transpose[{listXa2, listYa2, listZa1}] // Reverse;
listCad3 = Transpose[{listXa3, listYa3, listZa1}];
listCad4 = Transpose[{listXa4, listYa4, listZa1}] // Reverse;
listCad1m = Transpose[{-listXa1, listYa1, listZa1}] // Reverse;
listCad2m = Transpose[{-listXa2, listYa2, listZa1}];
listCad3m = Transpose[{-listXa3, listYa3, listZa1}] // Reverse;
listCad4m = Transpose[{-listXa4, listYa4, listZa1}];

listCad = Partition[Flatten[{listCad4m, listCad3m, listCad2m,
        listCad1m, listCad1, listCad2, listCad3, listCad4}], 3];


(* Invisible plotting of the curve Cad *)

<< Graphics`Graphics3D`

curveCad = ScatterPlot3D[listCad, PlotJoined -> True,
   AspectRatio -> Automatic, PlotStyle -> AbsoluteThickness[2],
   DisplayFunction -> Identity]
```

```
 (* Cmbd4, Cmbd3, Cmbd2, Cmbd1, Cbd1, Cbd2, Cbd3, Cbd4
   are segments of curve Cbd *)

listXb2 = Table[2*Xc[π/2] - listXa1[[i]], {i, 1, j + 1}];
listYb2 = Table[2*Yc[π/2] - listYa1[[i]], {i, 1, j + 1}];
listZb2 = listZb1;

listXb3 = Table[4*Xc[π/2] - listXb2[[i]], {i, 1, j + 1}];
listYb3 = listYb2;
listZb3 = listZb1;

listXb4 = Table[4*Xc[π/2] - listXb1[[i]], {i, 1, j + 1}];
listYb4 = listYb1;
listZb4 = listZb1;

listCbd1 = Transpose[{listXb1, listYb1, listZb1}];
listCbd2 = Transpose[{listXb2, listYb2, listZb1}] // Reverse;
listCbd3 = Transpose[{listXb3, listYb3, listZb1}];
listCbd4 = Transpose[{listXb4, listYb4, listZb1}] // Reverse;
listCbd1m = Transpose[{-listXb1, listYb1, listZb1}] // Reverse;
listCbd2m = Transpose[{-listXb2, listYb2, listZb1}];
listCbd3m = Transpose[{-listXb3, listYb3, listZb1}] // Reverse;
listCbd4m = Transpose[{-listXb4, listYb4, listZb1}];

listCbd = Partition[Flatten[{listCbd4m, listCbd3m, listCbd2m,
        listCbd1m, listCbd1, listCbd2, listCbd3, listCbd4}], 3];

(* invisible plotting of the curve Cbd *)

curveCbd = ScatterPlot3D[listCbd, PlotJoined -> True,
     AspectRatio -> Automatic, PlotStyle -> AbsoluteThickness[2],
     DisplayFunction -> Identity];


(*  The list of the curve Sc based on (68) *)

sccosθβ[t_] :=
  r*Sqrt[1 + Sqrt[2]*Cos[t]]*(Cos[θ[t]]*cosβ[t] - Sin[θ[t]]*sinβ[t])
scsinθβ[t_] :=
  r*Sqrt[1 + Sqrt[2]*Cos[t]]*(Sin[θ[t]]*cosβ[t] + Cos[θ[t]]*sinβ[t])
```

```
listsccosθβ = Table[sccosθβ[t] /. t -> i*Pi/(2*j) // N, {i, 0, j}];
listscsinθβ = Table[scsinθβ[t] /. t -> i*Pi/(2*j) // N, {i, 0, j}];

(* Smc4, Smc3, Smc2, Smc1, Sc1, Sc2, Sc3, Sc4
   are segments of curve Sc *)

listXc1 = listXa1 + listsccosθβ;
listYc1 = listYa1 + listscsinθβ;
listZc1 = listZa1;

listXc2 = Table[2*Xc[π/2] - listXc1[[i]], {i, 1, j + 1}];
listYc2 = Table[2*Yc[π/2] - listYc1[[i]], {i, 1, j + 1}];
listZc2 = listZa1;

listXc3 = Table[4*Xc[π/2] - listXc2[[i]], {i, 1, j + 1}];
listYc3 = listYc2;
listZc3 = listZa1;

listXc4 = Table[4*Xc[π/2] - listXc1[[i]], {i, 1, j + 1}];
listYc4 = listYc1;
listZc4 = listZa1;

listSc1 = Transpose[{listXc1, listYc1, listZc1}];
listSc2 = Transpose[{listXc2, listYc2, listZc1}] // Reverse;
listSc3 = Transpose[{listXc3, listYc3, listZc1}];
listSc4 = Transpose[{listXc4, listYc4, listZc1}] // Reverse;
listSc1m = Transpose[{-listXc1, listYc1, listZc1}] // Reverse;
listSc2m = Transpose[{-listXc2, listYc2, listZc1}];
listSc3m = Transpose[{-listXc3, listYc3, listZc1}] // Reverse;
listSc4m = Transpose[{-listXc4, listYc4, listZc1}];

listSc = Partition[Flatten[{listSc4m, listSc3m, listSc2m,
        listSc1m, listSc1, listSc2, listSc3, listSc4}], 3];

(* Invisible plotting of the curve Sc. *)

curveSc = ScatterPlot3D[listSc, PlotJoined -> True,
   AspectRatio -> Automatic, PlotStyle -> {Hue[0.66],
      AbsoluteThickness[2]}, DisplayFunction -> Identity]
```

```
(* Left and right lines *)
pla = {-listXa4[[1]], listYa4[[1]], 0};
plb = {-listXb4[[1]], listYb4[[1]], 0};
pra = {listXa4[[1]], listYa4[[1]], 0};
prb = {listXb4[[1]], listYb4[[1]], 0};
leftline = Show[Graphics3D[Line[{pla, plb}]],
   DisplayFunction -> Identity]
rightline = Show[Graphics3D[Line[{pra, prb}]],
   DisplayFunction -> Identity]


(* A locus of a gravity - center of the T.C.R. *)
listZg1 = Table[r/Sqrt[2], {i, 0, j}];
listg1 = Transpose[{listXc1, listYc1, listZg1}];
listg2 = Transpose[{listXc2, listYc2, listZg1}] // Reverse;
listg3 = Transpose[{listXc3, listYc3, listZg1}];
listg4 = Transpose[{listXc4, listYc4, listZg1}] // Reverse;
listg1m = Transpose[{-listXc1, listYc1, listZg1}] // Reverse;
listg2m = Transpose[{-listXc2, listYc2, listZg1}];
listg3m = Transpose[{-listXc3, listYc3, listZg1}] // Reverse;
listg4m = Transpose[{-listXc4, listYc4, listZg1}];

listg = Partition[Flatten[{listg4m, listg3m, listg2m, listg1m,
        listg1, listg2, listg3, listg4}], 3];

(* "locusg" is a invisible plotting of a locus
     of a gravity - center of the T.C.R. *)

locusg = ScatterPlot3D[listg, PlotJoined -> True,
   AspectRatio -> Automatic, PlotStyle -> {Hue[0.0],
      AbsoluteThickness[2]}, DisplayFunction -> Identity]

(* Invisible plotting of the Cad, Cbd, Sc and locusg *)

developmentTCR = Show[curveCad, curveCbd, curveSc,
   locusg, leftline, rightline, DisplayFunction -> Identity]
```

```
(* A entire plotting of the Moving - Frame for the T.C.R. *)

Print[]
Print[]
Print["Moving Frame Transformation "]
Print[" X = ", X]
Print[" Y = ", Y]
Print[" Z = ", Z]

mfTCR = Show[movCadCbd, movOxyz, movcenterlineCa, movcenterlineCb,
   movO, movAd, movSd, movBd, line1, line2, line3, line4,
   AodXYZ, movΩ, movGw, movMc, developmentTCR,
   DisplayFunction -> $DisplayFunction,
   ViewPoint -> {6.358, -5.485, 3.237}, Boxed -> False,
   AxesEdge -> None, Shading -> False]
```