

```

In[296]:= Clear["Global`*"];
Δr[C1_, C2_] := C1 (r2 - r1) + C2 (r3 - r1) + {字2x, 字2y, 字2z};
AliSeSekata[r1_, r2_, r3_, r4_] := If[
  (*TUKI RAZMISLT 0 ≤*)
  (( (r1 - r2) [[2]] (r4 - r1) [[1]] + (r2 - r1) [[1]] (r4 - r1) [[2]])
    ((r1 - r2) [[2]] (r3 - r1) [[1]] + (r2 - r1) [[1]] (r3 - r1) [[2]]) < 0) ^
  (( (r3 - r4) [[2]] (r1 - r3) [[1]] + (r4 - r3) [[1]] (r1 - r3) [[2]])
    ((r3 - r4) [[2]] (r2 - r3) [[1]] + (r4 - r3) [[1]] (r2 - r3) [[2]]) < 0),
  1,
  0
];
PloskevIz3DV2D[ploskev3D_] := {
  sezvektprod无 = ((#[[2]] - #[[1]]) × (#[[3]] - #[[2]])) & /@ Table[
    {
      ploskev3D[[n]],
      ploskev3D[[
        ((If[# == 0, n + 1, #]) & /@ {Mod[n + 1, Length[ploskev3D]]}) [[1]] ]],
      ploskev3D[[ ((If[# == 0, n + 2, #]) & /@ {Mod[n + 2, Length[ploskev3D]]}) [[1]] ]]]
    ],
    {n, 1, Length[ploskev3D], 1}
  ];
  Do[
    If[sezvektprod无[[i]] ≠ {0, 0, 0}, vekt无 = sezvektprod无[[i]];
    , {i, Length[sezvektprod无]};
    rot无 = RotationMatrix[{vekt无, {0, 0, 1}}];
    Take[(rot无.#), 2] & /@ ploskev3D
  ] [[1]];
Orientacija[ploskev2D_] := Round[
  Total[
    (If[ArcSin[(- (#[[2]] - #[[1]]) [[2]] (#[[3]] - #[[2]]) [[1]] +
      (#[[2]] - #[[1]]) [[1]] (#[[3]] - #[[2]]) [[2]]) /
      (Norm[#[[2]] - #[[1]]) * Norm[#[[3]] - #[[2]])] < 0, -1, 1] *
      VectorAngle[(#[[2]] - #[[1]]), (#[[3]] - #[[2]])] & /@ Table[
        {
          ploskev2D[[n]],
          ploskev2D[[
            ((If[# == 0, n + 1, #]) & /@ {Mod[n + 1, Length[ploskev2D]]}) [[1]] ]],
            ploskev2D[[ ((If[# == 0, n + 2, #]) & /@ {Mod[n + 2, Length[ploskev2D]]}) [[1]] ]]]
          ],
          {n, 1, Length[ploskev2D], 1}
        ]
      ] / (2 π)
    ];
PloskevNaTrikotnike[ploskev3D_] := {
  ploskev2D无 = PloskevIz3DV2D[ploskev3D];
  ploskev2D0st无 = ploskev2D无;
  orientacija无 = Orientacija[ploskev2D无];
  šločeztofazo = 0;

```

```

trikotniki无 = Reap[

  While[Length[ploskev2D0st无] > 3, (*1 krog*)

    n = 1;
    While[n ≤ Length[ploskev2D0st无],
      trojica无 = {
        ploskev2D0st无[[
          (If[# == 0, n + 0, #]) & /@ {Mod[n + 0, Length[ploskev2D0st无]]}][[1]]],
        ploskev2D0st无[[
          ((If[# == 0, n + 1, #]) & /@ {Mod[n + 1,
            Length[ploskev2D0st无]]})[[1]]],
        ploskev2D0st无[[
          ((If[# == 0, n + 2, #]) & /@ {Mod[n + 2,
            Length[ploskev2D0st无]]})[[1]]]
      ];
      If[orientacija无 * (- (trojica无[[2]] - trojica无[[1]])[[2]]
        (trojica无[[3]] - trojica无[[2]])[[1]] + (trojica无[[2]] -
          trojica无[[1]])[[1]] (trojica无[[3]] - trojica无[[2]])[[2]]) > 0,
        odreži = 1;
        Do[
          odreži *= 1 - AliSeSekata[
            trojica无[[1]],
            trojica无[[3]],
            ploskev2D0st无[[i]],
            ploskev2D0st无[[
              ((If[# == 0, i + 1, #]) & /@
                {Mod[i + 1, Length[ploskev2D0st无]]})[[1]]]
            ], {i, Length[ploskev2D0st无]};
          If[odreži == 1,
            šločeztofazo++;
            Sow[trojica无];
            ploskev2D0st无 = Delete[ploskev2D0st无,
              (If[# == 0, n + 1, #]) & /@ {Mod[n + 1, Length[ploskev2D0st无]]}][[1]]
            ];
          n++];

    ];

    If[Length[ploskev2D0st无] == 3,
      Sow[ploskev2D0st无]

    ]

  ][[2]][[1]];
  ((Position[ploskev2D无, #][[1, 1]]) & /@ #) & /@ trikotniki无

]

]

PovezaveOglišč[ploskve_] := {
  vmesnisez1 = ((Position[ploskve, #]) & /@ #) & /@ ploskve;
  vmesnisez2 = Table[
    DeleteCases[vmesnisez1[[plo, ogl]], {plo, ogl}],

```

```

    {plo, Length[vmesnizez1]}, {ogl, Length[vmesnizez1[[plo]]]}];
vmesnizez3 = Table[
  (If[Count[Transpose[Flatten[vmesnizez2[[plo]]], 1]][[1]], #[[1]]] < 2,
    {},
    #
  ] & /@ vmesnizez2[[plo, ogl]],
  {plo, Length[vmesnizez2]}, {ogl, Length[vmesnizez2[[plo]]]}];
((DeleteCases[#, {}]) & /@ #) & /@ vmesnizez3
}[[1]];

PovezavePloskev[povezaveoglišč_] :=
  (DeleteDuplicates[Transpose[Flatten[#, 1]][[1]]]) & /@ povezaveoglišč;

(*
VektPregibov[skupogl_] := {
  ena = skupogl[[1]];
  odenadodva = skupogl[[2]] - ena;
  sortiranaskupogl = skupogl[[Ordering@
    (Total[(If[#[[2]] == 0, 0, #[[1]]/#[[2]]]) & /@ Transpose[{# - ena, odenadodva}]] & /@
    skupogl)]];
  N[Total[Table[(-1)^i, {i, Length[sortiranaskupogl]}] sortiranaskupogl]
]
}[[1]];

VektPregibovSez[ploskve_] := Table[
  If[p1 == p2,
    0,
    skogl = ploskve[[p1]] ∩ ploskve[[p2]];
    If[Length[skogl] < 2,
      0,
      VektPregibov[skogl]
    ]
  ],
  {p1, Length[ploskve]}, {p2, Length[ploskve]}}];

Lpregibov[skupogl_] := {
  ena = skupogl[[1]];
  odenadodva = skupogl[[2]] - ena;
  sortiranaskupogl = skupogl[[Ordering@
    (Total[(If[#[[2]] == 0, 0, #[[1]]/#[[2]]]) & /@ Transpose[{# - ena, odenadodva}]] & /@
    skupogl)]];
  N[Norm[Total[Table[(-1)^i, {i, Length[sortiranaskupogl]}] sortiranaskupogl]]
]
}[[1]];

LpregibovSez[ploskve_] := Table[
  If[p1 == p2,
    0,
    skogl = ploskve[[p1]] ∩ ploskve[[p2]];
    If[Length[skogl] < 2,
      0,
      Lpregibov[skogl]
    ]
  ],
  {p1, Length[ploskve]}, {p2, Length[ploskve]}}];

```

*)

```

Pravokotni[ploskev3D_] :=
  {sezvektprod无 = ((#[[2]] - #[[1]]) × (#[[3]] - #[[2]])) & /@ Table[{ploskev3D[[n]],
    ploskev3D[[ (If[# == 0, n + 1, #]) & /@ {Mod[n + 1, Length[ploskev3D]]}][[1]]],
    ploskev3D[[ (If[# == 0, n + 2, #]) & /@ {Mod[n + 2, Length[ploskev3D]]}][[1]]]},
    {n, 1, Length[ploskev3D], 1}];
Do[If[sezvektprod无[[i]] ≠ {0, 0, 0}, vekt无 = sezvektprod无[[i]]];,
  {i, Length[sezvektprod无]}];
N[vekt无][[1]];

```

```

Ploščina[trikotnik_] :=

```

```

.5 Norm[(trikotnik[[3]] - trikotnik[[1]]) × (trikotnik[[2]] - trikotnik[[1]])];

```

```

Ploščine[trikotniki_] := Ploščina[#] & /@ trikotniki;

```

```

Težišče[trikotnik_] := N[Total[trikotnik] / 3];

```

```

Težišča[trikotniki_] := Težišče[#] & /@ trikotniki;

```

```

SkupnoTežišče[ploščine_, težišča_] := Total[ploščine težišča] / Total[ploščine];

```

```

J[trikotnik_, ρ2_, tež_] := {

```

```

  (
    r1x r1y r1z
    r2x r2y r2z
    r3x r3y r3z
  ) = {r1, r2, r3} = trikotnik;

```

```

  字1 = (1 / 24) ρ2 √ ( (r1y r2x - r1x r2y - r1y r3x + r2y r3x + r1x r3y - r2x r3y)² +
    (r1z r2x - r1x r2z - r1z r3x + r2z r3x + r1x r3z - r2x r3z)² +
    (r1z r2y - r1y r2z - r1z r3y + r2z r3y + r1y r3z - r2y r3z)² );

```

```

  {字2x, 字2y, 字2z} = r1 - tež;

```

```

  字1

```

```

  { {2 (6 字2y² + 6 字2z² - 8 字2y r1y + 3 r1y² - 8 字2z r1z + 3 r1z² + 4 字2y r2y - 3 r1y r2y + r2y² +
    4 字2z r2z - 3 r1z r2z + r2z² + 4 字2y r3y - 3 r1y r3y + r2y r3y + r3y² + 4 字2z r3z -
    3 r1z r3z + r2z r3z + r3z²), -12 字2x 字2y + 8 字2y r1x + 8 字2x r1y - 6 r1x r1y -
    4 字2y r2x + 3 r1y r2x - 4 字2x r2y + 3 r1x r2y - 2 r2x r2y - 4 字2y r3x + 3 r1y r3x -
    r2y r3x - 4 字2x r3y + 3 r1x r3y - r2x r3y - 2 r3x r3y, -12 字2x 字2z + 8 字2z r1x +
    8 字2x r1z - 6 r1x r1z - 4 字2z r2x + 3 r1z r2x - 4 字2x r2z + 3 r1x r2z - 2 r2x r2z -
    4 字2z r3x + 3 r1z r3x - r2z r3x - 4 字2x r3z + 3 r1x r3z - r2x r3z - 2 r3x r3z},
    { -12 字2x 字2y + 8 字2y r1x + 8 字2x r1y - 6 r1x r1y - 4 字2y r2x + 3 r1y r2x -
    4 字2x r2y + 3 r1x r2y - 2 r2x r2y - 4 字2y r3x + 3 r1y r3x - r2y r3x - 4 字2x r3y +
    3 r1x r3y - r2x r3y - 2 r3x r3y, 2 (6 字2x² + 6 字2z² - 8 字2x r1x + 3 r1x² -
    8 字2z r1z + 3 r1z² + 4 字2x r2x - 3 r1x r2x + r2x² + 4 字2z r2z - 3 r1z r2z + r2z² +
    4 字2x r3x - 3 r1x r3x + r2x r3x + r3x² + 4 字2z r3z - 3 r1z r3z + r2z r3z + r3z²),
    -12 字2y 字2z + 8 字2z r1y + 8 字2y r1z - 6 r1y r1z - 4 字2z r2y + 3 r1z r2y -
    4 字2y r2z + 3 r1y r2z - 2 r2y r2z - 4 字2z r3y + 3 r1z r3y -
    r2z r3y - 4 字2y r3z + 3 r1y r3z - r2y r3z - 2 r3y r3z},
    { -12 字2x 字2z + 8 字2z r1x + 8 字2x r1z - 6 r1x r1z - 4 字2z r2x + 3 r1z r2x -
    4 字2x r2z + 3 r1x r2z - 2 r2x r2z - 4 字2z r3x + 3 r1z r3x - r2z r3x - 4 字2x r3z +
    3 r1x r3z - 2 r3x r3z, -12 字2y 字2z + 8 字2z r1y + 8 字2y r1z -
    6 r1y r1z - 4 字2z r2y + 3 r1z r2y - 4 字2y r2z + 3 r1y r2z - 2 r2y r2z -
    4 字2z r3y + 3 r1z r3y - r2z r3y - 4 字2y r3z + 3 r1y r3z - r2y r3z - 2 r3y r3z,
    2 (6 字2x² + 6 字2y² - 8 字2x r1x + 3 r1x² - 8 字2y r1y + 3 r1y² + 4 字2x r2x -
    3 r1x r2x + r2x² + 4 字2y r2y - 3 r1y r2y + r2y² + 4 字2x r3x -
    3 r1x r3x + r2x r3x + r3x² + 4 字2y r3y - 3 r1y r3y + r2y r3y + r3y²) } }

```

```

}[[1]];

```

```

Jji[trikotniki_, ρ2_, tež_] := J[#, ρ2, tež] & /@ trikotniki;
SkupniJ[jji_] := Total[jji];

Fpomož4[C1_, C2_, trikotnik_, tež_, V_, ω_, n_] := {
  {r1_, r2_, r3_} = trikotnik_;
  Δr = (r1_ + C1 (r2_ - r1_) + C2 (r3_ - r1_) - tež);
  v_ = V + ω × Δr;
  If[v_.n > 0, -1, 1] (v_.n)2 (Δr × n)
}[[1]];

MzrakaTrikotnik[trikotnik_, tež_, V_, ω_, n_] := {
  {r1_, r2_, r3_} = trikotnik_;
  ρ Norm[(r2_ - r1_) × (r3_ - r1_)] * NIntegrate[
    Fpomož4[C1, C2, trikotnik_, tež, V, ω, n]
    , {C1, 0, 1}, {C2, 0, 1 - C1}]
}[[1]];

MzrakaPloskev[trikotniki_, tež_, V_, ω_, n_] :=
  Total[MzrakaTrikotnik[#, tež, V, ω, n] & /@ trikotniki];

KotV1V20koliV3[vekt1_, vekt2_, vekt3_] := {
  kot0 = VectorAngle[vekt1, vekt2];
  If[(vekt1 × vekt2) . vekt3 > 0,
    kot0,
    2 π - kot0
  ]
}[[1]];

Deltafi[vekt1_, vekt2_, vekt3_, Δφ0_] :=
  {(*velja, če spremembe kota niso večje od π, ka realno najbrž tut niso*)
    Δφ = KotV1V20koliV3[vekt1, vekt2, vekt3] - Δφ0;
    Δφ += If[Δφ > π, -2 π, 0];
    Δφ += If[Δφ < -π, 2 π, 0];
    Δφ
  }[[1]];

Frez[mase_, g_, k_, ploskve_, ploščinecelihploskev_,
  pritrdišča_, kordpritrdišča_, povezaveoglišč_] := Table[
  mase[[i]] g +
  Total[Table[
    If[pritrdišča[[i, i2]] == 0,
      {0, 0, 0},
      r1 = ploskve[[i, i2]];
      r2 = kordpritrdišča[[i, i2]];
      rt1 = tež[[i]];
      vt1 = v[[i]];
      ω1 = Ω[[i]];
      vektvzmet = r2 - r1;
      lvektvzmet = Norm[vektvzmet];
      lkadrati += lvektvzmet2;
      (*v imenovalcu, če je 0 napišemo pač neki npr 1*)
      (k - (kd (vt1 + ω1 × (r1 - rt1)) . vektvzmet) / If[lvektvzmet == 0, 1, lvektvzmet])
      vektvzmet
    ] +
    If[povezaveoglišč[[i, i2]] == {},

```

```

{0, 0, 0},
Total[
(
  r1 = ploskve[[i, i2]];
  r2 = ploskve[[#[[1]], #[[2]]]];
  rt1 = tež[[i]];
  rt2 = tež[[#[[1]]]];
  vt1 = v[[i]];
  vt2 = v[[#[[1]]]];
  ω1 = Ω[[i]];
  ω2 = Ω[[#[[1]]]];
  vektvzmet = r2 - r1;
  lvektvzmet = Norm[vektvzmet];
  lkadrati += lvektvzmet2 / 2 (*polovica, ker bomo 2krat štel*);
  (*v imenovalcu, če je 0 napišemo pač neki npr 1*)
  (k + (kd (vt2 + ω2 × (r2 - rt2) - vt1 - ω1 × (r1 - rt1)) . vektvzmet) /
    If[lvektvzmet == 0, 1, lvektvzmet]) vektvzmet
) & /@povezaveoglišč[[i, i2]]]
(*GOR za vsako ogl. na drugi ploskvi s kjerim je to povezano *)
],
{i2, Length[ploskve[[i]]]}] +
Fzraka[[i]],
{i, Length[ploskve]}}];

```

```

Mrez[k_, DD_, pravokotni_, ploskve_,
  tež_, pritrdišča_, kordpritrdišča_, povezaveoglišč_] := {

```

```

Table[
  Total[Table[
    If[pritrdišča[[i, i2]] == 0,
      {0, 0, 0},
      r1 = ploskve[[i, i2]];
      r2 = kordpritrdišča[[i, i2]];
      rt1 = tež[[i]];
      vt1 = v[[i]];
      ω1 = Ω[[i]];
      vektvzmet = r2 - r1;
      lvektvzmet = Norm[vektvzmet];
      (*v imenovalcu, če je 0 napišemo pač neki npr 1*)
      (r1 - rt1) × ((k - (kd (vt1 + ω1 × (r1 - rt1)) . vektvzmet) /
        If[lvektvzmet == 0, 1, lvektvzmet]) vektvzmet)
    ] +
    If[povezaveoglišč[[i, i2]] == {},
      {0, 0, 0},
      Total[
        (
          r1 = ploskve[[i, i2]];
          r2 = ploskve[[#[[1]], #[[2]]]];
          rt1 = tež[[i]];
          rt2 = tež[[#[[1]]]];
          vt1 = v[[i]];
          vt2 = v[[#[[1]]]];
          ω1 = Ω[[i]];

```

```

       $\omega_2 = \Omega[\#[[1]]]$ ;
      vektvzmet = r2 - r1;
      lvektvzmet = Norm[vektvzmet];
      (*v imenovalcu, če je 0 napišemo pač neki npr 1*)
      (r1 - rt1) * ((k + (kd (vt2 +  $\omega_2 \times (r2 - rt2)$  - vt1 -  $\omega_1 \times (r1 - rt1)$ )).
        vektvzmet) / If[lvektvzmet == 0, 1, lvektvzmet]) vektvzmet)
    ) & /@povezaveoglišč[[i, i2]]
    (*GOR za vsako ogl. na drugi ploskvi s kjerim je to povezano *)
  ],
  {i2, Length[ploskve[[i]]]}] +
Total[
  Table[
     $\Delta\varphi = \Delta\varphi_{Plo1Plo2}[[i, i2]]$ ;
     $\varphi_{kvadrati} += \Delta\varphi^2$ ;
    If[ListQ[ $\Delta\varphi$ ], (*če je {}*)
      {0, 0, 0},
      DD  $\Delta\varphi \times$  vektpregibov[[i, i2]]
    ],
    {i2, Length[ploskve]}]
  ] +
  Mzraka[[i]],
  {i, Length[ploskve]}}][[1]];

NaključnaTVKrogli[] := {
  Točka无 = {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]}];
  If[Norm[Točka无] < 1,
    Točka无,
    NaključnaTNaKrogli[]
  ]][[1]];

NaključnaTNaKrogli[] := {
  Točka无 = {RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]}];
  If[Norm[Točka无] < 1,
    Normalize[Točka无],
    NaključnaTNaKrogli[]
  ]][[1]];

Funkcija无1[x无1_] := If[x无1 == {}, {}, ploskve[[x无1[[1]], x无1[[2]]]]];
(*pomožna funkcija za map*)
Funkcija无1Plo0[x无1_] := If[x无1 == {}, {}, ploskve0[[x无1[[1]], x无1[[2]]]]];
(*pomožna funkcija za map*)

Wp[tež_, mase_] := gg Total[Table[
  mase[[i]] * tež[[i, 3]],
  {i, Length[ploskve]}]];

Wk[v_, mase_] := (1/2) Total[Table[
  mase[[i]] * Norm[v[[i]]]^2,
  {i, Length[ploskve]}]];

Wr[ $\Omega$ _, j_] := (1/2) Total[Table[
   $\Omega[[i]] \cdot j[[i]] \cdot \Omega[[i]]$ ,
  {i, Length[ploskve]}]];

```

(*za Wpr se bo bl slačal sprot pr silah*)

```

ρ = 1.2;
ρ2 = .16;
vzorcevnasliko = 200;
dt = N[1 / (vzorcevnasliko * 60)];
g = {0, 0, -9.81};
gg = -g[[3]];
k = 1000  $\left(\frac{1.4}{10}\right)^2$  * .16 * 70;

```


$$kd = 100 \left(\frac{1.4}{10} \right)^2 * .16;$$

$$DD = 100 \left(\frac{1.4}{10} \right)^2 * .16 * .5;$$

```

ploskve0 = ((RotationMatrix[0 * (π/2), {0, 1, 0}].#) & /@#) & /@N[{
  {
    {-4, -1, 4},
    {0, -1, 0},
    {0, 0, 0},
    {-4, 0, 4}
  },
  {
    {0, 0, 0},
    {4, 0, 4},
    {4, 1, 4},
    {0, 1, 0}
  },
  {
    {0, -1, 0},
    {0, -1, -2},
    {0, -.5, -5},
    {0, -.5, -14 + 5 * (1.5/1.4)},
    {0, .5, -14 + 5 * (1.5/1.4)},
    {0, .5, -5},
    {0, 1, -2},
    {0, 1, 0},
    {0, 0, 0}
  },
  {
    {0, -.5, -14 + 5 * (1.5/1.4)},
    {0, -.5, -14 + 5 * (1.5/1.4)} + (1.5/1.4) Normalize[ {.3, 0, 1}],
    {0, .5, -14 + 5 * (1.5/1.4)} + (1.5/1.4) Normalize[ {.3, 0, 1}],
    {0, .5, -14 + 5 * (1.5/1.4)}
  }
} * 1.4/100];

sezWp = {};

pritrdisča = Table[0, Length[#]] & /@ ploskve0;

kordpritrdisča = Table[
  (*če je pritrjeno, kam? (začetna lega)*)
  If[pritrdisča[[i, j]] == 0,
    0,
    ploskve0[[i, j]]
  ],
  {i, Length[pritrdisča]}, {j, Length[pritrdisča[[i]]]}}];

triangulacija = (PloskevNaTrikotnike[#]) & /@ ploskve0;
(*seznam triangulacij ploskev*)
trikotniki0 =
  Table[{(ploskve0[[i, #]]) & /@#} & /@ triangulacija[[i]], {i, Length[ploskve0]}}];

```

```

ploščine = (Ploščine[#]) & /@trikotniki0;
(*seznam seznamov ploščin trikotnikov od ploskev*)
ploščinecelihploskev = Total[#] & /@ploščine;
mase =  $\rho_2$  {1, 1, 1, 5} ploščinecelihploskev;
povezaveoglišč = PovezaveOglišč[ploskve0];
povezaveploskev = PovezavePloskev[povezaveoglišč];
(*lpregibovsez=LpregibovSez[ploskve0];*)

OglPlo1Plo2IndVPloskve = Table[{}, {i1, Length[ploskve0]}, {i2, Length[ploskve0]}}];
Do[
  AppendTo[OglPlo1Plo2IndVPloskve[[i1,
    povezaveoglišč[[i1, i2, i3, 1]] (*ind plo1,ind plo2*) ]], { {i1, i2},
    {povezaveoglišč[[i1, i2, i3, 1]], povezaveoglišč[[i1, i2, i3, 2]] } ]],

    {i1, Length[povezaveoglišč]}, (*ind ploskev*)
    {i2, Length[povezaveoglišč[[i1]] ]}, (*ind oglišče*)
    {i3, Length[povezaveoglišč[[i1, i2]] ]}
    (*ktero povrsti od povezanih oglišč z nekim ogliščem*)
  ];
(*za plo1, za plo2, za vsako skupno oglišče,
{indeks v ploskvah1, indeks v ploskvah 2}*)
(*tedva indeksa sta pa ind ploskve, indoglišča*)
(*2 sta pa ker gledamo skupna oglišča 2 ploskev*)
pravokotni0 = Pravokotni[#] & /@ploskve0;
KordOglPlo1Plo2 = Map[Funkcija无1Plo0, OglPlo1Plo2IndVPloskve, {-2}];
vektpregibov =
  ((Flatten[If[# == {}, {}, {#[[2]] - #[[1]]}] & /@{Total[#] / 2 & /@#(*pod indplo1,
    indplo2*)}]] & /@#) & /@KordOglPlo1Plo2;
ZačetniKotiPlo1Plo2 = Table[
  vekt3无 = vektpregibov[[i1, i2]];
  If[
    vekt3无 == {},
    {},
    KotV1V20koliV3[pravokotni0[[i1]], pravokotni0[[i2]], vekt3无]
  ],
  {i1, Length[ploskve0]}, {i2, Length[ploskve0]}}];

```

(*****)

```
(*****
(*****OD TUKI SO STVARI, K SE SPREMINJAJO*****
(*****
(*****
```

```
ploskve = ploskve0;
bazni = Table[{{1, 0, 0}, {0, 1, 0}, {0, 0, 1}}, Length[ploskve]];
trikotniki = trikotniki0; (*seznam trikotnikov ploskev*)
pravokotni = Pravokotni[#] & /@ ploskve;
```

```
tež = Table[
  SkupnoTežišče[ploščine[[i]], Težišča[trikotniki[[i]] ]],
  {i, Length[trikotniki]}; (*seznam težišč ploskev*)
j = {1, 1, 1, 5} Table[
  SkupniJ[Jji[trikotniki[[i]], ρ2, tež[[i]] ]],
  {i, Length[ploskve]}; (*seznam J-jev ploskev*)
```

```
Ω = {0, 0, 0} & /@ ploskve;      (*seznam ω za vsako ploskev*)
v = {0, 0, 0} & /@ ploskve;      (*seznam V za vsako ploskev*)
```

```
Γ = Table[
  j[[i]].Ω[[i]],
  {i, Length[trikotniki]}; (*seznam Γ ploskev*)
n = Normalize[#] & /@ pravokotni;
Mzraka = Table[
  MzrakaPloskev[trikotniki[[i]], tež[[i]], v[[i]], Ω[[i]], n[[i]] ],
  {i, Length[ploskve]};
```

```
A = 0;
W0 = Wp[tež, mase] + Wk[v, mase] + Wr[Ω, j];
```

```
pot0 = "c:\\Users\\gal\\Downloads\\" <>
  StringDrop[StringSplit[NotebookFileName[], "\\"]][[-1]], -3] <> "\\";
If[Not[DirectoryQ[pot0]],
  CreateDirectory[pot0]];
```

```

štej = 0;
Do[
  Kord0glPlo1Plo2 = Map[Funkcija无1, OglPlo1Plo2IndVPloskve, {-2}];
  vektpregibovprejšnji = vektpregibov;
  vektpregibov =
    ((Flatten[If[# == {}, {}, (#[[2]] - #[[1]])] & /@ {Total[#] / 2 & /@# (*pod indplo1,
      indplo2*)}]] & /@#) & /@ Kord0glPlo1Plo2;
  vektpregibov = Table[
    dfk11 = vektpregibov[[i1, i2]];
    If[dfk11 == {},
      {},
      If[dfk11.vektpregibovprejšnji[[i1, i2]] > 0,
        dfk11,
        -dfk11
      ]
    ],
    {i1, Length[ploskve]}, {i2, Length[ploskve]}}];
  ΔφPlo1Plo2 = Table[
    vekt3无 = vektpregibov[[i1, i2]];
    If[
      vekt3无 == {},
      {},
      Deltafi[pravokotni[[i1]],
        pravokotni[[i2]], vekt3无, ZačetniKotiPlo1Plo2[[i1, i2]] ]
    ],
    {i1, Length[ploskve0]}, {i2, Length[ploskve0]}}];

  lkvadrati = 0;
  φkvadrati = 0;
  Fzraka = Table[
    VVSmerin = v[[i]].n[[i]];
    If[VVSmerin > 0, -1, 1] * ploščinecelihploskev[[i]] ρ (VVSmerin)2 n[[i]],
    {i, Length[ploskve]}}];

  F = Frez[mase, g, k, ploskve,
    ploščinecelihploskev, pritrdišča, kordpritrdišča, povezaveoglišč];
  M = Mrez[k, DD, pravokotni, ploskve, tež, pritrdišča, kordpritrdišča, povezaveoglišč];
  Wpr = (k / 2) lkvadrati;
  Wprm = (DD / 2) φkvadrati;
  A += -Total[Table[
    Mzraka[[i]].Ω[[i]] + Fzraka[[i]].v[[i]],
    {i, Length[ploskve]}}] dt;

  Γ += M dt;
  menjavaks = (Transpose[#]) & /@ bazni;
  Ω = Table[
    menjavaks[[i]].Inverse[j[[i]]].Inverse[menjavaks[[i]]].Γ[[i]],
    {i, Length[ploskve]}}];
  v += Table[(F[[i]] / (mase[[i]])) dt, {i, Length[ploskve]}}];

  ξ = 1;
  (*ξ=If[štej<10vzorcevnasliko,
    1,

```

```


$$\sqrt{\text{Abs}[(W_0 - A - W_p[\text{tež}, \text{mase}] - W_{pr} - W_{prm}) / (W_k[v, \text{mase}] + W_r[\Omega, j])]}];$$


v* = ξ;
Ω* = ξ;

*)
RotM = If[Norm[#] == 0,
  
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

  RotationMatrix[Norm[#] dt, #]
] & /@ Ω;
ploskve = Table[
  (# + tež[[i]] + v[[i]] dt) & /@
  ((RotM[[i]].#) & /@ ((# - tež[[i]]) & /@ ploskve[[i]])),
  {i, Length[ploskve]}];
tež += Table[v[[i]] dt, {i, Length[ploskve]}];
bazni = Table[(RotM[[i]].#) & /@ bazni[[i]], {i, Length[ploskve]}];
pravokotni = Table[RotM[[i]].pravokotni[[i]], {i, Length[ploskve]}];
If[IntegerQ[2 štejj / vzorcevnasliko],
  n = Normalize[#] & /@ pravokotni;
  trikotniki =
    Table[(ploskve[[i, #]]) & /@ # & /@ triangulacija[[i]], {i, Length[ploskve]}];
  Mzraka = Table[
    MzrakaPloskev[trikotniki[[i]], tež[[i]], v[[i]], Ω[[i]], n[[i]] ],
    {i, Length[ploskve]}];
];

(*****)

If[IntegerQ[štejj / vzorcevnasliko],
  (*AppendTo[sezWp, Wp[tež, mase]];*)
  grafika = Show[

    Graphics3D[{
      RGBColor[0, 1, 1, 1],
      EdgeForm[],

      Polygon[

        #

      ]
    }] & /@ ploskve,
    (*Table[
      Graphics3D[{
        RGBColor[0, 1, 1, 1],
        Arrowheads[.05],

        Arrow[Tube[{tež[[i]], tež[[i]] + M[[i]]},
          .005]]
      }],
    ]),

```

```

    {i, Length[ploskve] }], *)
(*Graphics3D[{
    RGBColor[1, 0, 0, 1],
    Arrowheads[.02],

    Arrow[Tube[{tež, tež+ $\omega$ /200},
    .0001]]
}],
*)
Boxed → False,
ViewPoint → .2 {0, -20, 5},
(*SphericalRegion→Sphere[{0, 0, -10}, 10.2], *)

(*PlotRange→{{, }, {, }, {, }}, *)
(*Background→Black, *)
ImageSize → {1920, 1080}
];
(*****

številka = ToString[štej / vzorcevna_slika];
Which[
    StringLength[številka] == 1,
    številka = "000" <> številka,
    StringLength[številka] == 2,
    številka = "00" <> številka,
    StringLength[številka] == 3,
    številka = "0" <> številka
];

pot = pot0 <> "slika" <> številka <> ".png";
Export[pot, grafika];
];
štej++;
(*****
    , {t, 0, 5, dt}];
(*****

Run["naredi-ff-video \"\" <> StringTake[pot, StringLength[pot0] - 1] <> \"\""];

```