

Priprava koorinat strukturirane mreže za cilindar

Podane koordinate plašča cilindra

```
In[298]:= << AceFEM` ; (*naloži programski paket AceFEM*)
          gc = Rest[Import["def_geo.xlsx"][[1]]];
          (*prikliče podatke iz excelove datoteke "def_geo"*)
```

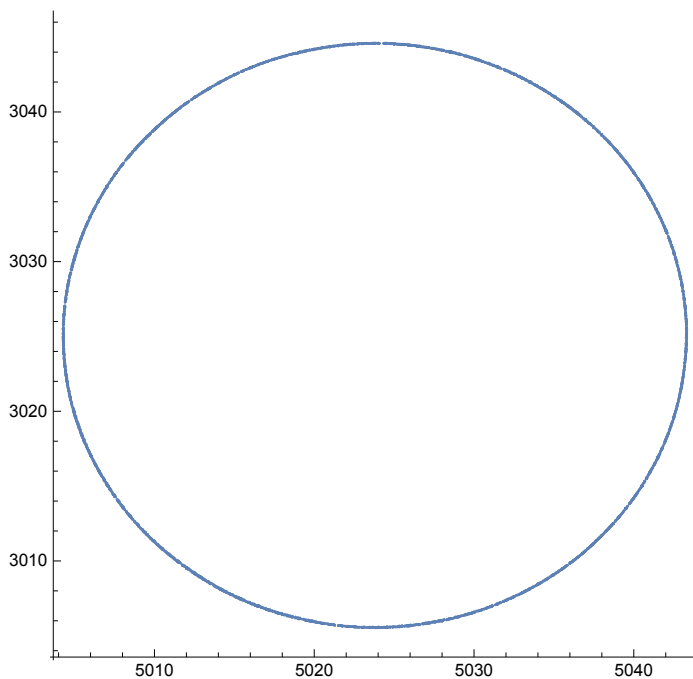
(Opomba : Rest pomeni da se iz seznama izloči prvi člen, ki v tem primeru ne predstavlja koordinat)

Določitev x in y koordinat središča valja

Določitev x in y koordinat središča valja (CenterValja)

```
In[300]:= CenterValja = Mean[gc[[All, 2 ;; 3]]]
          {5023.72, 3025.2}
```

```
In[301]:= ListPlot[gc[[All, 2 ;; 3]], AspectRatio -> 1]
```



Translatorni premik središča cilindra v X in Y smeri v koordinatno izhodišče

```
In[302]:= gc[[All, 2]] = gc[[All, 2]] - CenterValja[[1]];
          gc[[All, 3]] = gc[[All, 3]] - CenterValja[[2]];
```

Določitev koordinate z spodnjega roba (zmin), koordinate z zgornjega roba (zmax) cilindra (v poročilu je nominalna višina cilindra 9.144m)

```
In[304]:= zmin1 = Min[gc[[All, 4]]]
          zmax1 = Max[gc[[All, 4]]]

          49.852

          59.05
```

```
In[306]:= Δz = ((zmax1 - zmin1) - 9.144) / 2;
```

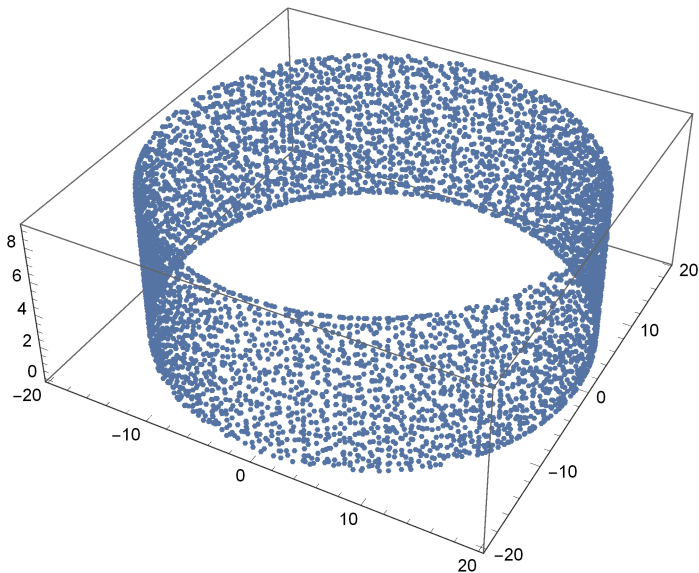
```
In[307]:= zmin = zmin1 + Δz
          zmax = zmax1 - Δz

          49.879

          59.023
```

Premik cilindra tako, da je z koordinata spodnjega roba (zmin): 0

```
In[309]:= gc[[All, 4]] = gc[[All, 4]] - zmin;
In[310]:= ListPointPlot3D[gc[[All, 2 ;; 4]]]
```



```
In[311]:= zmax = zmax - zmin
          zmin = 0

          9.144

          0
```

Prevod v polarne koordinate

Določitev kota ϕ (ϕ_{all}) ter radija (radij) za posamezno točko

```
In[313]=  $\phi_{all} = \text{Map}[\text{Which}[\text{gc}[[\#, 2]] \geq 0 \ \&\& \ \text{gc}[[\#, 3]] \geq 0,$ 
 $\text{ArcTan}[\text{gc}[[\#, 3]] / \text{gc}[[\#, 2]]]$ 
 $, (\text{gc}[[\#, 2]] < 0 \ \&\& \ \text{gc}[[\#, 3]] \geq 0) \ || \ (\text{gc}[[\#, 2]] < 0 \ \&\& \ \text{gc}[[\#, 3]] < 0),$ 
 $(\pi + \text{ArcTan}[\text{gc}[[\#, 3]] / \text{gc}[[\#, 2]])]$ 
 $, \text{gc}[[\#, 2]] \geq 0 \ \&\& \ \text{gc}[[\#, 3]] < 0,$ 
 $(2 \pi + \text{ArcTan}[\text{gc}[[\#, 3]] / \text{gc}[[\#, 2]])]$ 
 $] \ \&, \text{Range}[\text{Length}[\text{gc}]]];$ 
```

```
In[314]=  $\text{radij} = (\text{gc}[[\text{All}, 2]]^2 + \text{gc}[[\text{All}, 3]]^2)^{0.5};$ 
```

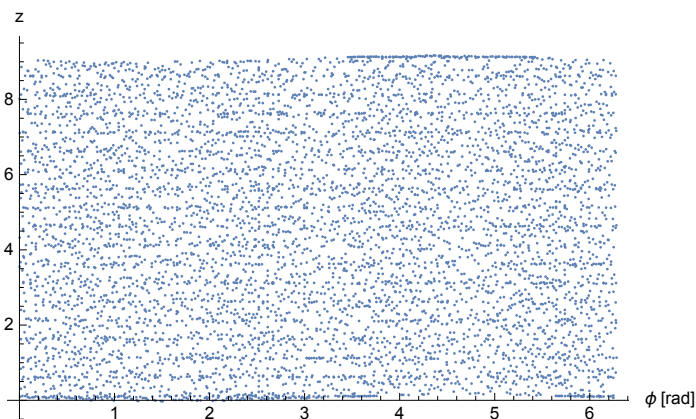
Določitev povprečnega radija (povprečenRadij)

```
In[315]=  $\text{PovprečenRadij} = \text{Mean}[\text{radij}]$ 
19.5059
```

Razvoj oboda valja v ravnino

```
In[316]=  $\text{razvitValj} = \text{Transpose}[\{\phi_{all}, \text{gc}[[\text{All}, 4]], \text{radij}\}];$ 
```

```
In[317]=  $\text{ListPlot}[\text{razvitValj}[[\text{All}, 1 ;; 2]], \text{AxesLabel} \rightarrow \{\phi \text{ [rad]}, \text{"z"}\}]$ 
```



Razširitev točk (za gladko interpolacijo)

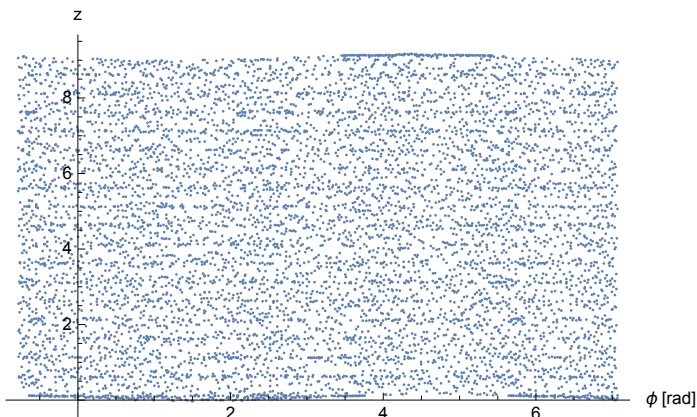
Ker je prerez po obodu cilindra sklenjen krog - na interpolacijo geometrijo pri $\phi=0$ in $\phi=2\pi$ vplivajo tudi točke preko tega "roba" - razviti valj razširimo še za $\pi/4$ zaradi interpolacije na vsako stran:

```
In[318]=  $\text{razsiritevPrekoRobov1} = \text{Select}[\text{razvitValj}, \#[[1]] > 7 \pi / 4 \ \&];$ 
 $\text{razsiritevPrekoRobov1}[[\text{All}, 1]] = \text{razsiritevPrekoRobov1}[[\text{All}, 1]] - 2 \pi;$ 
```

```
In[320]=  $\text{razsiritevPrekoRobov2} = \text{Select}[\text{razvitValj}, \#[[1]] < \pi / 4 \ \&];$ 
 $\text{razsiritevPrekoRobov2}[[\text{All}, 1]] = \text{razsiritevPrekoRobov2}[[\text{All}, 1]] + 2 \pi;$ 
```

```
In[322]=  $\text{razvitValj} = \text{Join}[\text{razvitValj}, \text{razsiritevPrekoRobov1}, \text{razsiritevPrekoRobov2}];$ 
```

```
In[323]= ListPlot[razvitValj[[All, 1 ;; 2]], AxesLabel -> {" $\phi$  [rad]", "z"}]
```



Določitev koordinat točk po spodnjem in zgornjem robu oboda cilindra:

Določimo, na koliko točk (enakomerno razdeljenih vzdolž sp. in zg. roba oboda) naj se rob oboda razdeli

```
In[324]= stTockObod = 612;
```

Koti ϕ ki pripadajo posameznim točkam

```
In[325]= phiPoOboduSP = Table[{phi, zmin}, {phi, 0, 2 pi - 2 pi / stTockObod, 2 pi / stTockObod}];
          phiPoOboduZG = Table[{phi, zmax}, {phi, 0, 2 pi - 2 pi / stTockObod, 2 pi / stTockObod}];
```

Koordinate točk naj se določijo iz povprečja koordinat n bližnjih podanih koordinat oboda cilindra

```
In[327]= nBliznjihTock = 2;
```

Seznam n bližnjih podanih koordinat oboda cilindra za vsako točko

```
In[328]= seznamSP =
          Table[Nearest[Select[razvitValj[[All, 1 ;; 2]], phiPoOboduSP[[i, 1]] - 2 pi / stTockObod <
            #[[1]] <= phiPoOboduSP[[i, 1]] + 2 * 2 pi / stTockObod &],
            phiPoOboduSP[[i]], nBliznjihTock], {i, Length[phiPoOboduSP]}];
          seznamZG = Table[Nearest[Select[razvitValj[[All, 1 ;; 2]], phiPoOboduZG[[i, 1]] -
            2 pi / stTockObod <= #[[1]] <= phiPoOboduZG[[i, 1]] + 2 * 2 pi / stTockObod &],
            phiPoOboduZG[[i]], nBliznjihTock], {i, Length[phiPoOboduZG]}];
```

Izračun radija za vsako točko

```
In[331]= radijSP = Table[Mean[razvitValj[[All, 3]]][[
          Map[Position[razvitValj[[All, 1 ;; 2]], seznamSP[[i, #]]][[1, 1]] &,
          Range[nBliznjihTock]]]], {i, Length[phiPoOboduSP]}];
          radijZG = Table[Mean[razvitValj[[All, 3]]][[
          Map[Position[razvitValj[[All, 1 ;; 2]], seznamZG[[i, #]]][[1, 1]] &,
          Range[nBliznjihTock]]]], {i, Length[phiPoOboduZG]}];
```

Koordinate točk so tako

```
In[333]= tockePoOboduSP = Map[Append[phiPoOboduSP[[#]], radijSP[[#]] &, Range[Length[phiPoOboduSP]]];
          AppendTo[tockePoOboduSP, {2 pi, tockePoOboduSP[[1, 2]], tockePoOboduSP[[1, 3]]}];
```

```
In[335]= tockePoOboduZG = Map[Append[ $\phi$ hPoOboduZG[[#]], radijZG[[#]] &, Range[Length[ $\phi$ hPoOboduZG]]];
AppendTo[tockePoOboduZG, {2  $\pi$ , tockePoOboduZG[[1, 2]], tockePoOboduZG[[1, 3]]}];
```

Določitev koordinat točk pri $\phi=0$ oz. $\phi=360$ (koordinate morajo biti iste!):

Določimo, na koliko točk (enakomerno razdeljenih po višini oboda) naj se višina oboda cilindra razdeli

```
In[337]= stTockVisina = 45;
```

Koordinate $\{\phi, z\}$ teh točk (točko, ki pripada spodnjemu robu in točko, ki pripada zg. robu izpustimo, saj smo jo določili že, ko smo določali koordinate točk po sp. in zg. robu oboda cilindra)

```
In[338]=  $\phi$ zPri $\phi$ 0oz2 $\pi$  = Rest[Most[Table[{0, h}, {h, zmin, zmax, (zmax - zmin) / stTockVisina}]]];
```

Seznam n bližnjih podanih koordinat oboda cilindra za vsako točko

```
In[339]= nBliznjihTock = 10;
```

```
In[340]= seznam $\phi$ 0oz2 $\pi$  = Map[Nearest[razvitValj[[All, 1 ;; 2]],  $\phi$ zPri $\phi$ 0oz2 $\pi$ [[#]], nBliznjihTock] &,
Range[Length[ $\phi$ zPri $\phi$ 0oz2 $\pi$ ]]];
```

Izračun radija za vsako točko

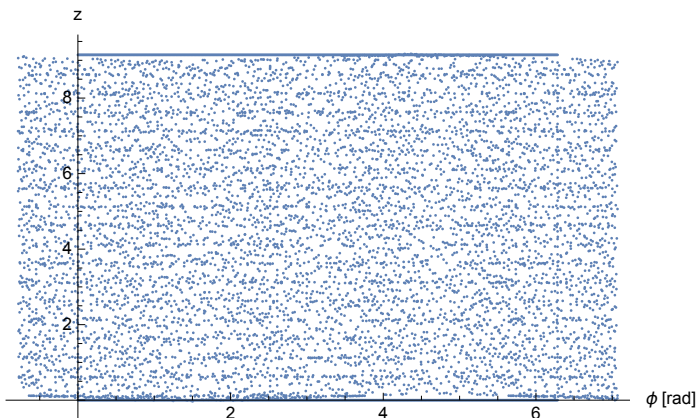
```
In[341]= radij $\phi$ 0oz2 $\pi$  = Table[Mean[razvitValj[[All, 3]]][[
Map[Position[razvitValj[[All, 1 ;; 2]], seznam $\phi$ 0oz2 $\pi$ [[i, #]]][[1, 1]] &,
Range[nBliznjihTock]]]], {i, Length[ $\phi$ zPri $\phi$ 0oz2 $\pi$ ]}];
```

```
In[342]= tockePri $\phi$ 0 = Map[Append[ $\phi$ zPri $\phi$ 0oz2 $\pi$ [[#]], radij $\phi$ 0oz2 $\pi$ [[#]]] &, Range[Length[ $\phi$ zPri $\phi$ 0oz2 $\pi$ ]]];
tockePri $\phi$ 2 $\pi$  = tockePri $\phi$ 0;
tockePri $\phi$ 2 $\pi$ [[All, 1]] = ConstantArray[2  $\pi$ , Length[tockePri $\phi$ 2 $\pi$ ]];
```

Podane koordinate razvitega cilindra razširimo z izračunanimi koordinatami po robovih plašča cilindra

```
In[345]= razvitValjRazsirjen =
Join[razvitValj, tockePoOboduSP, tockePoOboduZG, tockePri $\phi$ 0, tockePri $\phi$ 2 $\pi$ ];
```

```
In[346]= ListPlot[razvitValjRazsirjen[[All, 1 ;; 2]], AxesLabel -> {" $\phi$  [rad]", "z"}]
```

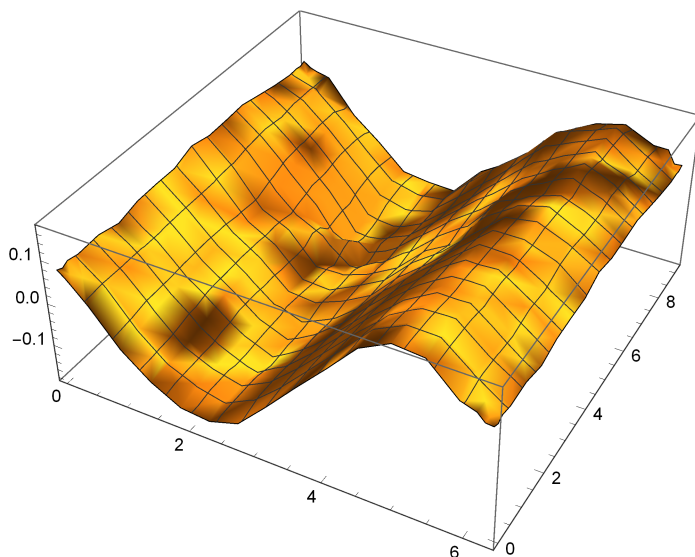


Interpolacija radija preko celotnega plašča cilindra

```
In[347]:= interpolacijaObodaValja = Interpolation[Transpose[{razvitValjRazsirjen[[All, 1 ;; 2]],
    razvitValjRazsirjen[[All, 3]] - PovprecenRadij}], InterpolationOrder -> 1]
```

InterpolatingFunction[  Domain: {{-0.784, 7.07}, {-0.027, 9.17}}
Output: scalar

```
In[348]:= Plot3D[interpolacijaObodaValja[phi, h], {phi, 0, 2 pi}, {h, zmin, zmax}]
```



Določitev koordinat strukturirane mreže, ki jo bomo podali v AceFEM

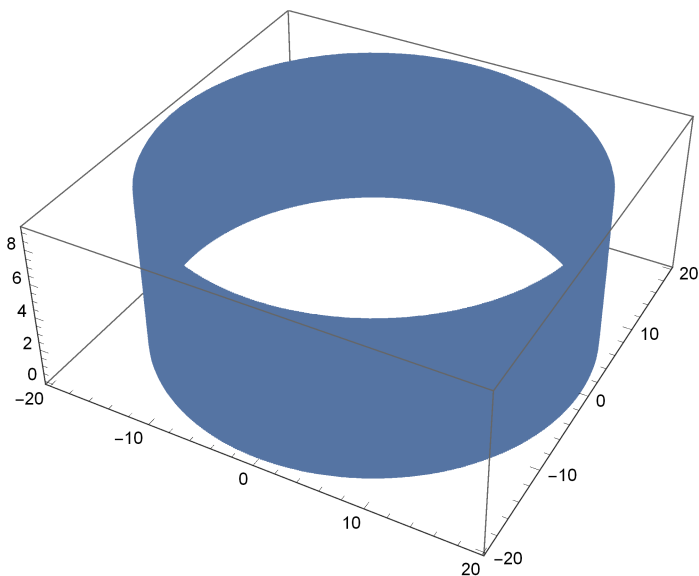
```
In[349]:= DeltaVisina = (9.144 - 9.05) / 8;
visina1 = 1.78 + DeltaVisina;
visina2 = visina1 + 1.78 + 2 * DeltaVisina;
visina3 = visina2 + 1.78 + 2 * DeltaVisina;
visina4 = visina3 + 1.78 + 2 * DeltaVisina;
stTockVisina1 = Round[visina1 / zmax * stTockVisina];
stTockVisina2do4 = Round[(visina2 - visina1) / zmax * stTockVisina];
stTockVisina5 = Round[(zmax - visina4) / zmax * stTockVisina];
```

```

In[357]:= PrviSegment = Table[N[{(interpolacijaObodaValja[φ, h] + PovprečenRadij) Cos[φ - 3 π / 4],
    (interpolacijaObodaValja[φ, h] + PovprečenRadij) Sin[φ - 3 π / 4], h}],
    {h, zmin, visina1, visina1 / stTockVisina1}, {φ, 0, 2 π, 2 π / stTockObod}];
DrugiSegment = Table[N[{(interpolacijaObodaValja[φ, h] + PovprečenRadij) Cos[φ - 3 π / 4],
    (interpolacijaObodaValja[φ, h] + PovprečenRadij) Sin[φ - 3 π / 4], h}],
    {h, visina1, visina2, (visina2 - visina1) / stTockVisina2do4}, {φ, 0, 2 π, 2 π / stTockObod}];
TretjiSegment = Table[N[{(interpolacijaObodaValja[φ, h] + PovprečenRadij) Cos[φ - 3 π / 4],
    (interpolacijaObodaValja[φ, h] + PovprečenRadij) Sin[φ - 3 π / 4], h}],
    {h, visina2, visina3, (visina3 - visina2) / stTockVisina2do4}, {φ, 0, 2 π, 2 π / stTockObod}];
CetrstiSegment = Table[N[{(interpolacijaObodaValja[φ, h] + PovprečenRadij) Cos[φ - 3 π / 4],
    (interpolacijaObodaValja[φ, h] + PovprečenRadij) Sin[φ - 3 π / 4], h}],
    {h, visina3, visina4, (visina4 - visina3) / stTockVisina2do4}, {φ, 0, 2 π, 2 π / stTockObod}];
PetiSegment = Table[N[{(interpolacijaObodaValja[φ, h] + PovprečenRadij) Cos[φ - 3 π / 4],
    (interpolacijaObodaValja[φ, h] + PovprečenRadij) Sin[φ - 3 π / 4], h}],
    {h, visina4, zmax, (zmax - visina4) / stTockVisina5}, {φ, 0, 2 π, 2 π / stTockObod}];

In[363]:= ZgRob = PetiSegment[[-1]]; (*točke zgornjega roba*)
SpRob = PrviSegment[[1]]; (*točke spodnjega roba*)
dejanskoStTockVisina = stTockVisina1 + 3 * stTockVisina2do4 + stTockVisina5;
StrukturiranaMreza =
    Flatten[Join[PrviSegment, DrugiSegment, TretjiSegment, CetrstiSegment, PetiSegment], 1];
Razdeljena = DeleteDuplicates[Partition[StrukturiranaMreza, stTockObod + 1]];
ListPointPlot3D[StrukturiranaMreza]

```



Podajanje mreže končnih elementov v AceFEM

```
In[423]= SMTInputData[];
SMTAddDomain[{"segment1", "BI:SEMSS1ESJCANSE4P6SVenantMises",
  {"E *" → 210 000 000, "σy *" → 243 000, "t *" → 0.0113, "ν *" → 0.3}}];
SMTAddDomain[{"segment2", "BI:SEMSS1ESJCANSE4P6SVenantMises",
  {"E *" → 210 000 000, "σy *" → 243 000, "t *" → 0.0090, "ν *" → 0.3}}];
SMTAddDomain[{"segment3", "BI:SEMSS1ESJCANSE4P6SVenantMises",
  {"E *" → 210 000 000, "σy *" → 243 000, "t *" → 0.0069, "ν *" → 0.3}}];
SMTAddDomain[{"segment4", "BI:SEMSS1ESJCANSE4P6SVenantMises",
  {"E *" → 210 000 000, "σy *" → 243 000, "t *" → 0.0055, "ν *" → 0.3}}];
SMTAddDomain[{"segment5", "BI:SEMSS1ESJCANSE4P6SVenantMises",
  {"E *" → 210 000 000, "σy *" → 243 000, "t *" → 0.0055, "ν *" → 0.3}}];
SMTMesh["segment1", "S1", {stTockObod, stTockVisina1}, PrviSegment];
SMTMesh["segment2", "S1", {stTockObod, stTockVisina2do4}, DrugiSegment];
SMTMesh["segment3", "S1", {stTockObod, stTockVisina2do4}, TretjiSegment];
SMTMesh["segment4", "S1", {stTockObod, stTockVisina2do4}, CetrtiSegment];
SMTMesh["segment5", "S1", {stTockObod, stTockVisina5}, PetiSegment];
SMTAddEssentialBoundary[{Line[SpRob], 1 → 0, 2 → 0, 3 → 0, 4 → 0, 5 → 0, 6 → 0}];
```

OJAČITVE

Spodnja ojacitev

```
In[115]= (*SpOjacitevCilinder={});
hSpOjacitev=5.10;
hSp=Nearest[StrukturiranaMreza[[All,3]],hSpOjacitev][[1]];
n=1;
While[n<=Length[StrukturiranaMreza],If[StrukturiranaMreza[[n,3]]==hSp,
  SpOjacitevCilinder=Append[SpOjacitevCilinder,StrukturiranaMreza[[n]],Nothing];
  n++];
SpOjacitevRob=
Table[{{(interpolacijaObodaValja[φ,hSp]+PovprecenRadij+0.16)*Cos[φ-zasukKonstrukcije],
  (interpolacijaObodaValja[φ,hSp]+PovprecenRadij+0.16)*Sin[φ-zasukKonstrukcije],
  hSp},{φ,0,2π,2π/stTockObod}}];
SpOjacitevPasnica=Table[{{(interpolacijaObodaValja[φ,hSp]+PovprecenRadij+0.16)*
  Cos[φ-zasukKonstrukcije],(interpolacijaObodaValja[φ,hSp]+PovprecenRadij+0.16)*
  Sin[φ-zasukKonstrukcije],hSp-0.065},{φ,0,2π,2π/stTockObod}}];
SMTAddDomain["ojacitevSpStojina","BI:SEMSS1ESHYANSE4P6SVenant",
  {"E *"→210000000,"t *"→0.0075}];
SMTAddDomain["ojacitevSpPasnica","BI:SEMSS1ESHYANSE4P6SVenant",
  {"E *"→210000000,"t *"→0.0105}];
SMTMesh["ojacitevSpStojina","S1",{stTockObod,2},{SpOjacitevCilinder,SpOjacitevRob}];
SMTMesh["ojacitevSpPasnica","S1",{stTockObod,2},{SpOjacitevRob,SpOjacitevPasnica}];*)
```

Zgornja ojacitev


```

(*ZgOjacitevCilinder={};
hZgOjacitev=7.07;
hZg=Nearest[StrukturiranaMreza[[All,3]],hZgOjacitev][[1]];
n=1;
While[n<=Length[StrukturiranaMreza],If[StrukturiranaMreza[[n,3]]==hZg,
  ZgOjacitevCilinder=Append[ZgOjacitevCilinder,StrukturiranaMreza[[n]]],Nothing];
n++];
ZgOjacitevRob=
Table[{(interpolacijaObodaValja[φ,hZg]+PovprečenRadij+0.16)*Cos[φ-zasukKonstrukcije],
(interpolacijaObodaValja[φ,hZg]+PovprečenRadij+0.16)*Sin[φ-zasukKonstrukcije],
hZg},{φ,0,2π,2π/stTockObod}];
ZgOjacitevPasnica=Table[{(interpolacijaObodaValja[φ,hZg]+PovprečenRadij+0.16)*
Cos[φ-zasukKonstrukcije],(interpolacijaObodaValja[φ,hZg]+PovprečenRadij+0.16)*
Sin[φ-zasukKonstrukcije],hZg-0.065},{φ,0,2π,2π/stTockObod}];
SMTAddDomain["ojacitevZgStojina","BI:SEMSS1ESHYANSE4P6SVenant",
{"E *"->210000000,"t *"->0.0075}];
SMTAddDomain["ojacitevZgPasnica","BI:SEMSS1ESHYANSE4P6SVenant",
{"E *"->210000000,"t *"->0.0105}];
SMTMesh["ojacitevZgStojina","S1",{stTockObod,2},{ZgOjacitevCilinder,ZgOjacitevRob}];
SMTMesh["ojacitevZgPasnica","S1",{stTockObod,2},{ZgOjacitevRob,ZgOjacitevPasnica}];*)

```

Streha

```

SMTAddDomain["streha","BI:SEMSS1ESHYANSE4P6SVenant",{E *"->10^9,"t *"->5}];
obrisStreha = Table[{(interpolacijaObodaValja[φ,zmax]+PovprečenRadij+0.2)*Cos[φ-3π/4],
(interpolacijaObodaValja[φ,zmax]+PovprečenRadij+0.2)*Sin[φ-3π/4],
zmax},{φ,0,2π,2π/stTockObod}];
SMTMesh["streha","S1",{stTockObod,2},{ZgRob,obrisStreha}];

```

Obtežba

In[438]= $\gamma g = 1.17$; (*v primeru polnega rezervoarja $\gamma g = 1.35$ *)

```

γq = 1.53;
ρw = 1;
g = 9.81;
q = 1.27;
str = -2.3;
r = Abs[Min[StrukturiranaMreza[[All,1]]]];
γ = 78;
cpi = -0.4;

```

Tekočina

```
In[447]= (*Map[SMTAddDistributedBoundary[#,1→Function[{X,Y,Z,nx,ny,nz},(zmax-Z)γg ρw g nx],
      2→Function[{X,Y,Z,nx,ny,nz},(zmax-Z)γg ρw g ny]]&,
      {"segment1","segment2","segment3","segment4","segment5"}];*)
```

Veter

```
In[448]= Map[SMTAddDistributedBoundary[
      {"X" <= -r Cos[π * 75 / 180] &, #}, 1 → Function[{X, Y, Z, nx, ny, nz},
      -q γq nx ((1 - 2.583 * Sin[1.183 * ArcCos[-X / r]] ^ 2) - cpi)], 2 → Function[
      {X, Y, Z, nx, ny, nz}, -q γq ny ((1 - 2.583 * Sin[1.183 * ArcCos[-X / r]] ^ 2) - cpi)]] &,
      {"segment1", "segment2", "segment3", "segment4", "segment5"}];
Map[SMTAddDistributedBoundary[{-r Cos[π * 75 / 180] < "X" < -r Cos[π * 105 / 180] &, #},
      1 → Function[{X, Y, Z, nx, ny, nz},
      -q γq nx ((1 - 2.583 * Sin[1.183 * ArcCos[-X / r]] ^ 2) * (0.93 + (1 - 0.93) *
      Cos[π / 2 * (ArcCos[-X / r] - π * 75 / 180) / (π * 105 / 180 - π * 75 / 180)]) - cpi)],
      2 → Function[{X, Y, Z, nx, ny, nz}, -q γq ny ((1 - 2.583 * Sin[1.183 * ArcCos[-X / r]] ^ 2) *
      (0.93 + (1 - 0.93) * Cos[π / 2 * (ArcCos[-X / r] - π * 75 / 180) /
      (π * 105 / 180 - π * 75 / 180)]) - cpi)]] &,
      {"segment1", "segment2", "segment3", "segment4", "segment5"}];
Map[SMTAddDistributedBoundary[{"X" >= -r Cos[π * 105 / 180] &, #},
      1 → Function[{X, Y, Z, nx, ny, nz},
      -q γq nx (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180] ^ 2) - cpi)], 2 → Function[
      {X, Y, Z, nx, ny, nz}, -q γq ny (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180] ^ 2) - cpi)]] &,
      {"segment1", "segment2", "segment3", "segment4", "segment5"}];
```

Veter območje 75 ° in 105 ° (zgornje formule ne zajamejo elemente v kotu 75° oz. 105°)

```
In[451]= X1 = Map[Nearest[Razdeljena[[#]][[ ; ; , 1]], -r Cos[π * 75 / 180], 4] &,
      Range[Length[Razdeljena]]];
```

```
In[452]= Pozicija = Partition[DeleteDuplicates[Flatten[Table[
      Position[Razdeljena, X1[[i]][[j]]], {i, 1, Length[Razdeljena]}, {j, 1, 4}], 2]], 4];
```

```
In[453]= Tocke75 = DeleteDuplicates[Flatten[Table[
      Razdeljena[[i]][[Pozicija[[i]][[j]][[2]]], {i, 1, Length[Razdeljena]}, {j, 1, 4}], 1]];
```

```
In[454]= Tocke75Poz =
      Flatten[Table[{Tocke75[[1 + 4 i]], Tocke75[[4 + 4 i]]}, {i, 0, dejanskoStTockVisina}], 1];
Tocke75Neg = Flatten[Table[{Tocke75[[2 + 4 i]], Tocke75[[3 + 4 i]]},
      {i, 0, dejanskoStTockVisina}], 1];
```

```
In[456]= X2 = Map[Nearest[Razdeljena[[#]][[ ; ; , 1]], -r Cos[π * 105 / 180], 4] &,
      Range[Length[Razdeljena]]];
```

```
In[457]= Pozicija2 = Partition[DeleteDuplicates[Flatten[Table[
      Position[Razdeljena, X2[[i]][[j]]], {i, 1, Length[Razdeljena]}, {j, 1, 4}], 2]], 4];
```

```

In[458]= Tockel105 = DeleteDuplicates[Flatten[Table[Razdeljena[[i]][[Pozicija2[[i]][[j]][[2]]]],
      {i, 1, Length[Razdeljena]}, {j, 1, 4}], 1]];

In[459]= Tockel105Poz =
      Flatten[Table[{Tockel105[[1 + 4 i]], Tockel105[[4 + 4 i]]}, {i, 0, dejanskoStTockVisina}], 1];
Tockel105Neg = Flatten[Table[{Tockel105[[2 + 4 i]], Tockel105[[3 + 4 i]]},
      {i, 0, dejanskoStTockVisina}], 1];

In[461]= Map[SMTAddNaturalBoundary[Point[Tocke75[[#]]], 1 → Null, 2 → Null] &,
      Range[Length[Tocke75]]];
Map[SMTAddNaturalBoundary[Point[Tocke75Poz[[#]]],
      1 → q γ q ((1 - 2.583 * Sin[1.183 * 75 π / 180]^2) - cpi) * PovprečenRadij *
      2 * π / stTockObod * zmax / stTockVisina * Cos[75 π / 180],
      2 → -q γ q ((1 - 2.583 * Sin[1.183 * 75 π / 180]^2) - cpi) * PovprečenRadij * 2 *
      π / stTockObod * zmax / stTockVisina * Sin[75 π / 180]] &, Range[Length[Tocke75Poz]]];
Map[SMTAddNaturalBoundary[Point[Tocke75Neg[[#]]],
      1 → q γ q ((1 - 2.583 * Sin[1.183 * 75 π / 180]^2) - cpi) * PovprečenRadij *
      2 * π / stTockObod * zmax / stTockVisina * Cos[75 π / 180],
      2 → q γ q ((1 - 2.583 * Sin[1.183 * 75 π / 180]^2) - cpi) * PovprečenRadij * 2 *
      π / stTockObod * zmax / stTockVisina * Sin[75 π / 180]] &, Range[Length[Tocke75Neg]]];

In[464]= Map[SMTAddNaturalBoundary[Point[Tockel105[[#]]], 1 → Null, 2 → Null] &,
      Range[Length[Tockel105]]];
Map[SMTAddNaturalBoundary[Point[Tockel105Poz[[#]]],
      1 → q γ q (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180]^2) - cpi) *
      PovprečenRadij * 2 * π / stTockObod * zmax / stTockVisina * Cos[105 π / 180],
      2 → -q γ q (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180]^2) - cpi) * PovprečenRadij * 2 *
      π / stTockObod * zmax / stTockVisina * Sin[105 π / 180]] &, Range[Length[Tockel105Poz]]];
Map[SMTAddNaturalBoundary[Point[Tockel105Neg[[#]]],
      1 → q γ q (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180]^2) - cpi) *
      PovprečenRadij * 2 * π / stTockObod * zmax / stTockVisina * Cos[105 π / 180],
      2 → q γ q (0.93 (1 - 2.583 * Sin[1.183 * π * 105 / 180]^2) - cpi) * PovprečenRadij * 2 *
      π / stTockObod * zmax / stTockVisina * Sin[105 π / 180]] &, Range[Length[Tockel105Neg]]];

```

Lastna teža

```

In[467]= SMTAddDistributedBoundary["segment1", 3 → -γ q γ * 0.0113];
SMTAddDistributedBoundary["segment2", 3 → -γ q γ * 0.0090];
SMTAddDistributedBoundary["segment3", 3 → -γ q γ * 0.0069];
SMTAddDistributedBoundary["segment4", 3 → -γ q γ * 0.0055];
SMTAddDistributedBoundary["segment5", 3 → -γ q γ * 0.0055];

```

Teža strehe

```

In[472]= SMTAddNaturalBoundary[Line[ZgRob], 3 → Line[{γ q str}]];

```

Analiza

```
In[473]= SMTAnalysis[];
```

Metoda ločne dolžine

Parametri analize:

```
In[474]= λtarget = 1;
λerror = 0.005;
sEstimated = SMTArcLengthSet["λTarget" → 1];
sMax = 100;
ΔsMax = sEstimated / 4;
s0 = sEstimated / 100;
ΔsMin = 0;
```

Ob analizi bojo odčitane napetosti σ_{xx} , ločna dolžina, LPF, pomiki in stevilo negativnih lastnih vrednosti

```
In[481]= graf = {};
(*točka na kateri bojo določeni poteka pomikov in napetosti σxx*)
tockaSpremljanjaRezultata =
{-19.631582285117222`, -0.9076228250394042`, 5.999916666666667`};
CollectResults[] := (
AppendTo[graf, {SMTPostData["Sxx", Point[tockaSpremljanjaRezultata]],
SMTRData["Parameter"], SMTRData["Multiplier"],
SMTPostData["u", Point[tockaSpremljanjaRezultata]], SMTIData["NegativePivots"]}]];
)
```

Analiza:

```

In[484]:= CollectResults[];
SMTNextStep["Δγ" → s0];
While[
  While[step =
    SMTConvergence[10-10, 50, {"Adaptive γ", 16, ΔsMin, ΔsMax, sMax}, "AlternativeTarget" →
      (SMTRData["Multiplier"] < λtarget &)], SMTArcLengthIteration[]];
  If[Not[step[[1]]]
    , CollectResults[];
    If[Abs[SMTRData["Multiplier"] - λtarget] / λtarget < λerror
      , SMTStatusReport[{"Target λ: ", λtarget, " calculated λ: ", SMTRData["Multiplier"]}];
      step[[3]] = False;
    ];
  ];
  If[step[[4]] === "MinBound", SMTStatusReport["Analyze"]; SMTStepBack[]];
  step[[3]]
  ,
  If[step[[1]], SMTStepBack[]; , SMTArcLengthNext[]];
  SMTNextStep["Δγ" → step[[2]]
];
In[408]:= SMTArcLengthFree[];

```

Newtonova iteracija

```

In[373]= λMax = 1; λ0 = λMax / 100; ΔλMin = λMax / 1000; ΔλMax = λMax / 10;
tolNR = 10^-12; maxNR = 15; targetNR = 8;
SMTNextStep["Δλ" → λ0];
While[
  While[
    step = SMTConvergence[tolNR, maxNR, {"Adaptive BC", targetNR, ΔλMin, ΔλMax, λMax}]
    , SMTNewtonIteration[];
  ];
  If[step[[4]] == "MinBound", SMTStatusReport["Analyze"];
    SMTStepBack[]];];
(*predlagana vrednost prirastka parametra je manjša kakor je določena spodnja meja*)
step[[3]]
, If[step[[1]], SMTStepBack[]];];
SMTNextStep["Δλ" → step[[2]]]
];

```