

SVEUČILIŠTE U RIJECI
TEHNIČKU FAKULTET
Sveučilišni diplomski studij strojarstva

Teorija strojeva i mehanizama
**PRORAČUN KOEFICIJENATA KRUTOSTI I PRIGUŠENJA
ČETVEROSEGMENTNOG ULJNOG KLIZNOG LEŽAJA U
PROGRAMSKOM JEZIKU PYTHON**

Mentor: Prof. dr. sc. Roberto Žigulić

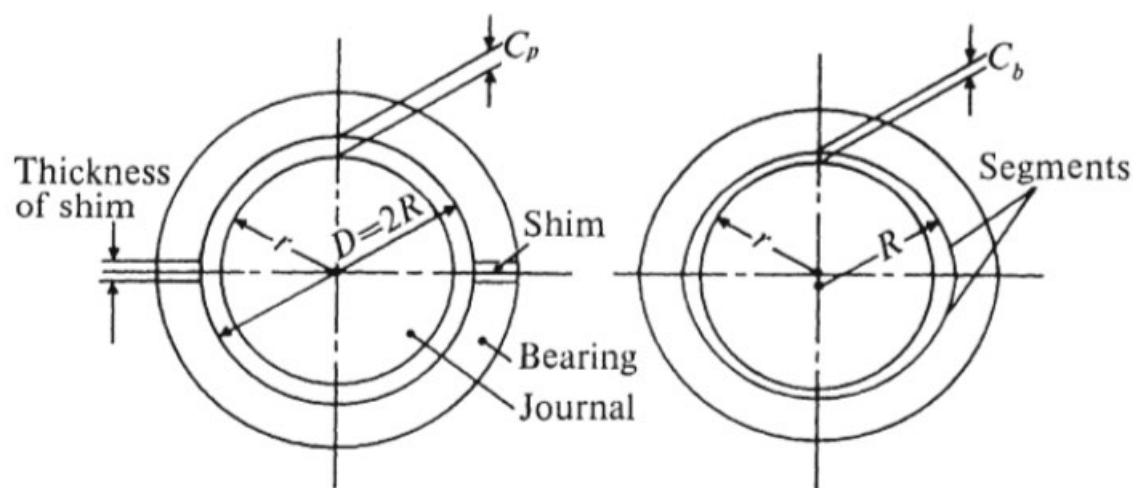
SADRŽAJ

1. UVOD	2
2. MATEMATIČKI MODEL	4
2.1. Definicija promatrane domene	4
2.2. Reynoldsove jednačbe	5
2.3. Rubni uvjeti	7
3. NUMERIČKO RJEŠAVANJE REYNOLDSOVE JEDNAČBE	8
3.1. Primjena metode konačnih razlika	8
4. IMPLEMENTACIJA RJEŠENJA U PROGRAMSKOM JEZIKU PYTHON	14
4.1. <i>Main</i>	14
4.2. <i>Steady</i>	18
4.3. Dinamičke funkcije (Dynamic_1 i Dynamic_2)	27
5. REZULTATI.....	35
6. ZAKLJUČAK	43

1. UVOD

Klizni ležajevi su ležajevi koji podupiru klizno gibanje vratila opterećenog aksijalnom silom. Klizni ležajevi mogu biti podijeljeni na hidrodinamičke ležajeve u kojima je tlak generiran relativnim gibanjem kliznih površina, hidrostatičke ležajeve u kojima je sila opterećenja vratila suprotstavljena zbog dotoka ulja vanjskom pompom kroz utore i hibridni ležajevi koji se služe hidrodinamičkim i hidrostatičkim tlakom. U procesu projektiranja ležajeva potrebno je pomno analizirati prijenos opterećenja i odrediti stabilnost sustava pod djelovanjem tih opterećenja. Ukoliko se u procesu projektiranja ne obrati dovoljna pozornost na ispitivanje ponašanja ležaja pod određenim opterećenjima postoji veća opasnost da će ležaj biti neispravan. Kako bi se izbjegle rekonstrukcije i skupe intervencije nastoji se predvidjeti gibanje i opterećenja osovine na ležaj. Sveukupni sustav vibrira te se javlja opterećenje koje se može prikazati pomoću polja tlakova. Fluid u ležaju djeluje na način da postepeno prigušuje vibracije koje nastaju, stoga se takvo ponašanje i svojstva ležaja mogu opisati pomoću osnovnih elemenata vibracijskog sustava tj. pomoću koeficijenata krutosti i prigušenja. Cilj ovog rada je objasniti proračun koeficijenata metodom konačnih razlika koja nam omogućuje da dobijemo relativno precizne rezultate s obzirom na broj iteracija i dopuštene pogreške.

Kao što je već spomenuto proračun se zasniva na postupku matematičkog modeliranja koji započinje određivanjem područja proračuna. Drugim riječima potrebno je definirati domenu na kojoj se može opisati raspojela fizikalnih veličina duž čitave konstrukcije. Matematički model koji je korišten bazira se na pojednostavljenim Reynoldsovim jednažbama koje su same po sebi pojednostavljene verzije Navier-Stokesovih jednažbi za turbulentno gibanje. Također potrebno je i ispravno odrediti rubne uvjete. Metodom konačnih razlika potrebno je prevesti diferencijalne jednažbe u sustave algebarskih jednažbi koje će naposljetku biti jednostavnije za riješiti. Proračun je preveden u programskom jeziku *Python*, te su dobiveni rezultati ispisani u obliku dijagrama kako bi se mogli što je moguće bolje prikazati.



Slika 1.1. Ležaj i osovina

2. MATEMATIČKI MODEL

Kao što je navedeno u uvodu ovog rada matematički model kojim se dobije polje tlakova, te naposljetku raspodjela opterećenja na domeni je modificirani oblik Navier-Stokesovih jednažbi tj. Reynoldsove jednažbe. S obzirom na promatrani režim strujanja fluida unutar ležaja razlikujemo Reynoldsove jednažbe za laminarno i turbulentno strujanje. U ovom slučaju koristimo se Reynoldsovim jednažbama koje modeliraju polje tlakova u turbulentnom režimu iz razloga jer očekujemo veće brzine vrtnje osovine.

2.1. Definicija promatrane domene

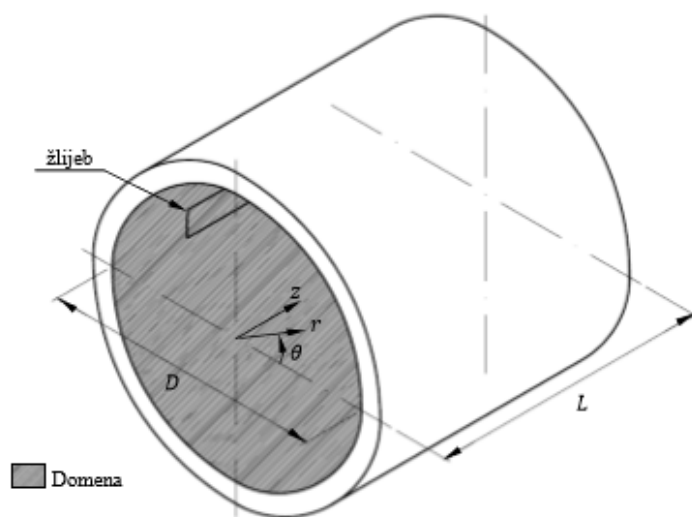
Primjena Meode konačnih razlika zahtjeva definiranje područja proračuna bolje poznatu kao domena. Diskretizacijom Reynoldsovih jednažbi dobit ćemo sustav algebarskih jednažbi koje će opisivati vrijednosti tlaka u svakoj točki na domeni. S obzirom da je za dobivanje traženih koeficijenta krutosti i prigušenja dovoljno poznavati raspodjelu tlakova između osovine i ležaja, promatrat ćemo samo domenu koju čini samo unutarnja ploha ležaja.

Područje proračuna je matematički definirano:

$$r = D/2,$$

$$0 \leq \theta \leq 360^\circ,$$

$$0 \leq z \leq L$$



Slika 1.2. Unutrašnjost ležaja

2.2. Reynoldsove jednačbe

Reynoldsova jednačba je parcijalna diferencijalna jednačba koja opisuje tlačno polje u tankom viskoznom sloju fluida. Kao što je već navedeno u ovom radu koristimo Reynoldsovu jednačbu u dvije dimenzije kako bi opisali polje tlakova između osovine i ležaja. Između osovine i ležaja nastaje tanki sloj ulja koji na različitim mjestima poprima različite vrijednosti tlaka.

Reynoldsova jednačba je oblika:

$$\frac{\partial}{\partial \theta} \left(h^3 * \frac{\partial p}{\partial \theta} \right) + \frac{\partial}{\partial z} \left(h^3 * \frac{\partial p}{\partial z} \right) = 6 * U * \eta * \frac{\partial h}{\partial \theta} \quad (2.1)$$

Gdje su:

θ, z - koordinate domene $[\circ], [m]$

h - debljina uljnog filma $[m]$

p – tlak $[Pa]$

U - periferna brzina osovine $[m/s]$

η - dinamički viskozitet $[m^2/s]$

Reynoldsova jednačba vrijedi ako se u obzir uzimaju sljedeći uvjeti:

- Fluid spada pod Newtonske fluide (fluid je viskozan)
- Viskozne sile u fluidu dominiraju nad silama inercije
- Varijacija tlaka kroz sloj fluida je zanemarivo malena ($\frac{\partial p}{\partial z} = 0$)
- Uljni film je puno manji od dužine i širine promatrane domene stoga se zanemaruju efekti zakrivljenosti domene
- Strujanje je laminarno
- Fluid je nestišljiv

Potrebno je prije numeričkog rješavanja Reynoldsove jednadžbe istu prevesti u bezdimenzijski oblik. Vršimo supstituciju $U = \omega * D/2$, te uvrštavanjem u početnu jednadžbu (2.1) dobivamo sređeni statički oblik Reynoldsove jednadžbe (2.2)

$$\frac{\partial}{\partial \theta} \left(h^3 * \frac{\partial p_0}{\partial \theta} \right) + \left(\frac{D}{L} \right)^2 * \frac{\partial}{\partial z} \left(h^3 * \frac{\partial p_0}{\partial z} \right) = 3 * \frac{\partial h}{\partial \theta} \quad (2.2)$$

Gdje su:

θ, z – bezdimenzijske koordinate domene [$^\circ$], [m]

h - bezdimenzijska debljina uljnog filma [m]

p – bezdimenzijski tlak [Pa]

D - promjer osovine [m]

L - duljina ležaja [m]

ω - kutna brzina osovine [rad/s]

Kako bi se u obzir uzimali perturbacijski efekti potrebno je računanje tlakova u dinamičkom polju, te dinamičkih koeficijenata. Zato se potrebe se koriste dinamički oblici Reynoldsove jednadžbe koji sadrže kompleksne brojeve.

$$\begin{aligned} \frac{\partial}{\partial \theta} \left(h^3 * \frac{\partial p_1}{\partial \theta} \right) + \left(\frac{D}{L} \right)^2 * \frac{\partial}{\partial z} \left(h^3 * \frac{\partial p_1}{\partial z} \right) + \frac{\partial}{\partial \theta} \left(3h^2 * \cos \theta * \frac{\partial p_0}{\partial \theta} \right) + \left(\frac{D}{L} \right)^2 \\ * \frac{\partial}{\partial z} \left(3h^2 * \cos \theta * \frac{\partial p_0}{\partial z} \right) = -3 * \sin \theta + i6 * \Omega * \cos \theta \quad (2.3) \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial \theta} \left(h^3 * \frac{\partial p_2}{\partial \theta} \right) + \left(\frac{D}{L} \right)^2 * \frac{\partial}{\partial z} \left(h^3 * \frac{\partial p_2}{\partial z} \right) + \frac{\partial}{\partial \theta} \left(3h^2 * \sin \theta * \frac{\partial p_0}{\partial \theta} \right) + \left(\frac{D}{L} \right)^2 \\ * \frac{\partial}{\partial z} \left(3h^2 * \sin \theta * \frac{\partial p_0}{\partial z} \right) = -3 * \cos \theta + i6 * \Omega * \sin \theta \quad (2.4) \end{aligned}$$

2.3. Rubni uvjeti

Kako bi matematički model bio u potpunosti definiran, potrebno je definirati rubne uvjete. Za ovaj problem postoje nekoliko Neumannovih i Dirichletovih rubnih uvjeta.

Tlak na krajevima ležaja jednak je nuli.

$$p = 0$$

$$r = D/2, \quad 0 \leq \theta < 360^\circ, \quad z = 0$$

$$r = D/2, \quad 0 \leq \theta < 360^\circ, \quad z = L$$

Promjena tlaka u radijalnom smjeru je nula, nema kaviacije u mazivu

$$\frac{\partial p}{\partial \theta} = 0$$

$$r = D/2, \quad 0 \leq \theta < 360^\circ, \quad 0 \leq z \leq L$$

Promjena tlaka u uzdužnom smjeru jednaka je nuli

$$\frac{\partial p}{\partial z} = 0$$

$$r = D/2, \quad 0 \leq \theta < 360^\circ, \quad z = L/2$$

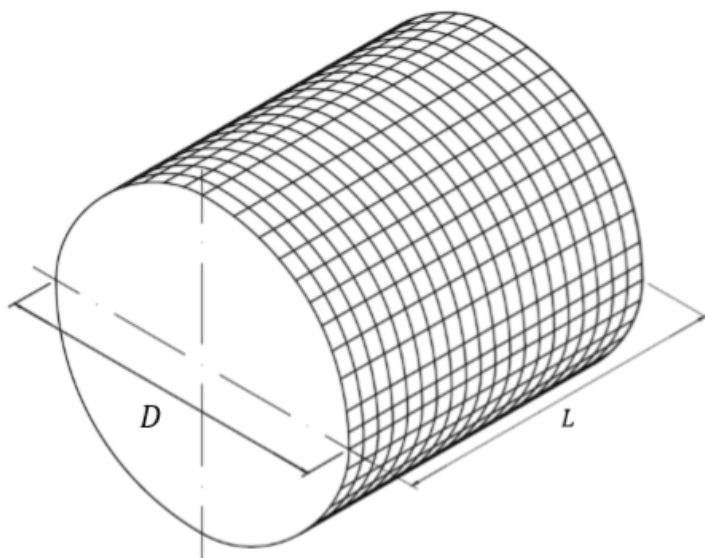
3. NUMERIČKO RJEŠAVANJE REYNOLDSOVE JEDNADŽBE

Kao što je slučaj sa mnogim drugim problemima u inženjerstvu, tako je i u ovom slučaju pomoću numeričkih metoda potrebno diskretizirati Reynoldsovu jednadžbu. Općenito matematički modeli kao što su Navier-Stokesove, Reynoldsove, Burgerove jednadžbe moguće riješiti i analitički ali samo u jednostavnim slučajevima u kojima se zanemaruje mnogo faktora. Iz navedenih razloga ovaj problem je rješavam numeričkom metodom konačnih razlika, te uz pomoć softverske podrške.

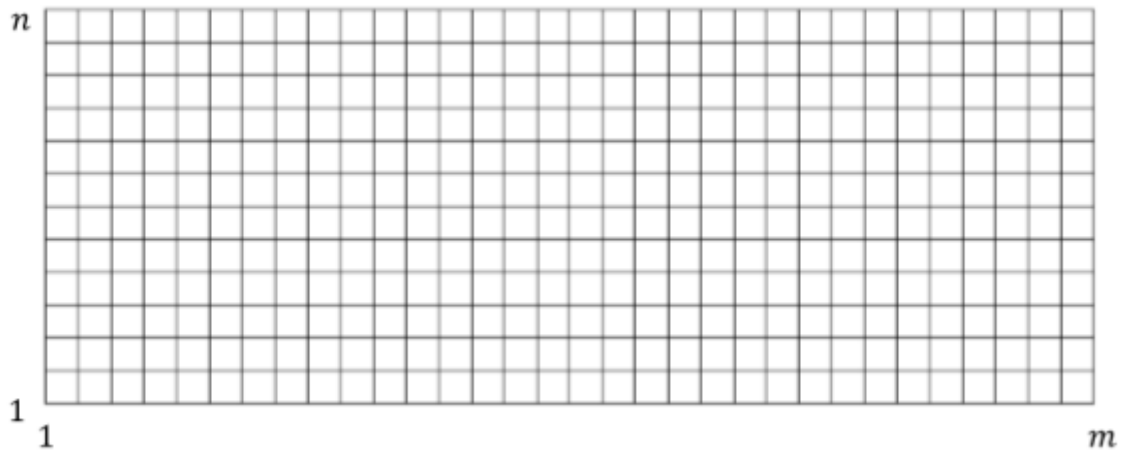
3.1. Primjena metode konačnih razlika

Odabir numeričke metode konačnih razlika omogućava nam pretvorbu diferencijalnih jednadžbi u sustav algebarskih jednadžbi. Rešavanje sustava algebarskih jednadžbi se nadalje može rješavati metodama za algebarsko računanje. Redukcija diferencijalnog problema u algebarski problem omogućava nam također korištenje softverske podrške.

Metoda konačnih razlika zahtjeva umrežavanje domene. Površina na unutrašnjem djelu ležaja zamjenjujemo mrežom diskretnih čvorova koji su određene gustoće. Gustoća mreže koju ćemo koristiti ovisi o željenoj preciznosti rezultat koje želimo postići i snazi računala kojim se koristimo. Na slikama 3.1. i 3.2. prikazan je izgled domene. Računanjem fizikalnih veličina i određivanjem njihovog odnosa sa veličinama susjednih čvorova moguće je aproksimirati njihovu raspodjelu po čitavoj domeni



Slika 3.1. Mreža konačnih razlika



Slika 3.2. Rastvorena mreža konačnih razlika

U podpoglavlju 2.2. naveli smo da zanemarujemo efekte zakrivljenja domene, stoga domenu možemo promatrati kao pravokutnik što nam uvelike smanjuje složenost programskog koda. Sljedeći je korak diskretizirati Reynoldsovu jednadžbu na sljedeći način.

$$3h^2 * \frac{\partial h}{\partial \theta} * \frac{\partial p_0}{\partial \theta} + h^3 \frac{\partial^2 p_0}{\partial \theta^2} + \left(\frac{D}{L}\right)^2 * h^3 * \frac{\partial^2 p_0}{\partial z^2} = 3 * \frac{\partial h}{\partial \theta} \quad (3.1)$$

$$\begin{aligned} \frac{3}{4} h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) * (p_{0\ i+1,j} - p_{0\ i-1,j}) + h_{i,j}^3 * (p_{0\ i+1,j} - 2 * p_{0\ i,j} + p_{0\ i-1,j}) + \alpha \\ * h_{i,j}^3 * (p_{0\ i,j} - 2 * p_{0\ i,j} + p_{0\ i,j-1}) = \frac{3}{2} \Delta \theta * (h_{i+1,j} - h_{i-1,j}) \end{aligned} \quad (3.2)$$

Za pojedini čvor na domeni vrijedi jednadžba (3.1) gdje je:

h - debljina uljnog filma [m]

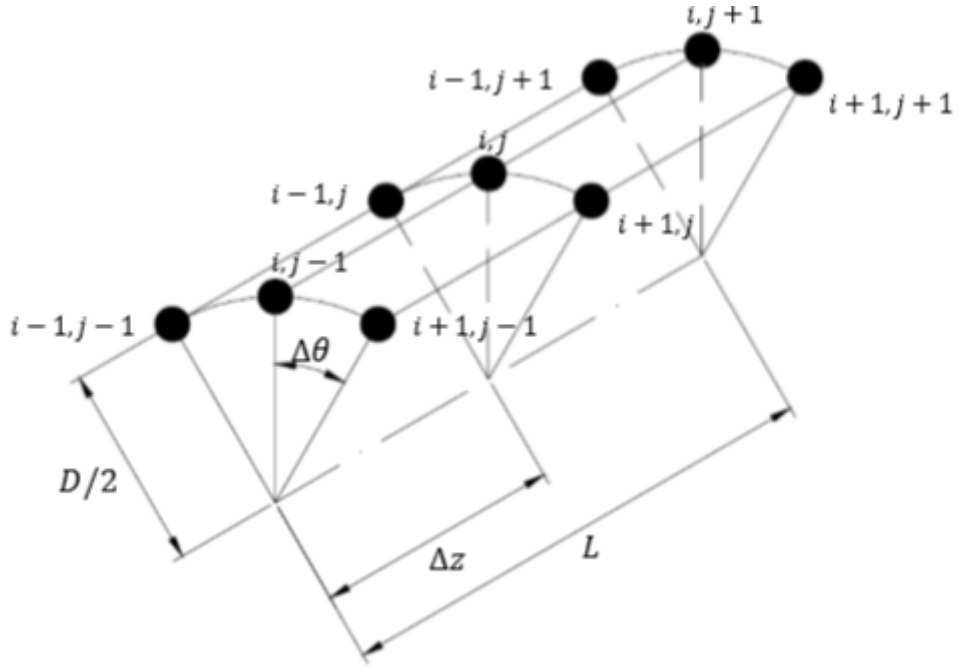
p_0 - statički tlak [Pa]

p_1 - komponenta dinamičkog tlaka [Pa]

$\Delta \theta$ - razlika između dva susjedna čvora u radijalnom smjeru

$$\alpha = \left(\frac{\Delta \theta}{\Delta z}\right)^2 * \left(\frac{D}{L}\right)^2$$

Indeksi i, j predstavljaju poziciju čvora na umreženoj domeni. Točke sa indeksima su prikazane na sljedećoj slici.



Slika 3.3. Čvorovi na domeni

Daljnijim sređivanjem izraza (3.1) moguće je dobiti:

$$p_{0\,i,j} = (A_1 * p_{0\,i+1,j} + A_2 * p_{0\,i-1,j} + A_3 * p_{0\,i,j+1} + A_4 * p_{0\,i,j-1} + A_5)/A_6 \quad (3.4)$$

$$A_1 = \frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.5)$$

$$A_2 = -\frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.6)$$

$$A_3 = \alpha * h_{i,j}^3 \quad (3.7)$$

$$A_4 = \alpha * h_{i,j}^3 \quad (3.8)$$

$$A_5 = \frac{3}{2}\Delta\theta * (h_{i+1,j} - h_{i-1,j}) \quad (3.9)$$

$$A_6 = 2h_{i,j}^3 * (1 + \alpha) \quad (3.10)$$

Dinamički oblici Reynoldsove jednačbe sadrže kompleksne izraze ali ih je također moguće dobiti diskretizacijom.

$$\begin{aligned}
& 3h^2 * \frac{\partial h}{\partial \theta} * \frac{\partial p_1}{\partial \theta} + h^3 * \frac{\partial^2 p_1}{\partial \theta^2} + \left(\frac{D}{L}\right)^2 * h^3 * \frac{\partial^2 p_1}{\partial z^2} + 3h^2 * \cos\theta * \frac{\partial^2 p_0}{\partial \theta^2} + 6h * \cos\theta * \frac{\partial h}{\partial \theta} \\
& * \frac{\partial p_0}{\partial \theta} - 3h^2 * \sin\theta * \frac{\partial p_0}{\partial \theta} + \left(\frac{D}{L}\right)^2 * \left(3h^2 * \cos\theta * \frac{\partial^2 p_0}{\partial z^2}\right) \\
& = -3 * \sin\theta + i6 * \Omega * \cos\theta \quad (3.11)
\end{aligned}$$

$$\begin{aligned}
& 3h^2 * \frac{\partial h}{\partial \theta} * \frac{\partial p_2}{\partial \theta} + h^3 * \frac{\partial^2 p_2}{\partial \theta^2} + \left(\frac{D}{L}\right)^2 * h^3 * \frac{\partial^2 p_2}{\partial z^2} + 3h^2 * \sin\theta * \frac{\partial^2 p_0}{\partial \theta^2} + 6h * \sin\theta * \frac{\partial h}{\partial \theta} \\
& * \frac{\partial p_0}{\partial \theta} - 3h^2 * \cos\theta * \frac{\partial p_0}{\partial \theta} + \left(\frac{D}{L}\right)^2 * \left(3h^2 * \sin\theta * \frac{\partial^2 p_0}{\partial z^2}\right) \\
& = -3 * \cos\theta + i6 * \Omega * \sin\theta \quad (3.12)
\end{aligned}$$

Metodom konačnih razlika dobijemo sljedeće jednačbe (3.13), (3.26)

$$\begin{aligned}
& \frac{3}{4} h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) * (p_{1\ i+1,j} - p_{1\ i-1,j}) + h_{i,j}^3 * (p_{1\ i+1,j} - 2 * p_{1\ i,j} + p_{1\ i-1,j}) + \alpha \\
& * h_{i,j}^3 * (p_{1\ i,j} - 2 * p_{1\ i,j} + p_{1\ i,j-1}) + 3h_{i,j}^2 * \cos\theta \\
& * (p_{0\ i+1,j} - 2 * p_{0\ i,j} + p_{0\ i-1,j}) + \frac{6}{4} h_{i,j} * \cos\theta * (h_{i+1,j} - h_{i-1,j}) \\
& * (p_{0\ i+1,j} - p_{0\ i-1,j}) - \frac{3}{2} h_{i,j}^2 * \sin\theta * \Delta\theta * (p_{0\ i+1,j} - p_{0\ i-1,j}) + 3 * \alpha \\
& * h_{i,j}^2 * \cos\theta * (p_{0\ i,j+1} - 2p_{0\ i,j} + p_{0\ i,j-1}) \\
& = \Delta\theta^2 * (-3\sin\theta + i6\Omega\cos\theta) \quad (3.13)
\end{aligned}$$

Gdje je:

$$\alpha = \left(\frac{\Delta\theta}{\Delta z}\right)^2 * \left(\frac{D}{L}\right)^2$$

Te se sređivanjem dobije:

$$\begin{aligned}
p_{1\ i,j} = & (A_1 * p_{1\ i+1,j} + A_2 * p_{1\ i-1,j} + A_3 * p_{1\ i,j+1} + A_4 * p_{1\ i,j-1} + A_5 * p_{0\ i,j} + A_6 \\
& * p_{0\ i+1,j} + A_7 * p_{0\ i-1,j} + A_8 * p_{0\ i,j+1} + A_9 * p_{0\ i,j-1} + A_{10})/A_{11} \quad (3.14)
\end{aligned}$$

Gdje su:

$$A_1 = \frac{3}{4} h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.15)$$

$$A_2 = -\frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.16)$$

$$A_3 = \alpha * h_{i,j}^3 \quad (3.17)$$

$$A_4 = \alpha * h_{i,j}^3 \quad (3.18)$$

$$A_5 = -6\cos\theta * h_{i,j}^2 * (1 + \alpha) \quad (3.19)$$

$$A_6 = 3h_{i,j}^2 * \cos\theta + \frac{6}{4}h_{i,j}^2 * \cos\theta * (h_{i+1,j} - h_{i-1,j}) - \frac{3}{2}h_{i,j}^2 * \sin\theta * \Delta\theta \quad (3.20)$$

$$A_7 = 3h_{i,j}^2 * \cos\theta - \frac{6}{4}h_{i,j}^2 * \cos\theta * (h_{i+1,j} - h_{i-1,j}) + \frac{3}{2}h_{i,j}^2 * \sin\theta * \Delta\theta \quad (3.21)$$

$$A_8 = 3\alpha * h_{i,j}^2 * \cos\theta \quad (3.22)$$

$$A_9 = 3\alpha * h_{i,j}^2 * \cos\theta \quad (3.23)$$

$$A_{10} = -\Delta\theta^2(-3\sin\theta + i6 * \Omega * \cos\theta) \quad (3.24)$$

$$A_{11} = 2h_{i,j}^3 * (1 + \alpha) \quad (3.25)$$

Oblik druge jednadžbe je sljedeći:

$$\begin{aligned} & \frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) * (p_{2\ i+1,j} - p_{2\ i-1,j}) + h_{i,j}^3 * (p_{2\ i+1,j} - 2 * p_{2\ i,j} + p_{2\ i-1,j}) + \alpha \\ & * h_{i,j}^3 * (p_{2\ i,j+1} - 2 * p_{2\ i,j} + p_{2\ i,j-1}) + 3h_{i,j}^2 * \sin\theta \\ & * (p_{0\ i+1,j} - 2 * p_{0\ i,j} + p_{0\ i-1,j}) + \frac{6}{4}h_{i,j} * \sin\theta * (h_{i+1,j} - h_{i-1,j}) \\ & * (p_{0\ i+1,j} - p_{0\ i-1,j}) - \frac{3}{2}h_{i,j}^2 * \cos\theta * \Delta\theta * (p_{0\ i+1,j} - p_{0\ i-1,j}) + 3 * \alpha \\ & * h_{i,j}^2 * \sin\theta * (p_{0\ i,j+1} - 2p_{0\ i,j} + p_{0\ i,j-1}) \\ & = \Delta\theta^2 * (3\cos\theta + i6\Omega\sin\theta) \quad (3.26) \end{aligned}$$

Gdje je:

$$\alpha = \left(\frac{\Delta\theta}{\Delta z}\right)^2 * \left(\frac{D}{L}\right)^2$$

Sređivanjem izraza (3.26) se dobije:

$$\begin{aligned} p_{1\ i,j} = & (A_1 * p_{2\ i+1,j} + A_2 * p_{2\ i-1,j} + A_3 * p_{2\ i,j+1} + A_4 * p_{2\ i,j-1} + A_5 * p_{0\ i,j} + A_6 \\ & * p_{0\ i+1,j} + A_7 * p_{0\ i-1,j} + A_8 * p_{0\ i,j+1} + A_9 * p_{0\ i,j-1} + A_{10})/A_{11} \quad (3.27) \end{aligned}$$

Gdje su:

$$A_1 = \frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.28)$$

$$A_2 = -\frac{3}{4}h_{i,j}^2 * (h_{i+1,j} - h_{i-1,j}) + h_{i,j}^3 \quad (3.29)$$

$$A_3 = \alpha * h_{i,j}^3 \quad (3.30)$$

$$A_4 = \alpha * h_{i,j}^3 \quad (3.31)$$

$$A_5 = -6\sin\theta * h_{i,j}^2 * (1 + \alpha) \quad (3.32)$$

$$A_6 = 3h_{i,j}^2 * \sin\theta + \frac{6}{4}h_{i,j}^2 * \sin\theta * (h_{i+1,j} - h_{i-1,j}) + \frac{3}{2}h_{i,j}^2 * \cos\theta * \Delta\theta \quad (3.33)$$

$$A_7 = 3h_{i,j}^2 * \sin\theta - \frac{6}{4}h_{i,j}^2 * \sin\theta * (h_{i+1,j} - h_{i-1,j}) - \frac{3}{2}h_{i,j}^2 * \cos\theta * \Delta\theta \quad (3.34)$$

$$A_8 = 3\alpha * h_{i,j}^2 * \sin\theta \quad (3.35)$$

$$A_9 = 3\alpha * h_{i,j}^2 * \cos\theta \quad (3.36)$$

$$A_{10} = -\Delta\theta^2(3\cos\theta + i6 * \Omega * \sin\theta) \quad (3.37)$$

$$A_{11} = 2h_{i,j}^3 * (1 + \alpha) \quad (3.38)$$

4. IMPLEMENTACIJA RJEŠENJA U PROGRAMSKOM JEZIKU PYTHON

Kod rješavanja numeričkih problema poput ovog od izuzetne je važnosti imati pristup računalu. Računalo nam omogućava rješavanje velikog broja algebarskih izraza u kratkom vremenu, koje bi u drugim uvjetima bilo nemoguće. Za rješavanje ovog zadatka korišten je programski jezik *Python* sadržan u paketu *Spyder*.

Zadatak je u *Pythonu* implementiran pomoću četiri datoteke *Main*, *Steady*, *Dynamic_1* i *Dynamic_2*. Korisniku je dovoljno pokretanje programa sadržanog u datoteci *Main* nakon što je definirao početne uvjete.

4.1. *Main*

Datotekom *Main* pokreće se rješavanje Reynoldsove jednadžbe uz postupak unosa vrijujednosti određenih parametara, definicijom varijabli i konstanti. Datoteka *Main* poziva ostale funkcije *Steady*, *Dynamic_1* i *Dynamic_2*, te se vrši ispis traženih podataka i crtanje dijagrama.

U kodu datoteke *Main* pojavljuju se varijable koje predstavljaju fizikalne veličine ili parametre, te ih je potrebno objasniti s obzirom da u samom kodu ne postoji objašnjenje svake pojedine varijable.

Varijable:

lod – Omjer promjera osovine i duljine ležaja, D/L

e – Relativni ekscentritet, $\varepsilon = e/D$

beta – Polovica pripadnog kuta žlijeba, β

len1 – Polovica pripadne duljine žlijeba, l

ps – Dobavni tlak, p_s

SN - Sommerfeldov broj, Sn

phi_d – Kut pomaka osovine u stupnjevima, α

phi_r – Kut pomaka osovine u radijanima, α

ω – Omjer kutne brzine i frekvencije uznemiravanja, Ω

m, n – Broj čvorova na kružnoj, odnosno uzdužnoj osi, m, n

K – Koeficijenti krutosti, K

C – Koeficijenti prigušenja, C

Na početku definirani su pozvani moduli koji su potrebni za funkcioniranje koda. Pozvane funkcije su `numpy`, `matplotlib.pyplot`, `steady6`, `d14segment`, `d24segment`.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from steady6 import function_steady6 #Steady
4 from d14segment import function_d14segment #Dynamic_1
5 from d24segment import function_d24segment #Dynamic_2
```

Nakon pozivanja potrebnih modula vrši se unos potrebnih varijabli, potrebno je promijeniti vrijednosti varijabli s obzirom na parametre promatranog uljnog ležaja.

```
beta = 5*np.pi/180
len1 = 0.5
ps = 10**(-10)
lod = 1
omega = 1
```

Kako bi se dobile vrijednosti koeficijenata krutosti i koeficijenata prigušenja potrebno je zadati niz vrijednosti relativnih ekscentriteta. Relativni ekscentriteti su zadani u obliku niza vrijednosti kako bi se za svaku određenu vrijednost odredile potrebne sile i polje tlakova. Također ovaj način zadavanja podataka pogodan je za uštedu memorije. Program također može uspoređivati dobivene vrijednosti sa standardnim tabličnim vrijednostima, takve vrijednosti su definirane u programu ali se nalaze pod kometarima.

```
e = np.array([0.103, 0.15, 0.224, 0.352, 0.46, 0.559, 0.65, 0.734, 0.773,
0.793, 0.811, 0.883])
"""ATT = np.array([75.99, 70.58, 63.54, 55.41, 49.27, 44.33, 39.72, 35.16,
32.82, 31.62, 30.39, 25.02])
SOM = np.array([1.47, 0.991, 0.635, 0.358, 0.235, 0.159, 0.108, 0.071, 0.056,
0.05, 0.044, 0.024])
Kxx = np.array([1.53, 1.56, 1.62, 1.95, 2.19, 2.73, 3.45, 4.49, 5.23, 5.69,
6.22, 9.77,])
Kxy = np.array([10.14, 7.29, 5.33, 3.94, 3.57, 3.36, 3.34, 3.5, 3.65, 3.75,
3.88, 4.69])
Kyx = np.array([-3.01, -2.16, -1.57, -0.97, -0.8, -0.48, -0.23, 0.03, 0.18,
0.26, 0.35, 0.83])
```



```
Kyy = np.array([1.5, 1.52, 1.56, 1.48, 1.55, 1.48, 1.44, 1.44, 1.45, 1.45, 1.46, 1.53])
Cxx = np.array([20.34, 14.66, 10.8, 8.02, 7.36, 6.94, 6.89, 7.15, 7.42, 7.6, 7.81, 9.17])
Cxy = np.array([1.53, 1.58, 1.7, 1.63, 1.89, 1.78, 1.72, 1.7, 1.71, 1.71, 1.72, 1.78])
Cyx = np.array([1.53, 1.58, 1.7, 1.63, 1.89, 1.78, 1.72, 1.7, 1.71, 1.71, 1.72, 1.78])
Cyy = np.array([6.15, 4.49, 3.41, 2.37, 2.19, 1.74, 1.43, 1.2, 1.1, 1.06, 1.01, 0.83]) """
```

Sljedeće parametre koje je potrebno definirati su brojevi čvorova na kružnoj i uzdužnoj osi.

Pomoću ovih parametara definiramo dimenzije mreže na 3.1., 3.2.

```
""" mesh """
m = 61
n = 21
```

Dobivene rezultate Sommerfeldovog broja, kuta pomaka osovine i koeficijente krutosti i prigušenja potrebno je spremati u prazne nizove iz tog razloga je prvo potrebno kreirati nizove s nulama. Na ovaj način se definiraju dimenzije varijabli.

```
""" kreiranje niza s nulama """
SN = np.zeros(np.size(e))
phi_d = np.zeros(np.size(e))
phi_r = np.zeros(np.size(e))
K_xx = np.zeros(np.size(e))
K_xy = np.zeros(np.size(e))
K_yx = np.zeros(np.size(e))
K_yy = np.zeros(np.size(e))
C_xx = np.zeros(np.size(e))
C_xy = np.zeros(np.size(e))
C_yx = np.zeros(np.size(e))
C_yy = np.zeros(np.size(e))
```

Nakon definiranja početnih parametara pokreće se glavna petlja. U glavnoj petlji se provodi proračun za svaku vrijednost relativnog ekscentriteta u nizu e. U glavnoj petlji se pozivaju funkcije function_steady6, function_d14segment, function_d24segment. U nastavku se računaju koeficijenti krutosti, prigušenja i Sommerfeldovi brojevi.

```
""" main loop """
for i in range(np.size(e)):
    epsi = e[i]

    kut, hpot1, theta2, theta1, theta4, theta3, sirin, hpot, theta, p, Fx, Fy, W,
    phi_r[i] = function_steady6(m, n, epsi, lod, beta, len1, ps)
    print('hpot1 = %.4f' % hpot1[i])
    print('hpot = %.4f' % hpot[i])
```

```

print('theta = %.4f' %theta[i])
print('att = %.4f' %phi_r[i])
WXD1,WZD1 = function_d14segment(m, n, epsi, lod, beta, len1, ps, omega,
p, phi_r[i])
WXD2,WZD2 = function_d24segment(m, n, epsi, lod, beta, len1, ps, omega,
p, phi_r[i])
    """figure"""
    """surf(p)"""
    SN[i] = 1/(6*Fx)
    phi_d[i] = phi_r[i]*180/np.pi
    K_xx[i] = -WXD1.real/Fx
    K_yx[i] = -WZD1.real/Fx
    K_xy[i] = -WXD2.real/Fx
    K_yy[i] = -WZD2.real/Fx
    C_xx[i] = -WXD1.imag/Fx/omega
    C_yx[i] = -WZD1.imag/Fx/omega
    C_xy[i] = -WXD2.imag/Fx/omega
    C_yy[i] = -WZD2.imag/Fx/omega

print('eccentricity ratio = %.3f' %epsi)
print(' ')
print('attitude angle = %.4f' %phi_d[i])
print(' ')
print('sommerfeled NO. = %.4f' %SN[i])
print(' ')
print('Fx = ', Fx)
print('Fy = ', Fy)
print(' ')
print('K_xx = %.4f ; K_xy = %.4f' %(K_xx[i],K_xy[i]))
print('K_yx = %.4f ; K_yy = %.4f' %(K_yx[i],K_yy[i]))
print('C_xx = %.4f ; C_xy = %.4f' %(C_xx[i],C_xy[i]))
print('C_yx = %.4f ; C_yy = %.4f' %(C_yx[i],C_yy[i]))
print('*****')

```

Dobivene vrijednosti se ispisuju, te se u nastavku koda *Main* crtaju dijagrami kako bi se rezultati bolje vizualizirali.

4.2. *Steady*

Za rješavanje statičkog dijela proračuna koristimo program u datoteci *Steady*. Kako bi u potpunosti opisali ponašanje uljnog filma odnosno tlakove i koeficijente prigušenja i krutosti koje se javljaju moramo uzeti u obzir situaciju kada je fluid opterećen statičkim opterećenjem. Funkcije u datoteci *Steady* se zasnivaju na rješavanju statičkog oblika Reynoldsove jednačbe (, odnosno njenog algebarskog oblika. Datoteka *Main* u glavnoj petlji poziva funkciju `function_steady6`. Funkcionalnost ove funkcije je opisana u nastavku.

Potrebno je na početku objasniti nazive varijabli koje predstavljaju fizikalne veličine i parametre korištene u datoteci *Steady*.

`epsi` – Promatrani relativni ekscentritet, ε

`groovx` – Polovica pripadnog kuta žlijeba, β

`groovz` – Polovica pripadne duljine žlijeba, l

`er1`, `er2` – Razlike tlakova i kutova između dviju iteracija

`att` – Kut pomaka osovine

`n1`, `n2` – Brojači iteracija

`ORF`, `URF` – Nadrelaksacijski i podrelaksacijski faktori

`DTH`, `DZ` – Udaljenost između dvaju čvora u radijalnom i uzdužnom smjeru

`Switch1` – Matrica položaja žljebova

`p`, `h` – Statički tlak i debljina uljnog filma, p, h

`ig1`, `ig2` – Početna i krajnja pozicija prvog žlijeba na radijalnoj osi

`ig3`, `ig4` – Početna i krajnja pozicija drugog žlijeba na radijalnoj osi

`ig3`, `ig4` – Početna i krajnja pozicija trećeg žlijeba na radijalnoj osi

`ig3`, `ig4` – Početna i krajnja pozicija četvrtog žlijeba na radijalnoj osi

`jg1`, `jg2` – Početna i krajnja pozicija žljebova na uzdužnoj osi

ig11, ig21 – Početna i krajnja pozicija prvog dijela trećeg segmenta na radijalnoj osi

ig31, ig41 – Početna i krajnja pozicija četvrtog segmenta na radijalnoj osi

ig51, ig61 – Početna i krajnja pozicija prvog segmenta na radijalnoj osi

ig71, ig81 – Početna i krajnja pozicija drugog segmenta na radijalnoj osi

ig81, ig91 – Početna i krajnja pozicija drugog dijela trećeg segmenta na radijalnoj osi

psizrac – Omjer zračnosti, ψ

Modul koji se poziva u ovoj datoteci je modul *Numpy* koji nam omogućava rješavanje raznih matematičkih operacija

```
import numpy as np
```

Sljedeći korak je definicija funkcije `function_steady6`. Funkcija zahtijeva unos sljedećih parametara.

m, n – Broj čvorova na kružnoj i uzdužnoj osi, m, n

epsi – Promatrani relativni ekscentritet, ϵ

lod – Omjer promjera osovine i duljine ležaja, D/L

groovx – Polovica pripadnog kuta žlijeba, β

groovz – Polovica pripadne duljine žlijeba, l

ps – Dobavni tlak, p_s

```
def function_steady6(m, n, epsi, lod, groovx, groovz, ps):
```

U definiranoj funkciji potrebno je definirati veličine matrica i nizova.

```
er2 = 1
n2 = 0
att = 0.01
att1 = 0.01
ORF = 1.8
URF = 0.6
DTH = 2.0*np.pi/(m-1)
DZ = 1.0/(n-1)
"""Naredba switch1 služi kako bi program
kasnije mogao prepoznati na kojim se mjestima nalaze utori"""
```

```

switch1 = np.zeros([m+1,n+1])
p = np.zeros([m+1,n+1])
h = np.zeros([m+1,n+1])
h1 = np.zeros([m+1,n+1])
hpot = np.zeros([m+1])
hpot1 = np.zeros([m+1])
theta = np.zeros([m+1])
hpot2 = np.zeros([m+1])
theta1 = np.zeros([m+1])
theta2 = np.zeros([m+1])
theta3 = np.zeros([m+1])
theta4 = np.zeros([m+1])
kut = np.zeros([m+1])
sirin = np.zeros([n+1])

```

Proračun treba uzeti u obzir utjecaj dovedenog ulja pomoću žljebova kako bi se mogao opisati njihov utjecaj na raspodjelu tlakova. Potrebno je odrediti koordinate točaka žljebova na mreži domene.

```

i1 = round(groovx/DTH)
j1 = round(groovz/DZ)

ig1 = int(((m-1)*0.67/4)-i1) #8
ig2 = int(((m-1)*0.67/4)+i1) #12
ig3 = int(((m-1)*1.67/4)-i1) #23
ig4 = int(((m-1)*1.67/4)+i1) #27
ig5 = int(((m-1)*2.67/4)-i1) #38
ig6 = int(((m-1)*2.67/4)+i1) #42
ig7 = int(((m-1)*3.67/4)-i1) #53
ig8 = int(((m-1)*3.67/4)+i1) #57
jg1 = int(((n-1)/2)-j1)
jg2 = int(((n-1)/2)+j1)

```

Kako bi proračun uzimao u obzir da se radi o četverosegmentnom uljnom ležaju sa četiri utora za dovod ulja, potrebno je pomoću definiranih koordinata definirati položaje žljebova. Matrica `switch1` koja je preslika umrežene domene omogućava programu da prepozna koordinate na kojima se nalaze žljebovi. U matrici statičkog tlaka se na tim mjestima dodaje vrijednost dobavnog tlaka.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Slika 4.1. Vizualizacija dijela matrice switch1

```

"""first groove"""
for J in range(jg1,jg2+1):
    for I in range(ig1,ig2+1):
        p[I,J] = ps
        switch1[I,J] = 1

"""second groove"""
for J in range(jg1,jg2+1):
    for I in range(ig3,ig4+1):
        p[I,J] = ps
        switch1[I,J] = 1

"""third groove"""
for J in range(jg1,jg2+1):
    for I in range(ig5,ig6+1):
        p[I,J] = ps
        switch1[I,J] = 1

"""fourth groove"""
for J in range(jg1,jg2+1):
    for I in range(ig7,ig8+1):
        p[I,J] = ps
        switch1[I,J] = 1

```

Postupak pronalaženja rješenja diferencijalne jednačbe je iterativan, stoga proračun mora biti postavljen u petlju. Korištena petlja je `while` petlja koja se zaustavlja kada se zadovolje oba

zadana uvjeta. Petlja se zaustavlja kada pogreška prijeđe definiranu najmanju preciznost i kada je broj iteracija veći od 1000.

```
while (er2>Accuracy) and (n2<10**3):
```

U nastavku potrebno je definirati formule kutova za definiciju pojedinih segmenata.

$$\varepsilon_1 = \sqrt{\varepsilon^2 + \delta^2 + 2 * \varepsilon * \delta * \cos\phi} \quad (4.1)$$

$$\varepsilon_2 = \sqrt{\varepsilon^2 + \delta^2 - 2 * \varepsilon * \delta * \cos\left(\frac{\phi}{2} + \phi\right)} \quad (4.2)$$

$$\varepsilon_3 = \sqrt{\varepsilon^2 + \delta^2 - 2 * \varepsilon * \delta * \cos\phi} \quad (4.3)$$

$$\varepsilon_4 = \sqrt{\varepsilon^2 + \delta^2 - 2 * \varepsilon * \delta * \cos\left(\frac{\phi}{2} - \phi\right)} \quad (4.4)$$

$$\arg1 = \frac{\varepsilon * \sin\phi}{\delta + \varepsilon * \cos\phi} \quad (4.5)$$

$$\arg2 = \frac{\delta + \varepsilon * \sin\phi}{\varepsilon * \cos\phi} \quad (4.6)$$

$$\arg3 = \frac{\varepsilon * \sin\phi}{-\delta + \varepsilon * \cos\phi} \quad (4.7)$$

$$\arg1 = \frac{-\delta + \varepsilon * \sin\phi}{\varepsilon * \cos\phi} \quad (4.8)$$

$$\phi_1 = \tan^{-1} \arg1 \quad (4.9)$$

$$\phi_2 = \tan^{-1} \arg2 \quad (4.10)$$

$$\phi_3 = \pi + \tan^{-1} \arg3 \quad (4.11)$$

$$\phi_4 = 2\pi + \tan^{-1} \arg4 \quad (4.12)$$

Gdje su:

ε –Relativni ekscentritet

ϕ - Kut relativnog ekscentriteta od vertikalne osi

```

eps1=np.sqrt(eps1**2+deltazrac**2+2*eps1*deltazrac*np.cos(att))
arg1=eps1*np.sin(att)/(deltazrac+eps1*np.cos(att))
fi1=np.arctan(arg1)

eps2=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(np.pi/2+att))
arg2=(deltazrac+eps1*np.sin(att))/(eps1*np.sin(att))
fi2=np.arctan(arg2)

eps3=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(att))
arg3=eps1*np.sin(att)/(-deltazrac+eps1*np.cos(att))
fi3=np.pi+np.arctan(arg3)

eps4=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(np.pi/2-att))
arg4=(-deltazrac+eps1*np.sin(att))/(eps1*np.sin(att))
fi4=np.arctan(arg4)

```

Debljine uljnog filma u pojedinom segmentu se računaju pomoću sljedećih izraza:

$$h_1 = 1 + (\psi - 1) + \varepsilon_1 * \cos(\phi - \phi_1 + \theta) \quad (4.13)$$

$$h_2 = 1 + (\psi - 1) + \varepsilon_2 * \cos(\phi - \phi_2 + \theta) \quad (4.14)$$

$$h_3 = 1 + (\psi - 1) + \varepsilon_3 * \cos(\phi - \phi_3 + \theta) \quad (4.15)$$

$$h_4 = 1 + (\psi - 1) + \varepsilon_4 * \cos(\phi - \phi_4 + \theta) \quad (4.16)$$

Gdje su:

ψ - Omjer zračnosti, $\psi = \frac{c_b}{c_v}$

θ – Kut između promatrane osi i kuta pomaka osovine

Debljine uljnog filma računaju se uzimajući u obzir segment u kojem se nalazi žljeb.

```

"""segment broj 3 - prvi dio"""
for I in range(ig11,ig21):#1-8
    x = (I)*DTH
    theta3[I] = -att + x
    hpot1[I] = 1+(psizrac-1)+eps3*np.cos(theta3[I]+ att-fi3)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps3*np.cos(theta3[I]+att-fi3)

"""segment broj 4"""
for I in range(ig31,ig41):#8-24
    x = (I)*DTH
    theta4[I] = -att + x
    hpot1[I] = 1+(psizrac-1)+eps4*np.cos(theta4[I]+ att-fi4)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps4*np.cos(theta4[I]+att-fi4)

"""segment broj 1"""

```



```

for I in range(ig51,ig61):#24-39
    x = (I)*DTH
    theta1[I] = -att + x
    """hpot1[I] = 1+psizrac-1+eps1*np.cos(theta1[I]) """
    hpot1[I] = 1+(psizrac-1)+eps1*np.cos(theta1[I]+ att-fi1)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps1*np.cos(theta1[I]+att-fi1)

    """segment broj 2"""
for I in range(ig71,ig81):#39-54
    x = (I)*DTH
    theta2[I] = -att + x
    """hpot2[I] =I"""
    hpot1[I] = 1+(psizrac-1)+eps2*np.cos(theta2[I]+ att-fi2)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps2*np.cos(theta2[I] + att - fi2)

    """segment broj 3 - drugi dio"""
for I in range(ig81,ig91):#54-62
    x = (I)*DTH
    theta3[I] = - att + x
    """hpot2[I] =I"""
    hpot1[I] = 1+(psizrac-1)+eps3*np.cos(theta3[I] + att - fi3)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps3*np.cos(theta3[I] + att - fi3)

```

Iz aproksimirane Reynoldsove jednadžbe dobivene metodom konačnih razlika računaju se koeficijenti koji su opisani jednadžbom (3.1). Također je potrebno definirati matrice koje će sadržavati vrijednosti spomenutih koeficijenata.

```

A1 = np.zeros([m,n])
A2 = np.zeros([m,n])
A3 = np.zeros([m,n])
A4 = np.zeros([m,n])
A5 = np.zeros([m,n])
A6 = np.zeros([m,n])

```

Unutar prve petlje se metodom konačnih razlika, nakon definiranja matrica koeficijenata, popunjavaju matrice koeficijenata. Vrijednosti se računaju izrazima od (3.5) do (3.10).

```

"""FDM za stacionarno stanje"""
for J in range(1,n):
    for I in range(1,m):
        cc1 = h[I+1,J]-h[I-1,J]
        A1[I,J] = h[I,J]**(3) + 0.75*cc1*h[I,J]**(2)
        A2[I,J] = h[I,J]**(3) - 0.75*cc1*h[I,J]**(2)
        A3[I,J] = alfa*h[I,J]**(3)
        A4[I,J] = alfa*h[I,J]**(3)
        A5[I,J] = -1.5*DTH*cc1
        A6[I,J] = 2*(1+alfa)*h[I,J]**(3)

```

Unutar prve `while` petlje mora se nalaziti petlja za računanje tlakja. Potrebno je također zadovoljiti uvjete nakon kojih se iterativni postupak zaustavlja. Parametar `er1` mora biti manji od najmanje preciznosti i broj iteracija mora biti manji od *1000*. Zatim se korača po domeni po svakoj točki, te se svakoj koordinati koja u matrici `switch1` nije jedan dodjeljuje vrijednost tlaka izračunata pomoću izraza (3.12). Svaka vrijednost statičkog tlaka u matrici tlakova `p` koja je manja od nule zamjenjuje se nulom. Drugim rječima pomoću Reynoldsove jednadžbe i definiranih koeficijenata se računaju vrijednosti tlaka za čvorove koji se ne nalaze na žljebovima.

```
n1 = 0
sum2 = 0
er1 = 1
while (er1>Accuracy) and (n1<10**3):
    for J in range(1,n):
        for I in range(1,m):
            if (switch1[I,J] != 1):
                term = (A1[I,J]*p[I+1,J]+A2[I,J]*p[I-1,J]+A3[I,J]*p[I,J+1]+A4[I,J]*p[I,J-1]+A5[I,J])/A6[I,J];
                """Gauss-Seidel over-relaxation"""
                p[I,J] = p[I,J]+ORF*(term-p[I,J])
                """first B.C's (no cavitation)"""
                if (p[I,J] < 0):
                    p[I,J] = 0
```

Zbog moguće divergencije potrebno je koristiti prethodno definirani nadrelaksacijski faktor.

```
"""Gauss-Seidel over-relaxation"""
p[I,J] = p[I,J]+ORF*(term-p[I,J])
```

Prije završetka svake iteracije dobiva se nova razlika rješenja te se `n1` povećava za jedan

```
"""estimate error"""
er1 = abs(sum1-sum2)/sum1
sum2 = sum1
n1 = n1+1
```

Na kraju, računaju se tangencijalna i aksijalna komponenta sile. Proračun završava integracijom tlakova po malim površinama na provrtu ležaja, nakon zadovoljavanja zakona statike.

```
"""calculating load carrying capacity (Trapezoidal Integration)
integration in axial direction"""
inty = np.zeros(m) #Zamjena?
for I in range(m):
    for J in range(1,n):
        inty[I] = inty[I]+p[I,J]+p[I,J-1]
    inty[I] = inty[I]*0.5*DZ

"""axialw, tangew: axial and tangential component of load"""
axialw = 0
tangew = 0
```

```

"""integration in circumferential direction"""
for I in range(1,m):
    x = (I)*DTH;
    x2 = (I-1)*DTH;
    axialw = axialw-np.cos(x)*inty[I]-np.cos(x2)*inty[I-1]
    tangew = tangew+np.sin(x)*inty[I]+np.sin(x2)*inty[I-1]
axialw = axialw*0.5*DTH
tangew = tangew*0.5*DTH
loadw = np.sqrt(axialw**2+tangew**2)
attang = np.arctan(tangew/axialw)
if (axialw > 0):
    attang1 = attang
elif (axialw < 0):
    attang1 = -attang

```

Prilikom iteracijskog postupka, kao što je već navedeno, moguća je divergencija. Iz tog razloga se treba koristiti podrelaksacijski faktor.

```

"""Gauss-Seidel under-relaxation"""
att = att + URF*attang1
er2 = abs(att-att1)/att
att1 = att

```

Na kraju funkcija `funcrion_steady6` vraća dobivene rezultate kako bi se mogli koristiti u datoteci *Main*.

```

return[kut,hpot1,theta2,theta1,theta4,theta3,sirin,hpot,theta,p,axialw,tangew,
loadw,att]

```

4.3. Dinamičke funkcije (Dynamic_1 i Dynamic_2)

Dinamički oblici Reynoldsove jednačbe opisani su datotekama *Dynamic_1* i *Dynamic_2*. U stvarnosti se osim statičkog opterećenja javlja dinamičko opterećenje gibanjem osovine u ležaju. U datotekama *Dynamic_1* i *Dynamic_2* nalaze se funkcije `function_d14segment` i `function_d24segment`, te se obje pozivaju u datoteci *Main*.

Parametri koje obe funkcije traže su sljedeći:

`m`, `n` – Broj čvorova na kružnoj i uzdužnoj osi mreže, m, n

`epsi` – Promatrani relativni ekscentritet, ε

`lod` – Omjer promjera osovine i duljine ležaja, D/L

`groovx` – Polovica pripadnog kuta žlijeba, β

`groovz` – Polovica pripadne duljine žlijeba, l

`ps` – Dobavni tlak

`omega` – Omjer kutne brzine i frekvencije uznemiravanja

`phi` – Kut pomaka osovine

`er` – Razlika tlakova između dvije iteracije

`n1` – Brojač iteracija

`ORF`, `URF` – Nadrelaksacijski i podrelaksacijski faktori

`DTH`, `DZ` – Udaljenost između dvaju čvora u radijalnom i uzdužnom smjeru

`Switch1` – Matrica položaja žljebova

`p`, `h` – Statički tlak i debljina uljnog filma, p, h

`ig1`, `ig2` – Početna i krajnja pozicija prvog žlijeba na radijalnoj osi

`ig3`, `ig4` – Početna i krajnja pozicija drugog žlijeba na radijalnoj osi

`ig3`, `ig4` – Početna i krajnja pozicija trećeg žlijeba na radijalnoj osi

ig3, ig4 – Početna i krajnja pozicija četvrtog žlijeba na radijalnoj osi

jg1, jg2 – Početna i krajnja pozicija žljebova na uzdužnoj osi

ig11, ig21 – Početna i krajnja pozicija prvog dijela trećeg segmenta na radijalnoj osi

ig31, ig41 – Početna i krajnja pozicija četvrtog segmenta na radijalnoj osi

ig51, ig61 – Početna i krajnja pozicija prvog segmenta na radijalnoj osi

ig71, ig81 – Početna i krajnja pozicija drugog segmenta na radijalnoj osi

ig81, ig91 – Početna i krajnja pozicija drugog dijela trećeg segmenta na radijalnoj osi

Slično kao i u proračunu statičke funkcije, potrebno je pozvati paket *Numpy* kako bi se sa lakoćom koristile osnovne matematičke funkcije.

```
import numpy as np
```

Funkcije `function_d14segment` i `function_d24segment` zahtijevaju unos pojedinih parametara.

```
def function_d14segment(m, n, epsi, lod, groovx, groovz, ps, omega, p, phi):
```

```
def function_d24segment(m, n, epsi, lod, groovx, groovz, ps, omega, p, phi):
```

Traženi parametri su :

m, n – Broj čvorova na kružnoj i uzdužnoj osi mreže

epsi – Promatrani relativni ekscentritet

lod – Omjer promjera osovine i duljine ležaja

groovx – Polovica pripadnog kuta žlijeba

groovz – Polovica pripadne duljine žlijeba

ps – Dobavni tlak, p_s

omega – Omjer kutne brzine i frekvencije uznemiravanja, Ω

p – Statički tlak, p

ϕ – Kut pomaka osovine, α

U nastavku su definirane vrijednosti varijabla i definirane dimenzije nizova i matrica koje se koriste u kodu.

```
ORF = 1.8
DTH = 2.0*np.pi/(m-1)
DZ = 1.0/(n-1)
switch1 = np.zeros([m+1,n+1])
p1 = np.zeros([m+1,n+1],dtype=complex)
h = np.zeros([m+1,n+1])
hpot = np.zeros([m+1])
hpot1 = np.zeros([m+1])
theta = np.zeros([m+1])
theta1 = np.zeros([m+1])
theta2 = np.zeros([m+1])
theta3 = np.zeros([m+1])
theta4 = np.zeros([m+1])
kut = np.zeros([m+1])
sirin = np.zeros([n+1])
```

Definicije varijabla i dimenzije nizova u funkciji `function_d24segment` su slične kao i u funkciji `function_d14segment` osim definicije matrice tlakova koja se naziva `p2`.

```
DTH = 2.0*np.pi/(m-1)
DZ = 1.0/(n-1)
switch1 = np.zeros([m+1,n+1])
p2 = np.zeros([m+1,n+1],dtype=complex)
h = np.zeros([m+1,n+1])
hpot = np.zeros([m+1])
hpot1 = np.zeros([m+1])
theta = np.zeros([m+1])
theta1 = np.zeros([m+1])
theta2 = np.zeros([m+1])
theta3 = np.zeros([m+1])
theta4 = np.zeros([m+1])
kut = np.zeros([m+1])
sirin = np.zeros([n+1])
```

Potrebno je također kao i u datoteci *Steady* definirati položaje žlijebova pomoću koordinata.

Pažnju treba posvetiti činjenici da koordinate žlijebova moraju biti jednake za sve tri datoteke kako bi se mogla pokrenuti datoteka *Main*.

```
i1 = round(groovx/DTH)
j1 = round(groovz/DZ)
ig1 = int(((m-1)*0.67/4)-i1) #8
ig2 = int(((m-1)*0.67/4)+i1) #12
ig3 = int(((m-1)*1.67/4)-i1) #23
ig4 = int(((m-1)*1.67/4)+i1) #27
ig5 = int(((m-1)*2.67/4)-i1) #38
ig6 = int(((m-1)*2.67/4)+i1) #42
```

```

ig7 = int(((m-1)*3.67/4)-i1) #53
ig8 = int(((m-1)*3.67/4)+i1) #57
jg1 = int(((n-1)/2)-j1)
jg2 = int(((n-1)/2)+j1)

```

Na tim pozicijama treba dodati vrijednost dobavnog tlaka u matricu komponenta dinamičkog tlaka. Također treba spomenuti da svaka funkcija (`function_d14segment` i `function_d24segment`) sadrži vlastitu komponentu dinamičkog tlaka.

```

"""first groove"""
for J in range(jg1,jg2+1):
    for I in range(ig1,ig2+1):
        p1[I,J] = ps
        switch1[I,J] = 1

"""second groove"""
for J in range(jg1,jg2+1):
    for I in range(ig3,ig4+1):
        p1[I,J] = ps
        switch1[I,J] = 1

"""third groove"""
for J in range(jg1,jg2+1):
    for I in range(ig5,ig6+1):
        p1[I,J] = ps
        switch1[I,J] = 1

"""fourth groove"""
for J in range(jg1,jg2+1):
    for I in range(ig7,ig8+1):
        p1[I,J] = ps
        switch1[I,J] = 1

```

Sljedeći korak je izračun debljine uljnog filma. Formule za izračun kutova i debljine uljnog filma su jednake kao i u datoteci *Steady*.

```

eps1=np.sqrt(eps1**2+deltazrac**2+2*eps1*deltazrac*np.cos(phi))
arg1=eps1*np.sin(phi)/(deltazrac+eps1*np.cos(phi))
fi1=np.arctan(arg1)

eps2=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(np.pi/2+phi))
arg2=(deltazrac+eps1*np.sin(phi))/(eps1*np.sin(phi))
fi2=np.arctan(arg2)

eps3=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(phi))
arg3=eps1*np.sin(phi)/(-deltazrac+eps1*np.cos(phi))
fi3=np.pi+np.arctan(arg3)

eps4=np.sqrt(eps1**2+deltazrac**2-2*eps1*deltazrac*np.cos(np.pi/2-phi))
arg4=(-deltazrac+eps1*np.sin(phi))/(eps1*np.sin(phi))
fi4=np.arctan(arg4)

```

Segmenti su također definirani pomoću niza koordinata.

```

ig11 = 1 #0
ig21 = 8 #2

```

```

ig31 = 8 #18
ig41 = 24 #22

ig51 = 24 #38
ig61 = 39 #42

ig71 = 39 #59
ig81 = 54 #59
ig91 = 62 #61

```

Na jednak način, kao i u datoteci *Steady*, popunjava se matrica h .

```

"""segment broj 3 - prvi dio"""
for I in range(ig11,ig21):
    x = (I)*DTH
    theta3[I] = -phi + x
    hpot1[I] = 1+(psizrac-1)+eps3*np.cos(theta3[I]+ phi-fi3)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps3*np.cos(theta3[I]+phi-fi3)

"""segment broj 4"""
for I in range(ig31,ig41):
    x = (I)*DTH
    theta4[I] = -phi + x
    hpot1[I] = 1+(psizrac-1)+eps4*np.cos(theta4[I]+ phi-fi4)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps4*np.cos(theta4[I]+phi-fi4)

"""segment broj 1"""
for I in range(ig51,ig61):
    x = (I)*DTH
    theta1[I] = -phi + x
    hpot1[I] = 1+(psizrac-1)+eps1*np.cos(theta1[I]+ phi-fi1)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps1*np.cos(theta1[I]+phi-fi1)

"""segment broj 2"""
for I in range(ig71,ig81):
    x = (I)*DTH
    theta2[I] = -phi + x
    hpot1[I] = 1+(psizrac-1)+eps2*np.cos(theta2[I]+ phi-fi2)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps2*np.cos(theta2[I] + phi - fi2)

"""segment broj 3 - drugi dio"""
for I in range(ig81,ig91):
    x = (I)*DTH
    theta3[I] = - phi+ x
    """hpot2[I] =I"""
    hpot1[I] = 1+(psizrac-1)+eps3*np.cos(theta3[I] + phi - fi3)

    for J in range(1,n+1):
        h[I,J] = 1+(psizrac-1)+eps3*np.cos(theta3[I] + phi - fi3)

```


Prije nego što se prijeđe na računanje koeficijenata dinamičkih oblika Reynoldsove jednačbe, potrebno je definirati nizove u kojima će njihove vrijednosti biti spremljene. Treba naglasiti da su koeficijenti kod dinamičkih oblika Reynoldsove jednačbe kompleksne prirode.

```
A1 = np.zeros([m,n], dtype=complex)
A2 = np.zeros([m,n], dtype=complex)
A3 = np.zeros([m,n], dtype=complex)
A4 = np.zeros([m,n], dtype=complex)
A5 = np.zeros([m,n], dtype=complex)
A6 = np.zeros([m,n], dtype=complex)
A7 = np.zeros([m,n], dtype=complex)
A8 = np.zeros([m,n], dtype=complex)
A9 = np.zeros([m,n], dtype=complex)
A10 = np.zeros([m,n], dtype=complex)
A11 = np.zeros([m,n], dtype=complex)
```

Nakon definiranja započinjemo sa rješavanjem koeficijenata dinamičkih oblika Reynoldsovih jednačbi metodom konačnih razlika.

```
for J in range(1,n):
    for I in range(1,m):
        seta = I*DTH
        cc1 = h[I+1,J]-h[I-1,J]
        A1[I,J] = h[I,J]**(3) + 0.75*cc1*h[I,J]**2
        A2[I,J] = h[I,J]**(3) - 0.75*cc1*h[I,J]**2
        A3[I,J] = alfa*h[I,J]**(3)
        A4[I,J] = alfa*h[I,J]**(3)
        A5[I,J] = -6*np.sin(seta)*(1+alfa)*h[I,J]**(2)
        A6[I,J] = 3*np.sin(seta)*h[I,J]**(2) +
1.5*np.sin(seta)*h[I,J]*cc1 + 1.5*np.cos(seta)*DTH*h[I,J]**(2)
        A7[I,J] = 3*np.sin(seta)*h[I,J]**(2) -
1.5*np.sin(seta)*h[I,J]*cc1 - 1.5*np.cos(seta)*DTH*h[I,J]**(2)
        A8[I,J] = 3*alfa*np.sin(seta)*h[I,J]**(2)
        A9[I,J] = 3*alfa*np.sin(seta)*h[I,J]**(2)
        A10[I,J] = -(DTH**2)*(3*np.cos(seta)+1j*6*omega*np.sin(seta))
        A11[I,J] = 2*(h[I,J]**(3))*(1+alfa)
```

Izračunati koeficijenti se koriste u petlji za računanje dinamičkih tlakova, gdje se rješavaju jednačbe dobivene metodom konačnih razlika. Kada se ispune uvjeti u petlji ona se zaustavlja.

```
"""main loop for calculating pressure"""
while (er > Accuracy) and (n1 < 10**3):
    for J in range(1,n):
        for I in range(1,m):
            if (switch1[I,J] != 1):
                po = p2[I,J]
                p2[I,J] = (A1[I,J]*p2[I+1,J] + A2[I,J]*p2[I-1,J] +
A3[I,J]*p2[I,J+1] + \
                        A4[I,J]*p2[I,J-1] + A5[I,J]*p[I,J] + A6[I,J]*p[I+1,J]
+ A7[I,J]*p[I-1,J] + \
                        A8[I,J]*p[I,J+1] + A9[I,J]*p[I,J-1] +
A10[I,J])/A11[I,J]
```

Divergencija rješenja se izbjegava nadrelaksacijom, pritom se koristi nadrelaksacijski faktor koji je prethodno definiran.

```
"""Gauss-Seidel over-relaxation"""  
p2[I,J] = po+ORF*(p2[I,J]-po)
```

Rubni uvjeti su slični kao i u statičkoj funkciji. Negativne komponente dinamičkih tlakova poprimaju vrijednosti nula.

```
"""first B.C's (no cavitation)"""  
if p[I,J] < 0:  
    p2[I,J] = 0.0
```

U rubne uvjete spada i gradijent promjene tlaka u radijalnom smjeru koji je također jednak nuli.

```
sum1 = sum(sum(p2))  
"""symmetry B.C's (dp/dz = 0)"""  
p2[:,0] = p2[:,2]
```

Na kraju petlje potrebno je računati razliku rješenja dviju iteracija, te povećati brojač za jedan.

```
"""estimate error"""  
er = abs(1-sum2/sum1)  
sum2 = sum1  
n1 = n1+1
```

Posljednji korak je računanje sila prilikom efekta perturbacije. Sile koje dobijemo su aksijalna i tangencijalna komponenta resultantne sile koja opisuje nosivost ležaja.

```
"""calculating dynamic load carrying capacity (Trapezoidal Integration)  
integration in axial direction"""  
inty = np.zeros(m,dtype=complex) #Zamjena?  
for I in range(m):  
    inty[I]=0  
    for J in range(1,n):  
        inty[I] = inty[I]+p2[I,J]+p2[I,J-1]  
    inty[I] = inty[I]*0.5*DZ  
"""axialw, tangew: axial and tangential component of load"""  
axialw = 0  
tangew = 0  
"""integration in circumferential direction"""  
for I in range(1,m):  
    x = (I)*DTH  
    x2 = (I-1)*DTH  
    axialw = axialw+np.cos(x)*inty[I]+np.cos(x2)*inty[I-1]  
    tangew = tangew+np.sin(x)*inty[I]+np.sin(x2)*inty[I-1]
```

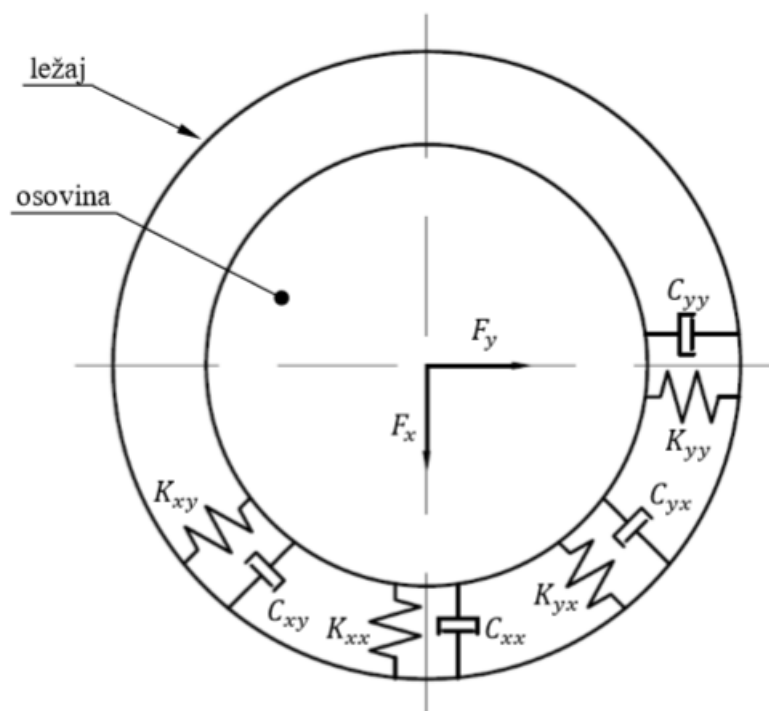
Komponente nosivosti ležaja prenose se funkcijama `function_d14segment` i `function_d24segment` u datoteku *Main*.

```
WXD1 = axialw*0.5*DTH  
WZD1 = tangew*0.5*DTH  
  
return[WXD1,WZD1]
```

```
WXD2 = axialw*0.5*DTH  
WZD2 = tangew*0.5*DTH  
  
return[WXD2,WZD2]
```

5. REZULTATI

Prilikom rotacije osovine dobiva se hidrodinamički tlak, koji je ključni faktor u određivanju nosivosti filma ulja. U prethodnim poglavljima detaljno je objašnjen postupak dobivanja nosivih sila, te se zaključuje da one ovise o relativnom ekscentritetu odnosno o položaju osovine u odnosu na ležaj. Za sustav vrijedi da je poznata i viskoznost lubrikanta kod određene brzine rotacije. Zbog poznavanja spomenutih svojstava i koeficijenata krutosti i prigušenja moguće je zamijeniti djelovanje fluida oprugom. Oni definiraju kakva opruga i prigušivač djeluje jednako kao ležaj u određenim osima.



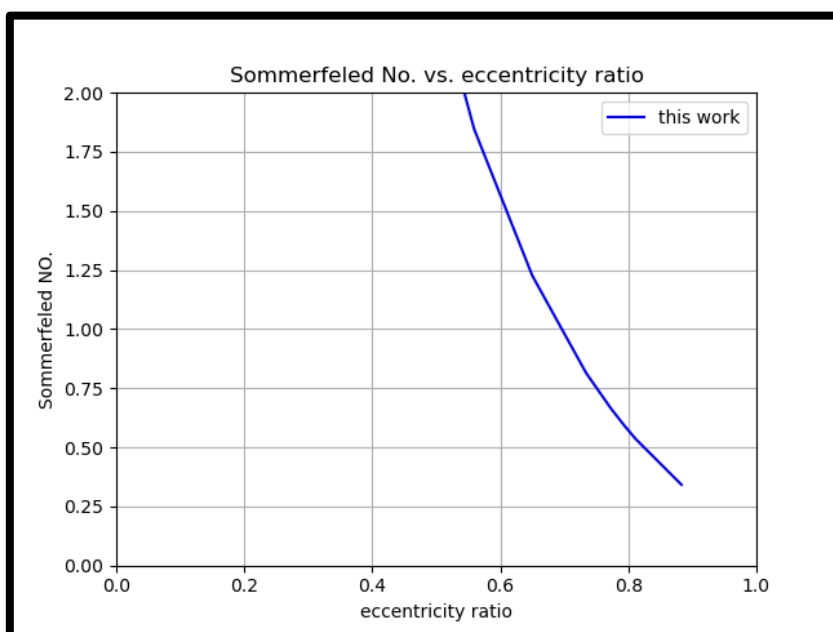
Slika 5.1. Vibrirajući sustav u ležaju

Datoteka Main ispisuje rezultate za svaku vrijednost relativnog ekscentriteta sadržane u nizu. Računa za svaku vrijednost kut pomaka osovine, Sommerfeldov broj, nosive sile i koeficijente krutosti i prigušenja.

hpot1 = 0.0000 hpot = 0.0000 theta = -1.2590 att = 1.2590 eccentricity ratio = 0.103 attitude angle = 72.1354 sommerfeled NO. = 19.0511 Fx = 0.00874841554774799 Fy = 1.746918543998812e-10 K_xx = 6.3136 ; K_xy = 30.7709 K_yx = -37.1043 ; K_yy = 6.3179 C_xx = 62.9340 ; C_xy = 0.3391 C_yx = 0.1644 ; C_yy = 76.4422 *****	hpot1 = 1.1011 hpot = 0.0000 theta = -0.9713 att = 1.0760 eccentricity ratio = 0.150 attitude angle = 61.6509 sommerfeled NO. = 12.4403 Fx = 0.013397307710150132 Fy = 2.2433443093230738e-10 K_xx = 4.4105 ; K_xy = 20.7305 K_yx = -24.7520 ; K_yy = 4.3299 C_xx = 42.6116 ; C_xy = 0.4268 C_yx = 0.3307 ; C_yy = 50.9412 *****
hpot1 = 1.2183 hpot = 0.0000 theta = -0.8203 att = 1.0298 eccentricity ratio = 0.224 attitude angle = 59.0011 sommerfeled NO. = 8.1318 Fx = 0.02049572486517028 Fy = 3.462183282938212e-10 K_xx = 3.2500 ; K_xy = 14.2370 K_yx = -17.0440 ; K_yy = 3.1437 C_xx = 29.4848 ; C_xy = 0.5828 C_yx = 0.5327 ; C_yy = 35.0226 *****	hpot1 = 1.4090 hpot = 0.0000 theta = -0.7306 att = 1.0447 eccentricity ratio = 0.352 attitude angle = 59.8585 sommerfeled NO. = 4.6341 Fx = 0.03596498092627808 Fy = 6.113110804519705e-10 K_xx = 2.5057 ; K_xy = 9.0052 K_yx = -11.2525 ; K_yy = 2.2530 C_xx = 18.8538 ; C_xy = 0.8573 C_yx = 0.8358 ; C_yy = 23.1093 *****
hpot1 = 1.6690 hpot = 0.0000 theta = -0.4722 att = 0.8911 eccentricity ratio = 0.460 attitude angle = 51.0566 sommerfeled NO. = 2.8390 Fx = 0.058705104749189614 Fy = 7.519662322579966e-10 K_xx = 2.4744 ; K_xy = 6.8433 K_yx = -7.8537 ; K_yy = 1.8825 C_xx = 14.6885 ; C_xy = 1.1310 C_yx = 1.1198 ; C_yy = 15.9712 *****	hpot1 = 1.9382 hpot = 0.0000 theta = -0.2861 att = 0.8097 eccentricity ratio = 0.559 attitude angle = 46.3949 sommerfeled NO. = 1.8469 Fx = 0.09024223501043968 Fy = 1.0383777956056408e-09 K_xx = 2.6623 ; K_xy = 5.6759 K_yx = -5.9237 ; K_yy = 1.6919 C_xx = 12.4997 ; C_xy = 1.3048 C_yx = 1.2958 ; C_yy = 11.8930 *****
hpot1 = 2.2175 hpot = 0.0000 theta = -0.1291 att = 0.7574 eccentricity ratio = 0.650 attitude angle = 43.3977 sommerfeled NO. = 1.2256 Fx = 0.13598696075039338 Fy = 1.4217931385214604e-09 K_xx = 2.9882 ; K_xy = 4.9388 K_yx = -4.6351 ; K_yy = 1.5530 C_xx = 11.1624 ; C_xy = 1.4167 C_yx = 1.4073 ; C_yy = 9.1543 *****	hpot1 = 2.5045 hpot = 0.0000 theta = 0.0130 att = 0.7200 eccentricity ratio = 0.734 attitude angle = 41.2542 sommerfeled NO. = 0.8143 Fx = 0.20468112352523435 Fy = 2.096344550168212e-09 K_xx = 3.4461 ; K_xy = 4.4404 K_yx = -3.6931 ; K_yy = 1.4363 C_xx = 10.2944 ; C_xy = 1.4868 C_yx = 1.4760 ; C_yy = 7.1460 *****

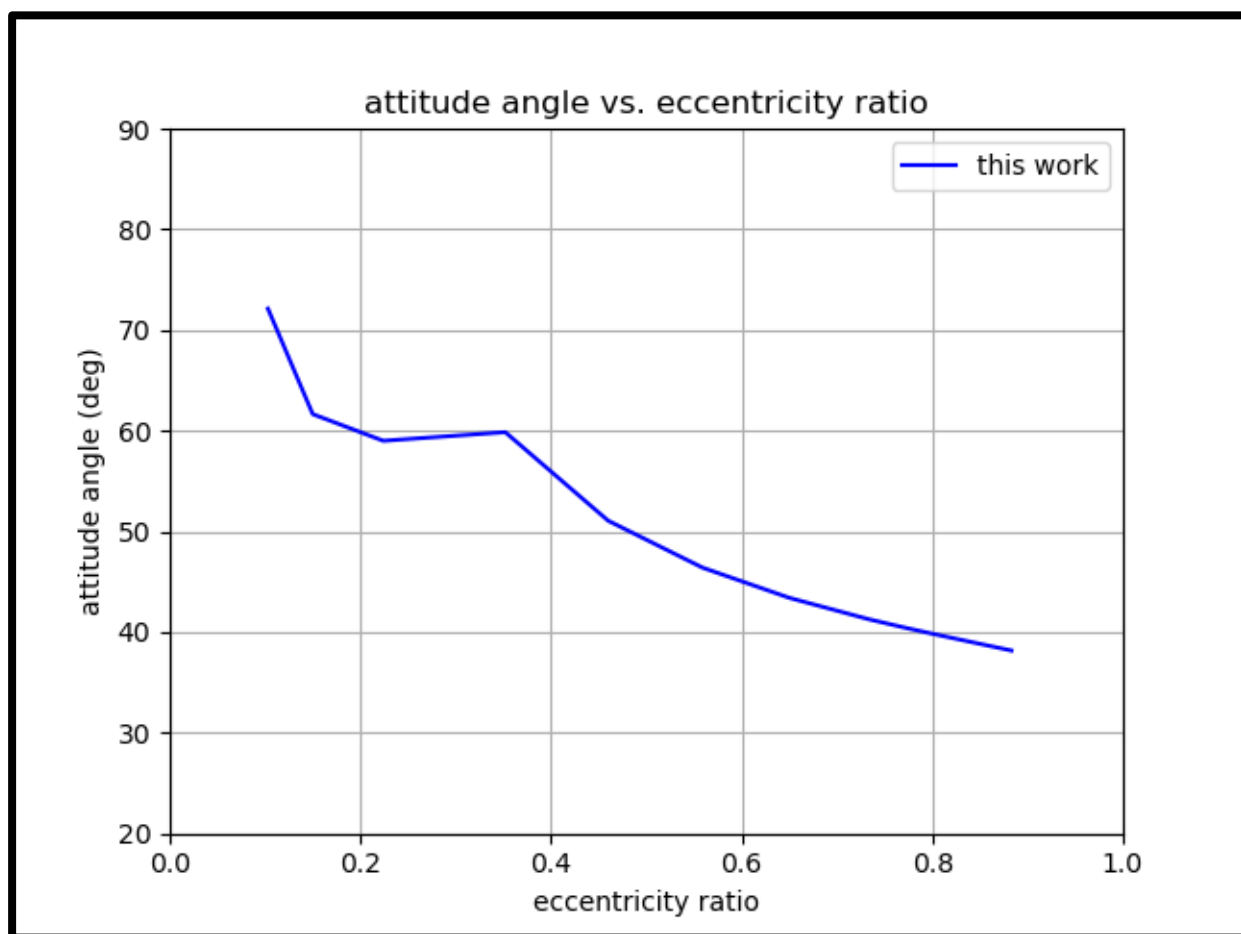
<pre> hpot1 = 2.4669 hpot = 0.0000 theta = 0.1327 att = 0.7051 eccentricity ratio = 0.773 attitude angle = 40.3970 sommerfeled NO. = 0.6633 Fx = 0.2512621692397719 Fy = 2.7683039666194143e-09 K_xx = 3.7265 ; K_xy = 4.2565 K_yx = -3.3110 ; K_yy = 1.3831 C_xx = 9.9861 ; C_xy = 1.5102 C_yx = 1.4986 ; C_yy = 6.3326 ***** </pre>	<pre> hpot1 = 2.2701 hpot = 0.0000 theta = 0.2447 att = 0.6978 eccentricity ratio = 0.793 attitude angle = 39.9785 sommerfeled NO. = 0.5942 Fx = 0.28051187632849545 Fy = 3.0610123041272895e-09 K_xx = 3.8924 ; K_xy = 4.1734 K_yx = -3.1249 ; K_yy = 1.3555 C_xx = 9.8505 ; C_xy = 1.5203 C_yx = 1.5083 ; C_yy = 5.9373 ***** </pre>
<pre> hpot1 = 2.0902 hpot = 0.0000 theta = 0.3559 att = 0.6913 eccentricity ratio = 0.811 attitude angle = 39.6091 sommerfeled NO. = 0.5363 Fx = 0.3107930619686571 Fy = 2.8873545130389846e-09 K_xx = 4.0574 ; K_xy = 4.1052 K_yx = -2.9619 ; K_yy = 1.3304 C_xx = 9.7421 ; C_xy = 1.5283 C_yx = 1.5159 ; C_yy = 5.5918 ***** </pre>	<pre> hpot1 = 1.9505 hpot = 0.0000 theta = 0.4859 att = 0.6661 eccentricity ratio = 0.883 attitude angle = 38.1621 sommerfeled NO. = 0.3415 Fx = 0.4880027601956612 Fy = 5.214769973397876e-09 K_xx = 4.9214 ; K_xy = 3.9052 K_yx = -2.3410 ; K_yy = 1.2274 C_xx = 9.4567 ; C_xy = 1.5538 C_yx = 1.5396 ; C_yy = 4.2876 ***** </pre>

Sljedećim dijagramom je prikazana ovisnost Sommerfeldovog broja o relativnom ekscentritetu, 5.1. Vidimo da se povećanjem relativnog ekscentriteta Sommerfeldov broj eksponencijalno smanjuje.



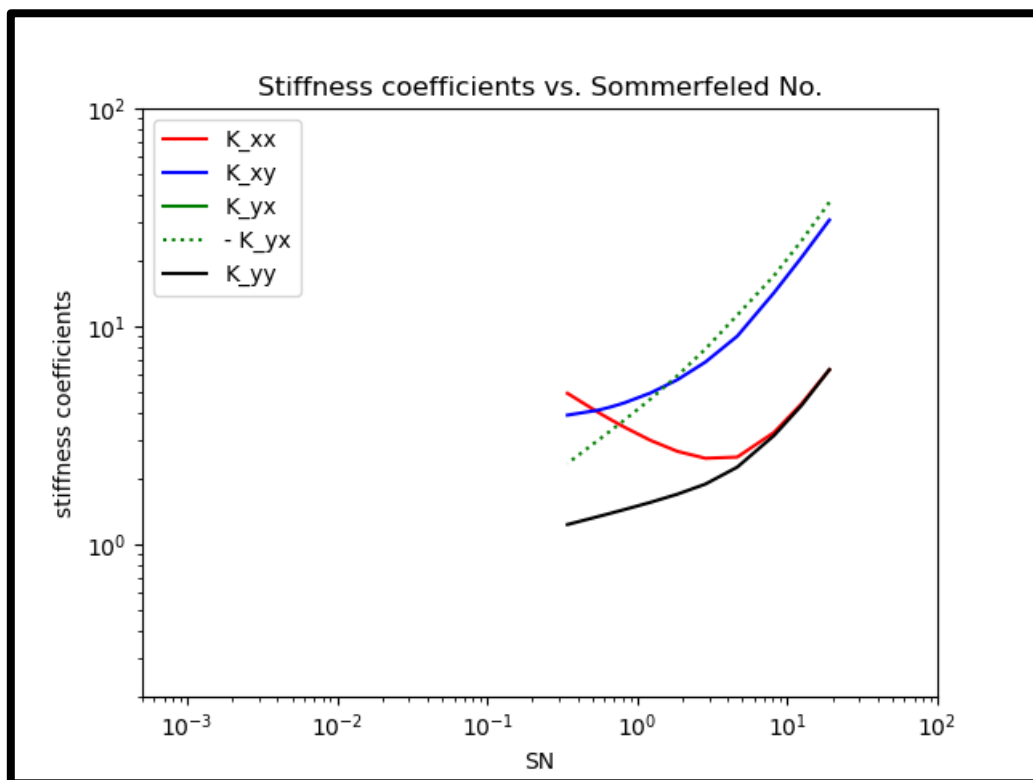
Slika 5.2. Ovisnost Sommerfeldovog broja o relativnom ekscentritetu

Sljedeći prikazani dijagram opisuje ovisnost kuta pomaka i relativnog ekscentriteta, 5.2. Moguće je zaključiti da kut pomaka osovine pada povećanjem ekscentriteta.



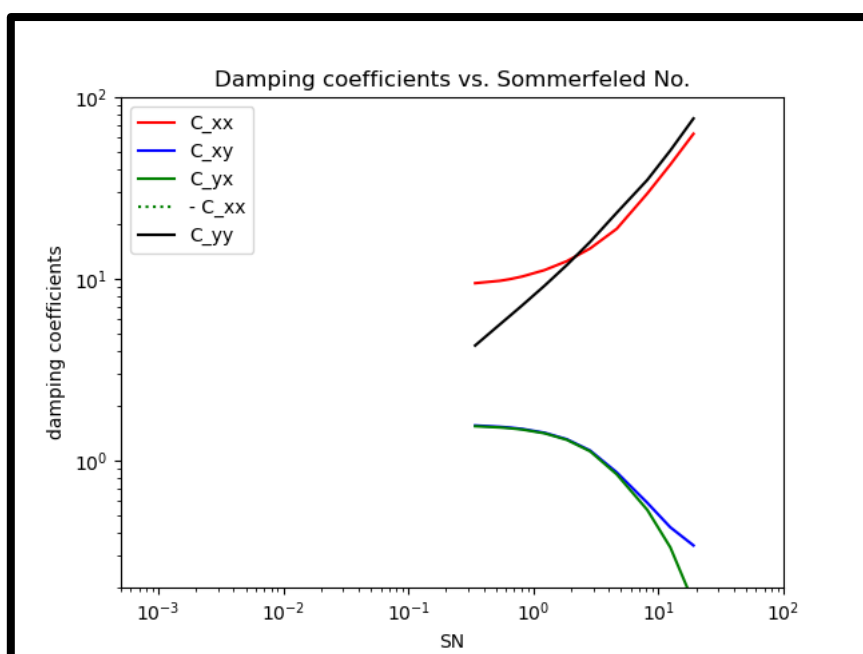
Slika 5.3. Ovisnost kuta pomaka osovine o ekscentritetu

Slika 5.3. prikazuje dijagram ovisnosti Sommerfeldovog broja i koeficijenta krutosti.



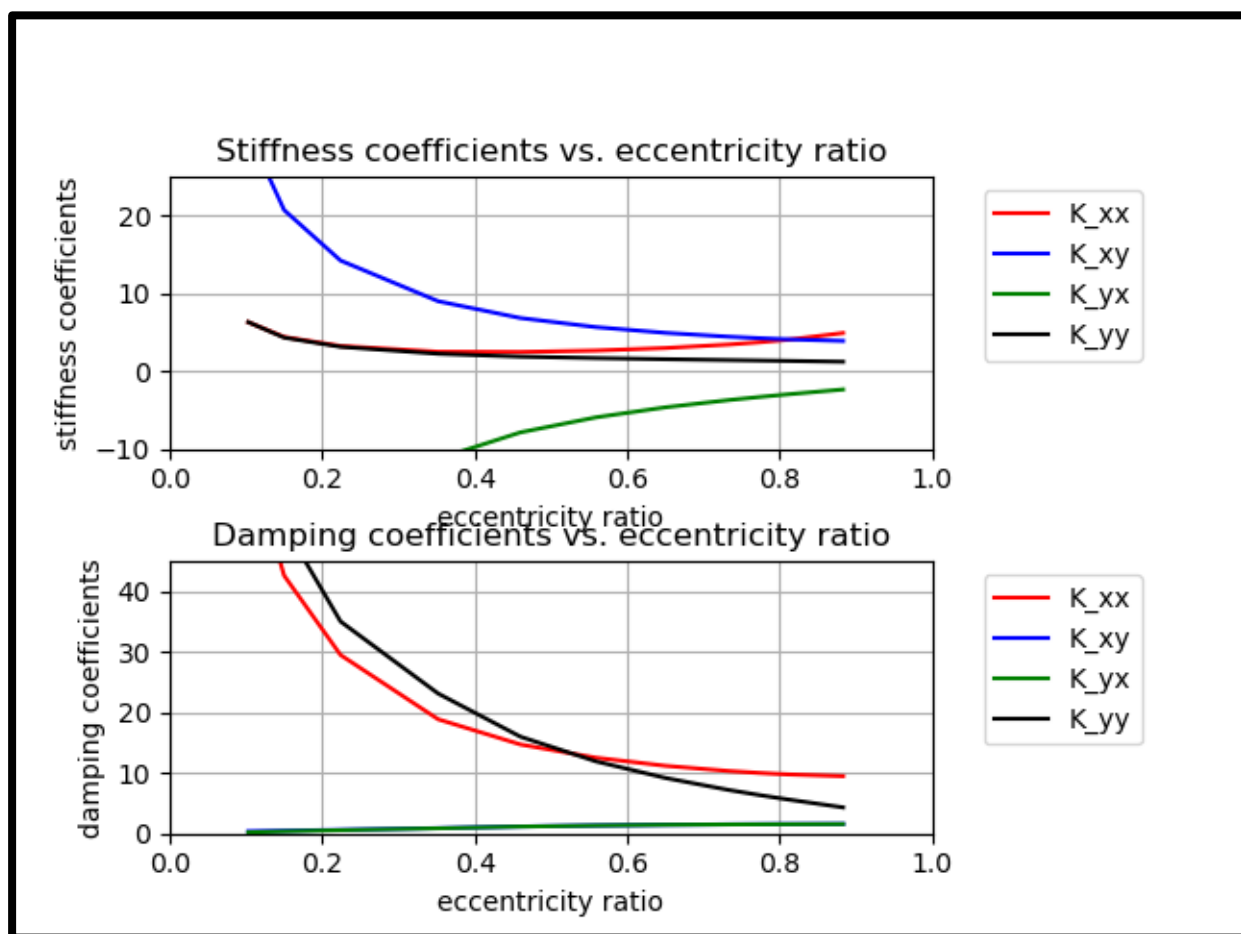
Slika5.4. Ovisnost koeficijenata krutosti o Sommerfeldovom broju

Isto se može prikazati za koeficijente krutosti,5.4.



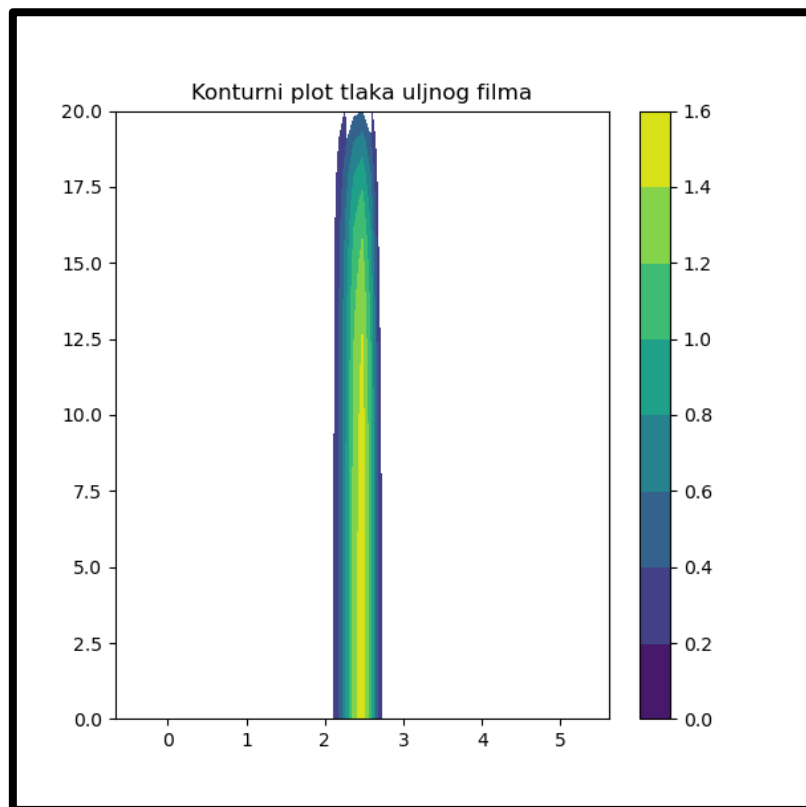
Slika 5.5. Ovisnost koeficijenata prigušenja o Sommerfeldovom broju

Na sljedećim dijagramima su prikazane ovisnosti koeficijenata krutosti i prigušenja u odnosu na relativni ekscentritet.

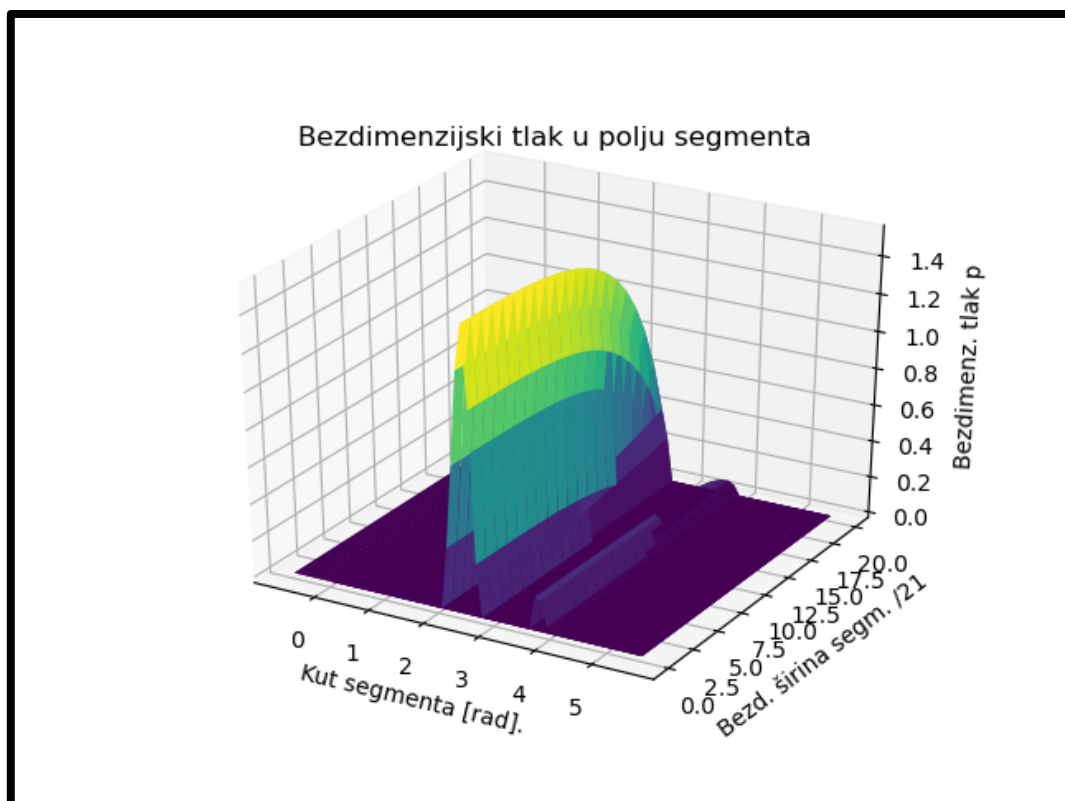


Slika 5.6. Ovisnost koeficijenata krutosti i prigušenja o ekscentritetu

Tlak raste sa porastom ekscentriteta. Tlak postaje veći na dijelu ležaja gdje opterećena osovina najviše pritišće ulje, te dostiže relativno visoke iznose u odnosu na ostatak plohe. Na sljedećim dijagramima moguće je vizualizirati intenzitet tlaka na različitim

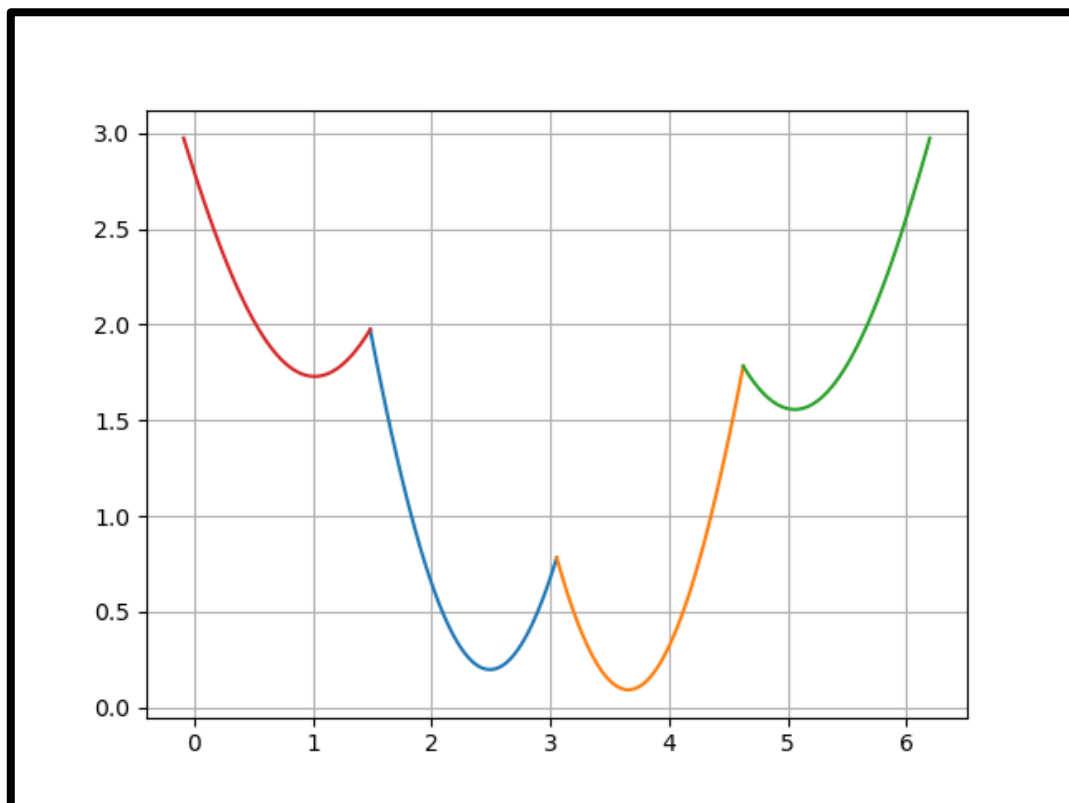


Slika 5.7. Raspodjela polja tlaka za zadnji ekscentritet

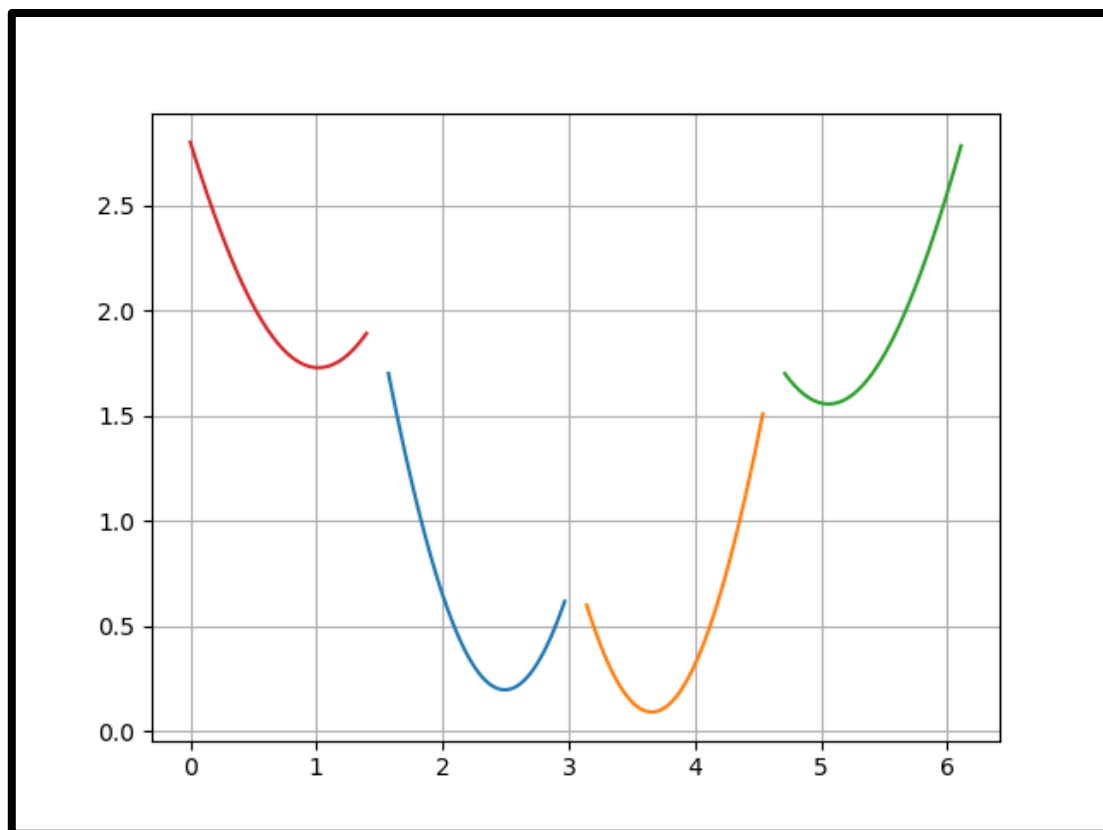


Slika 5.8. Raspodjela polja tlaka za zadnji ekscentritet

Također želimo prikazati debljine uljnog filma u svim segmentima (1, 2, 3 i 4).



Slika 5.9. Debljine uljnog filma po segmentima bez dovoda ulja



Slika 5.10. Debljine uljnog filma po segmentima sa dovodom ulja

6. ZAKLJUČAK

Ovaj rad opisuje postupak dobivanja koeficijenata krutosti i prigušenja četverosegmentnog uljnog ležaja sa četiri utora za dovod ulja. Osovina koja se nalazi u ležaju ima rotacijsko gibanje i vibrira, te je opterećena nekakvom silom. Zbog sličnosti koje sustav ležaja i osovine dijele sa vibracijskim sustavom može se opisati pomoću koeficijenata krutosti i prigušenja.

Proračunu je moguće pristupiti uz prethodno definirani matematički model, koji se dijeli na statički i dinamički dio Reynoldsove jednačbe. Jednom kada se odredi domena proračuna, postavljene diferencijalne jednačbe se sređuju metodom konačnih razlika, te se postavljaju potrebni rubni uvjeti. Iteracijskim postupkom se u programskom jeziku *Python* rješavaju sustavi algebarskih jednačbi, te se dobiju rezultati koji su prikazani dijagramima.