

GRADU AMAIERAKO LANA

Ardatzen neke analisia burutzen duen programa baten garapena

Iñigo Solaegi Barrio

Industria Teknologiaren Ingeniaritzako Gradua

Mikel Abasolo Bilbaok zuzendua

2017-2018 ikasturtea

AURKIBIDEA

AURKIBIDEA.....	2
1. LABURPENA.....	4
2. IRUDIAK, AKRONIMOAK ETA TAULAK	6
a. IRUDIAK	6
b. AKRONIMOAK.....	7
c. TAULAK.....	8
3. SARRERA.....	9
4. TESTUINGURUA.....	10
5. IRISMENA ETA HELBURUAK.....	13
6. PROIEKTUAREN ONURAK	14
7. ALTERNATIBEN ANALISIA	15
8. SOLUZIOAREN AZALPENA.....	17
8.1. OINARRI TEORIKOA:	17
8.1.1. S-N kurba:	17
8.1.2. Neke limitea:	17
8.1.3. Trantsizio tentsioa:.....	18
8.1.4. Diagrama logaritmikoak. Basquin-en ekuazioa:.....	18
8.1.5. Marin-en formula:	20
8.1.6. Bataz besteko tentsioen eragina:.....	21
8.1.7. Trakziozko bataz besteko tentsioentzako irizpideak:.....	23
8.1.8. Konpresioko bataz besteko tentsioaren eragina eta Haigh-en diagrama:.....	25
8.1.9. Tentsio kontzentrazioen efektua eta segurtasun koefizientea:.....	26
8.1.10. Tentsio multiaxiala:	27
8.2. PROGRAMAREN DESKIRBAPENA.....	32
8.3. PROGRAMA ERABILIZ PROBLEMA BATEN EBAZPENA	38
9. METODOLOGIA.....	44
10. AURREKONTUA.....	47
11. LANEN DESKIRBAPENA. GANTT.....	48
L1: Informazioa eskuratu.....	48

L2: Alternatiben analisia	48
L3: Programaren garapena.....	48
L5: Adibide ezberdinen eskuzko eta proiektu honetan sortutako programa bidezko kalkuluak egitea.....	49
L6: Emaitzak aztertzea.....	49
L7: Proiektua idaztea.....	49
L8: Aurkezpena prestatzea.....	49
12. ONDORIOAK	51
13. BIBLIOGRAFIA.....	52
14. ERANSKINAK.....	53
14.1. ERANSKINAK: KODEA.....	53
14.2. ERANSKINAK: INTERFAZE GRAFIKOA.....	77
14.3. ERANSKINAK: ESKUZKO ADIBIDEA.	81

1. LABURPENA

EUSKARAZ:

Proiektu honetan elementu finituetako postprozesatzaile batean ardatzen nekearen kalkulurako metodoen implementaziorako programa baten azalpena egingo da.

Nekearen analisisa sail zabala da eta etengabeko berrikuntza jasaten du eta horregatik, elementu finituen programa batzuk nekearen analisirako erraminta gutxi izaten dituzte, ANSYS programa esaterako.

Matlab software-a erabiliz eraikiko den programa honen bitartez, elementu finituen programen postprozesatzailean nekearen analisi sakonagoa egiteko aukera emango dio aztertzaileari, batez ere, neke multiaxialaren esparruan.

Programa hau beraz, elementu finituen programen osagarri moduan erabiltzeko proposa izango da, elementu erresistenteak (ardatzak) nekearen aurrean bizitza infinitua edukiko duen ala ez aurreikusteko, bizitza finitua izatekotan zenbat ziklo iraungo dituen kalkulatz baina elementuak denboran zehar jasango dituen tentsio aldakorren grafikaketa eta kalkulua eginez.

ESPAÑOL:

En el siguiente trabajo se expone un programa para la implementación de métodos de cálculo a fatiga para ejes en un postprocesador de elementos finitos.

El cálculo a fatiga es un campo amplio y en constante innovación, por lo cual algunos de los programas de elementos finitos carecen de herramientas para el análisis a fatiga, como por ejemplo el programa ANSYS.

Mediante el programa creado utilizando el software Matlab, el analista tendrá opción a realizar un cálculo más detallado a fatiga en un postprocesador de elementos finitos, sobre todo cuando se trata de un caso de fatiga multiaxial.

Por lo tanto, este programa es adecuado como complemento de otros programas de elementos finitos, con el cual se podrán calcular y graficar las tensiones variables a las que estarán sometidos los elementos resistentes analizados (ejes) y permitirá prever si dichos elementos tendrán o no vida infinita, en caso de no tenerla, se procedería al cálculo aproximado del número de ciclos hasta el fallo a fatiga.

ENGLISH:

The following project submits a program involving the establishment of methods for fatigue calculus for shafts in the postprocessor of a finite elements program.

Fatigue designing is a vast field which is in constant innovation, and that it is why some of finite elements programs lack of the necessary tools for proper and more complete fatigue analysis, such as ANSYS program.

The program created using Matlab's software in this project gives its user the chance to make a full fatigue analysis for shafts of different diameters and work conditions, especially for multiaxial fatigue.

That it is why this program can be useful as a complementary tool for a postprocessor of a finite elements program, which will calculate the variable strains the element (the shaft) be holding up and their graphic representation. It will also allow for foreseeing if the element would have infinite life or not, and if not, it will calculate approximately the number of cycles before failure.

2. IRUDIAK, AKRONIMOAK ETA TAULAK

a. IRUDIAK

IRUDIA 1: DENBORAN ALDAKORRAK DIREN TENTSIOEN ADIERAZPENA.....	10
IRUDIA 2: NEKEAGATIKO HUTSEGITEAREN FASEAK.....	10
IRUDIA 3: S-N KURBA.....	17
IRUDIA 4: ALTZAIRUEN S-N DIAGRAMA OROKORTUA.....	19
IRUDIA 5: PROBETA ETA PIEZAREN S-N KURBA OROKORTUAK.....	20
IRUDIA 6: TENTSIO EGOERA EZBERDINAK ERAKUSTEN DITUEN IRUDIA.....	22
IRUDIA 7: MATERIALEN HARIKOR(A) ETA HAUSKORRETAN(B) DIAGRAMA TRIDIMENSIONALAK.....	23
IRUDIA 8: SODERBERG(ΣP) ETA GOODMAN(ΣU)-EN IRIZPIDEAK MATERIAL HARIKORRETAN.....	23
IRUDIA 9: TRAKZIO ZONAKO IRIZPIDEEN ADIERAZPEN GRAFIKOA.....	24
IRUDIA 10: KONPRESIO ZONAKO TENTSIO EZBERDINETARAKO ZUZEN HURBILDUAK.EZKERRALDEAN, TRIDIMENSIONALAK ETA ESKUMALDEAN, BIDIMENSIONALAK.....	25
IRUDIA 11: HAIGH-EN DIAGRAMA HURBILDUA.....	25
IRUDIA 12: MATERIAL MOTAREN ARABERAKO 1.PROBLEMA MOTA, HUTSEGITEA EMANGO DENEKO N ZIKLO KOPURUAREN ADIERAZPEN GRAFIKOA. EZKERRALDEAN, MATERIAL HARIKORRENTZAKO ADIERAZPENA ETA ESKUMALDEAN, MATERIAL HAUSKORRENA.....	26
IRUDIA 13: MATERIAL MOTAREN ARABERAKO 2.PROBLEMA MOTA, HUTSEGITEA EMANGO DENEKO TENTSIOAREN ADIERAZPEN GRAFIKOA. EZKERRALDEAN,MATERIAL HARIKORRENTZAKO ADIERAZPENA ETA ESKUMALDEAN, MATERIAL HAUSKORRENA.....	26
IRUDIA 14: SEGURTASUN KOEFIZIENTEAK KURBETAN DUEN ERAGINAREN ADIERAZPEN GRAFIKOA. EZKERRALDEAN, MATERIAL HARIKORRENTZAT ETA ESKUMALDEAN, MATERIAL HAUSKORRENTZAT.	27
IRUDIA 15: TENTSIO KONTZENTRAZIO KOEFIZIENTEAK KURBETAN DUEN ERAGINAREN ADIERAZPEN GRAFIKOA. EZKERRALDEAN, MATERIAL HARIKORRENTZAT ETA ESKUMALDEAN, MATERIAL HAUSKORRENTZAT.	27
IRUDIA 16: SM METODOAREN SEKUENTZIA.....	29
IRUDIA 17: TENTSIO EGOERAREN ZONALDEAREN ARABERA TENTSIO EGOERA ESTATIKO BALIOKIDEAREN ADIERAZPEN GRAFIKOA.....	30

b. AKRONIMOAK

Oyp: Materialaren isurpen tentsioa.

Ou: Materialaren haustura tentsioa.

Og: LCF(Low Cycle Fatigue) eta HCF(High Cycle Fatigue) zonaldeak bereizten dituen tentsioa.

Oe: Piezaren neke limitea.

Ne: bizitza infinitua eta finituaren arteko muga markatzen duen ziklo kopurua.

Ng: LCF(Low Cycle Fatigue) eta HCF(High Cycle Fatigue) zonaldeak bereizten dituen ziklo kopurua.

CS: Segurtasun koefizientea.

I eta Ip: Piezaren inertziak.

Taum eta Taur: Tentsio ebakitzailaren batz besteko eta parte alternoa.

Taumax eta Taumin: Tentsio ebakitzaille maximoa eta minimoa.

Om eta Or: Tentsio normalaren batz besteko eta parte alternoa.

Omax eta Omin: Tentsio normal maximoa eta minimoa.

f, w eta T: Maiztasuna, abiadura angeluarra eta periodoa, hurrenez hurren.

r, R, theta eta γ : piezaren erradioa, aztertu nahia den puntuaren distantzia piezaren zentrotik erradialki neurtuta, aztertu nahi den puntuak ardatz bertikalarekiko osatzen duen angelua eta aztertu nahi den puntuaren altuera piezaren zentrotik norabide bertikalean neurtuta.

S1max, S2max, S3max, S1min, S2min, S3min: Tentsio nagusi maximoak eta minimoak.

O1r, O1m, O2r, O2m, O3r, O3m: Tentsio nagusien parte alternoa eta batz bestekoa.

z1max, z2max, z3max, z1min, z2min, z3min: Mohr-en zirkunferentzien zentroen balio maximoak eta minimoak.

r1max, r2max, r3max, r1min, r2min, r3min: Mohr-en zirkunferentzien erradioen balio maximoak eta minimoak.

Oeqm eta Oeqr: MS metodoan, Von Mises-en irizpidea aplikatzean lortzen diren tentsio baliokideak, batz bestekoa eta alternoa.

O1eeq, O2eeq eta O3eeq: SM metodoan, Soderberg-en irizpidea aplikatzean lortzen diren tentsio nagusi baliokide estatikoak.

Nms eta Nsm: MS eta SM metodoak aplikatzean piezak huts egingo duen momentuan igarotako ziklo kopurua

OeqMS eta OeqSM: MS eta SM metodoak aplikatzean lortzen diren tentsio uniaxial baliokideak.

c. TAULAK

TAULA 1: ANALISI ETA HUTSEGITE MOTEN KASU EZBERDINAK.	11
TAULA 2: NEKE ANALISIRAKO METODOEN ARTEKO KONPARAKETA.	12
TAULA 3: ALTERNATIBA EZBERDINEN KONPARAKETA.	15
TAULA 4: SM ETA MS METODOAK ZEIN KASUTARAKO ERABILI BEHAR DIREN ADIERAZTEN DUEN TAULA.	28
TAULA 5: EMAITZEN KONPARAKETA	42

3.SARRERA

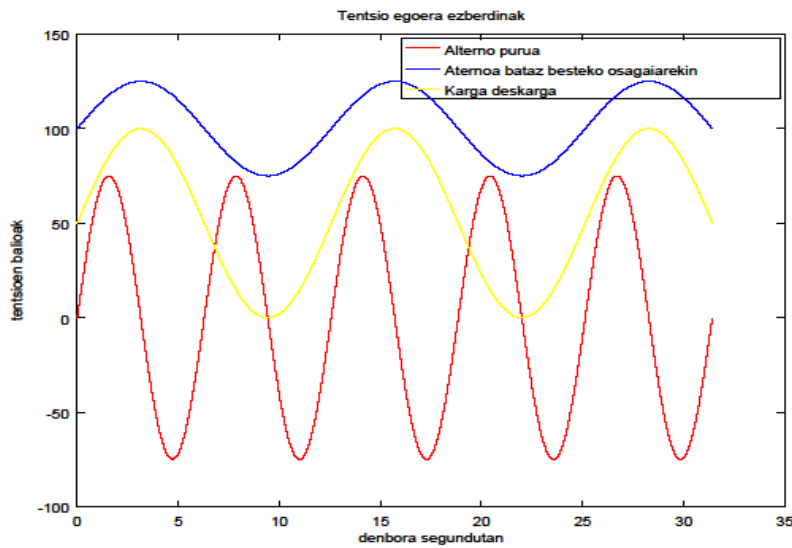
1830. urtean trenbidearen agerpenarekin batera neke analisiaren lehenengo urratsak ematen hasi ziren, eta hala eta guztiz ere, gaur egun etengabeko aldaketan eta ikerkuntzan ari den eremua da; 200 urteko metodo, entsegu eta ikuspegi anitzen ondoren ere ikertzaileen artean desadostasunak sortzen dituen saila izanik, lortutako emaitzen dispersio estatistikoa, eredu sinplifikatiboetan erabilitako irizpide ezberdinak eta beste hainbat ezjakintasunen agerpena direla-eta.

Sortutako programa osagarri honek tortsio eta makurdurapean lan egiten duten pieza zilindrikoen neke analisia egitea ahalbidetzen du, erabilera errazekoa izanda eta emaitzak argi adieraziz, beti ere kontuan izanik nekean lortutako emaitza numerikoak ez direla zehatzak izango aurretik aipatutako ezjakintasunengatik. Matlab software-aren bidez eratuko den programa honen erabilera mugatua badirudi ere, gaur egungo ia makina guztiek ardatzak edukitzen dituzte mugimenduak transmititzeko, zilindro forma dutenak eta espezifikoki nekeagatik huts egin ez dezaten diseinatzen dira, beraz, kasu horietarako piezaren aurre-diseinu bat egiteko oso erabilgarria izan daiteke.

Proiektu honetan nekearen inguruko teoria ezberdinak eta analisirako kalkulu guztiak justifikatuko dira, aurrerago argituko diren ikuspegi klasikoan oinarrituz , jarraitutako pausoak eta hauek ulertzeko beharrezkoa den oinarri teorikoa azalduz; baita programan erabilitako akronimoen zerrenda aurkeztuko da, ulergarriagoa izan dadin.

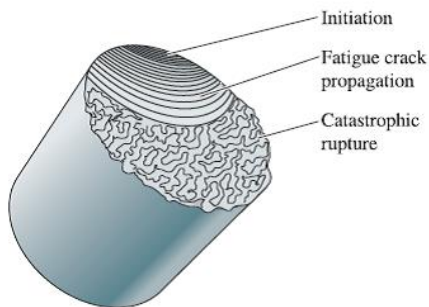
4. TESTUINGURUA

Nekegatiko hutsegitea isurpen tentsioaren balio baino txikiagoak diren tentsio egoera alternoetan ematen da, hau da, denboran aldakorrek diren tentsio egoeratan. Hainbat herrialdeetan egindako ikerkuntzek piezen ia %90-ak nekegatik huts egiten dutela ondorioztatu dute, beraz, produktuko diren piezen neke analisiak egitea ezinbestekoa da gaur egun.



Irudia 1: Denboran aldakorrek diren tentsioen adierazpena.

Aurretik aipatutako denboran aldakorrek diren tentsioak uneren batean piezan pitzadura bat agertzea eragiten dute, gehienetan piezaren gainazalean emango dena, bertan eman oi delako tentsio baliorik handiena. Denborak aurrera egin ahala, pitzadura piezan barrena hedatuko da piezaren sekzioa txikituz, estatikoki huts egin arte.



$$\sigma_{nom} = \frac{F}{A} \geq \sigma_{onargarria} = \frac{\sigma_{yp}}{CS}$$

$$\sigma_{nom} = \frac{F}{A} \geq \sigma_{onargarria} = \frac{\sigma_{ut}}{CS}$$

Irudia 2: Nekegatiko hutsegitearen faseak.

Hala ere, ez da beharrezkoa tentsio alternoko egoera bakoitzean neke analisia egitea, hainbat kasutan analisi estatikoa egitearekin nahikoa izango baita. Oso garrantzitsua izango da aztertzaileak zein kasutan zer analisi mota egin behar duen jakitea denbora nahiko aurreztu daitekeelako, izan ere, neke analisia egitea eta modelizatzea askoz ere konplexuagoa izango da. Horregatik analisi dinamikoa bat egin beharrean estatiko batekin piezaren analisia egin daitekeela jakiteak abantaila handia izango litzateke. Kasu bakoitzean zein analisi egin behar den jakiteko eta piezak nola huts egingo duen jakiteko hurrengo taula erabili daiteke piezari honako itxura duen indarra aplikatu zaiola jakinik:

$$F(t) = Fm + Fr * D(\omega) * \text{sen}(\omega * t)$$

Non ω laneko maiztasuna den, ω_1 piezaren lehenengo maiztasun naturala eta moteldura $D(\omega)$ izanik.

ω/ω_1	Fr/Fm	MOTELDURA		ANALISI ESTRUKTURALA	HUTSEGITE MOTA
Txikia	Txikia	Edozein	=>	Estatikoa	Estatikoa
Txikia	Ez txikia	Edozein	=>	Kuasiestatikoa	Nekea
Ez txikia	Txikia	Txikia	=>	Estatikoa dinamikoa baino probableagoa	Estatikoa probablea, nekea oso inprobablea
Ez txikia	Txikia	Handia	=>	Estatikoa oso probablea	Estatikoa oso probablea
Ez txikia	Ez txikia	edozein	=>	Dinamikoa	Nekea

Taula 1: Analisi eta hutsegite moten kasu ezberdinak.

Neke analisia egitea beharrezkoa den kasuetan ezinbestekoa izango da piezaren hainbat ezaugarri ezagutzea, hala nola, materiala, gainazal tratamendua, laneko temperatura, piezaren tamaina... baita laneko baldintzak zein izango diren; indar axialak, makurtzaileak, bihurtzaileak edo hiru hauen konbinaketa jasango duten. Neke analisiaren funtsa denborarekin aldakorrek diren tentsio egoerak, tentsio egoera estatiko baliokide batzuekin ordezkatzeta da eta dagokien tentsio onargarriarekin konparatzea, material hauskorretan haustura tentsioa eta harikorretan isurpen tentsioa izanik, piezak nekeagatik huts egingo duen ala ez jakiteko.

Lan honetan azalduko diren kalkulu metodoak material metalikoentzat soilik izango dira aplikagarriak. Hauetatik lortutako emaitzek produzituko diren piezen aurre dimentsionaketa egitea errazten dute, izan ere, prototipoak inolako informaziorik gabe eraikitzeak eta entseiatzeak kostu ekonomikoaz gain denbora kostu handia edukiko du. Hala ere, kasu hauetan ere prototipoak entseiatzea beharrezkoa izango da, baina egin beharreko entseiu kantitatea askoz txikiagoa izango da. Plastikoak, konpositeak... eta

antzeko materialak ordea, guztiz entseiatu egiten dira, aurre ikusi ezin diren portaerak izaten baitituzte.

Nekearen kalkulurako 3 metodo mota edo ikuspegi daude: ikuspegi klasikoa, ikuspegi globala eta plano kritikoan oinarritutako metodoak.

	KLASIKOA	GLOBALA	PLANO KRITIKOAN ONARRITUTAKO METODOAK
<i>ABANTAILAK</i>	Sinpleenak	Bataz besteko tentsio ebakitzailaren eragina mespretxatzen da eta tentsio egoera hidrostatiakoak kalkulatu daitezke	Norabide nagusien aldaketak nekean duen eragina aztertzeko gai, aurreko biek ez
<i>DESABANTAILAK</i>	Ezin daitezke tentsio egoera hidrostatiakoak kalkulatu	Kalkulu nahiko konplexuak	Konplexuenak, neke analisia aditua izan beharra

Taula 2: Neke analisirako metodoen arteko konparaketa.

Proiektu honetan azalduko den programak ikuspegi klasikoan oinarritutako bi metodoak jarraituz burutuko du neke multiaxialaren analisia. Metodo hauek Soderberg-Mises eta Mises-Soderberg izango dira, aplikatutako irizpideen ordenaren arabera izendatuak eta harrigarria badirudi ere, emaitza ezberdinetara helduz. Metodo hauen garapen teorikoa aurrerago azalduko da, soluzioaren analisiaren atalean, hain zuzen.

5. IRISMENA ETA HELBURUAK

Aurretik aipatu den bezala, proiektu honetan neke multiaxialaren analisirako programa bat sortuko da. Ikuspegi klasikoko bi metodoetan oinarrituz tentsio egoera multiaxiala, makurdura eta bihurtura hain zuzen, jasaten duen pieza zilindrikoen neke analisia burutzeko aukera ematen du.

Programaren helburu nagusia, erabiliko duen pertsonak planteatu ditzakeen egoera ezberdinetarako neke analisia zuzen burutzea da, piezak jasango dituen tentsioak kalkulatzu eta grafikoki adieraziz. Programak ematen dituen emaitzen fidagarritasuna bermatzeko piezaren karga egoera berarentzat eskuzko kalkulua eta ordenagailu bidezkoa burutuko dira, bi bideetatik lortutako emaitzak alderatuz.

Alderdi ekonomikoari dagokionez, merkatuan aurkitu daitezkeen elementu finituen programek ez bezala sortutako programa hau doakoa izango da.

Nekearen analisiaren saila oso zabala denez programaren erabilera mugatua da, pieza zilindrikoetara zuzendua baitago, hala ere, oso interesgarria izan daiteke irakaskuntzako arloan, alde batetik, ariketak erraz eta denbora laburrean ebazteko eta bestetik, tentsio egoera aldakorrek piezetan duten eraginaz jabetzeko. Irakaskuntzaz at, pieza zilindrikoak ekoizten dituzten enpresentzako ere erabilgarria izan daiteke programa hau, neke analisi sinple batekin piezen aurredeinua lortu daitekeelako.

Programa honek dituen oinarriak erabiliz etorkizunean programa osoago bat eratu daiteke, nekeगतिको hutsegitearen analisirako metodo klasikoek gain beste batzuk barneratuz, ikuspegi globaleko metodoak edota plano kritikoan oinarritutako metodoak besteak beste; programaren erabiltzaileak metodo ezberdinekin lortutako emaitzak alderatzeko aukera izanik eta horrela, metodo bakoitzean erabilitako irizpide sinplifikatiboek eta hurbilpen estatistikoek nekeगतिको hutsegitean duten eragina aztertuz.

Metodo berriak gehitzeaz ez ezik, proiektu honetan sortutako programari geometria ezberdinetako piezak kalkulatzu ahalbidetu dezakeen hobekuntza egin dakiok, piezak zilindrikoek gain merkatuan edo irakaskuntza sailean aurki daitezkeen beste geometria sinpleetako piezak (esferikoak, pieza lauak...) aztertzeko aukera izanik.

6. PROIEKTUAREN ONURAK

Alderdi tekniko edo zientifikoei dagokienez proiektu honetan sortuko den programak ardatzen neke analisi multiaxiala modu erraz eta argian egitea ahalbidetuko du. Erabiltzaileari datu ezberdinak eskatuz, hala nola, materialaren propietate estatikoak, neke limitean, aplikatutako momentu makurtzaile eta bihurtzailearen balioak, ardatzaren erradioa etab., programa gai izango da segundo batzuen buruan eta metodo klasikoetan oinarrituz, Soderberg-Mises eta Mises-Soderberg metodoak, ardatzaren neke analisia burutzeko, ardatzak bizitza infinitua edukiko badu ala ez kalkulatzuz, eta bizitza finitua edukitzekotan, huts egitea emango den momentuan igarotako ziklo kopurua kalkulatzeko du.

Gainera SM eta MS metodoak aplikatzean erabilitako metodologia eta kalkuluak urratsez urrats pantailaratuko dira, modu argian eta irudi esplikatiboak erabilia. Baita ere, ardatzak jasandako tentsioak eta tentsio nagusiak denboran nola aldatuko diren erakusten duten grafikoak aurkeztuko dira eta tentsio nagusien balio maximoen eta minimoen Mohr-en zirkunferentziak ere pantailaratuko dira.

Sortutako programaren erabilera erraza eta sinplea izango da, aparteko gaitasunik edo ezagutzarik eduki gabe ere egoki erabiltzeko arazorik ez izanik, erabiltzaileak eskatutako datuak sartuta programak berak neke analisia burutuko baitu.

Alderdi ekonomiko batetik begiratuta, neke analisiak burutzen dituzten elementu finituetako programak baino askoz merkeagoa da, dohainikoa baita. Hala ere, programa erabiltzeko Matlab lizentzia izatea beharrezkoa da, irakaskuntza alderdian unibertsitateko ordenagailuetan aurkitu eta erabili daiteke eta enpresa munduan, Matlab lizentzia erostea programa espezifikoen lizentziak erostea baino askoz merkeagoa izango da.

7.ALTERNATIBEN ANALISIA

Proiektu honetan sortutako programa erabili gabe neke analisia burutu nahi bada hiru modutan egin daiteke, eskuzko kalkuluaren bidez, neke analisi sakona egitea ahalbidetzen duten elementu finituetako programak erabiliz eta analisi esperimentalak.

Atal honetan hiru alternatiba hauek azaldu eta konparatuko dira taula baten bitartez, bakoitzak dituen abantailak eta desabantailak aurkeztuz:

	ESKUZKO KALKULUA	ELEMENTU FINITUEN PROGRAMAK	ANALISI ESPERIMENTALA
DESKRIBAPENA	Pieza geometria sinpleen neke analisia metodo ezberdinak erabiliz eskuz ebatzea.	Elementu finituetako programen bitartez neke analisia burutzea.	Piezaren prototipoak erabiliz tentsio egoera desberdinak entsegatzen dira
ABANTAILAK	Prozesu kalkulua guztia jarraitu eta ikusteko aukera ematen du.	Matlab softwareak baino potentzia handiagoa: 1.Azkarragoa 2.Material eta geometria ugari aztertze aukera 3.Metodo gehiago 4. Piezaren puntu edo nodo guztietan aztertu	Nekeak piezan duen eragina zuzenean aztertze aukera
DESABANTAILAK	1.Denbora nahiko behar 2.Akatsak kalkuluetan 3.Grafikoak eskuz egin beharra 4.Zehaztasuna galdu 5.Geometria sinpleak soilik	1.Programa erabiltzeko aditua izan beharra: Erabiltzaileak trebatzeko dirua eta denbora behar da 2.Matlab lizentzia baino garestiagoa	Garestia, entseguetarako prototipoak eta ekipoak eduki beharra

Taula 3: Alternatiba ezberdinen konparaketa.

Ondorioz, eskuzko kalkulua geometria sinpleko piezen analisia egiteko soilik izango da egokia, baita irakaskuntzako arloan neke analisiak egiten hasi berri direnentzat, metodo ezberdinetan jarraitutako pausuak eta egindako kalkuluak ulertzeko. Elementu finituetako programek berriz, edozein geometriako piezen tentsio egoerak kalkulatzeko gai direnez eskuzko kalkuluak baino malgutasun handiagoa izango du baina analista aditua izatea eta programen lizentziak erostea beharrezkoa izango da. Entsegu esperimentalak egitea garestia bada ere, nekeak piezetan daukan eragin zuzenean aztertzea ahalbidetzen du.

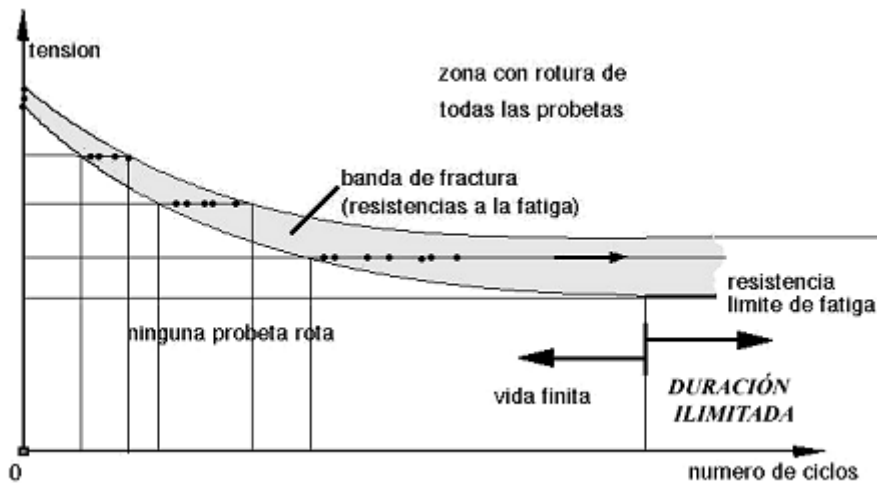
8. SOLUZIOAREN AZALPENA.

8.1. OINARRI TEORIKOA:

Atal honetan programak neke analisirako erabiltzen dituen metodoak azalduko dira baita hauek ulertzeko beharrezkoak den oinarri teorikoa eta kalkuluak.

8.1.1. S-N kurba:

Makurdura alternoko probeten neke entseguetatik lortuta emaitzak dispersio banda baten barnean batzen dira, S-N kurban hain zuzen, tentsioa hutsegitea ematen den ziklo kopuruarekiko adieraziz. Hala ere, ingenieritzako arloan egiten diren kalkuluak deterministak direnez banda hori tarteko kurba batera sinplifikatzen da, Q hutsegite probabilitate bati dagokiona, %50-koarekin lan egitea ohikoa izanik. Beheko irudian altzairu baten S-N kurba erakusten da.



Irudia 3: Altzairu baten S-N kurba.

8.1.2. Neke limitea:

Altzairu eta titaniozko aleazioen S-N kurbak 10^6 eta 10^7 ziklotik aurrera zonalde horizontal bat aurkezten dute, neke limitea adieraziko duena.

Beraz esan daiteke, tentsio balio batetik behera ez da nekeagatiko hutsegitea gertatu, tentsio balio horri neke limitea deritzo, eta piezak bizitza infinitua edukiko duela esaten da. Piezak diseinatzerakoan neke limitearen balio kontuan hartu beharra dago piezak bizitza infinitua izan dezan bilatzen bada.

Esperimentalki lortutako emaitzetatik ondorioztatu da material baten neke limiteak balio hauek hartzen dituela $N_e=10^6$ denerako:

$$\sigma_e = 0,5 \cdot \sigma_{ut} \quad \sigma_{ut} \leq 1400 \text{ MPa}$$

$$\sigma_e = 700 \text{ MPa} \quad \sigma_{ut} > 1400 \text{ MPa}$$

8.1.3. Trantsizio tentsioa:

Probetak 1000 zikloren ondoren hutsegingo duen tentsioari trantsizio tentsioa deitzen zaio, izan ere, ziklo txikietako neke zona eta ziklo handiko neke zona banatzen baititu. Metodo klasikoak ziklo handiko neke zonako egoerak soilik hartzen ditu kontuan, ziklo txikietako neke zonan lortzen diren emaitzek dispersio oso handia baitute eta materialek portaera oso ezberdinak aurkezten dituztelako. Beraz, programa honetan erabilitako kalkulu metodoak ziklo handiko neke zonako egoerak soilik aplikatuko dira.

Esperimentalki konprobatu da: $Ng = 10^3$ denerako, $\sigma g \approx 0,9 * \sigma_{ut}$ izango dela.

8.1.4. Diagrama logaritmikoak. Basquin-en ekuazioa:

Funtzio logaritmikoen helburua S-N kurba zatika zuzena den funtzio batean bihurtzea da, S-N kurba logS-logN kurba batean bilakatuz, irudian ikusi daitekeen moduan.

Zatikako funtzioaren makurdura alternoko biritza finituko tartea Basquin-en ekuazioaren bidez definituko da:

$$\log \sigma = \beta * \log N + \alpha$$

Bertatik honako bi adierazpenak lortu daitezke:

$$\sigma = 10^\alpha * N^\beta$$

$$N = 10^{-\alpha/\beta} * \sigma^{1/\beta}$$

Non:

$$\alpha = \log \sigma g - \beta * \log Ng$$

$$\beta = \frac{\log \sigma g - \log \sigma e}{\log Ng - \log N g}$$

logS-logN diagraman Basquin-en kurbaren malda β izango da, negatiboa izanik biritza finituko tartean eta zero biritza infiniturako, hala ere, nekean Basquin-en kurbaren maldari $-1/\beta$ balioa esleitzen zaio eta ondorioz:

$$m = -\frac{1}{\beta}$$

Aurreko ekuazioak kontuan hartuta, Basquin-en ekuazioa, neke limitea duten

altzairuen makurdura alternoko neke erresistentzia eta iraupena erlazionatzen dituen, hurrengo bi tarteen bidez definituko da:

$$\sigma = \sigma_0 * N^{-\frac{1}{m}}$$

$$N = \left(\frac{\sigma_0}{\sigma}\right)^m \quad Ng < n < Ne$$

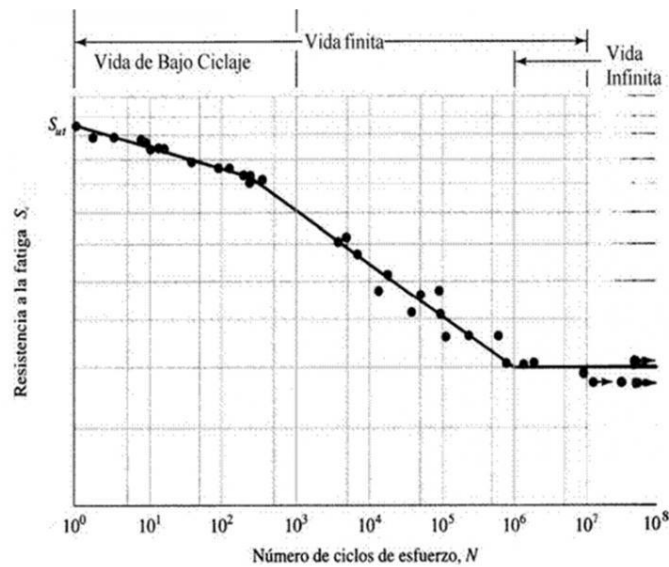
$$\sigma = \sigma_e \quad N > Ne$$

Non:

$$m = \frac{\log Ne - \log Ng}{\log \sigma_g - \log \sigma_e}$$

$$\sigma_0 = 10^\alpha$$

Esperimentalki lortutako emaitzetatik altzairuek nekearekiko duten portaera erakusten duen diagrama orientagarri orientagarria eratu daiteke, altzairuentzako “diagrama orokortua” deritzona eta 4.irudian adierazten delarik.



Irudia 4: Altzairuen S-N diagrama orokortua.

4. Irudiko AB lerroarentzat, $\sigma_{ut} < 1400MPa$ denerako hurrengo bi adierazpenak lortuko daitezke:

$$m = 11,752152$$

$$\alpha = \log(1,62 * \sigma_{ut}) \text{ izanik, } \sigma_0 = 1,62 * \sigma_{ut}$$

Aurreko ekuazioetan oinarrituz kurba sinplifikatua lortu daiteke:

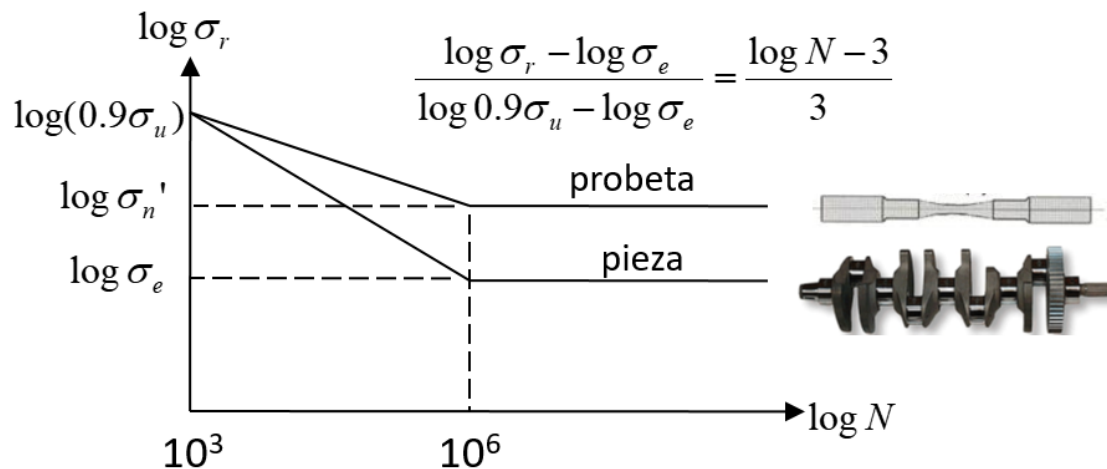
$$\sigma = 1,62 * \sigma_{ut} * N^{-0,08850908}$$

$$N = \left(\frac{1,62 * \sigma_{ut}}{\sigma} \right)^{11,752152} \quad 10^3 < n < 10^6$$

$$\sigma = 0,5 * \sigma_{ut} \quad N > 10^6$$

8.1.5. Marin-en formula:

Aurreko ataletan aipatutako irizpide eta erresistentzien balioak baita neke limitea laborategian probeta normalizatuei egindako entseguei dagokie, probeta hauek tamaina txikikoak eta zehaztasun handiz eginak, karga kontrolatuak jasaten dituztelarik. Entseguetan lortutako emaitzak pieza errealetan erabili ahal izateko koefiziente modifikatiboak batzuk erabiltzen dira, Marin-en koefizienteak hain zuzen. Koefiziente hauek unitatea baino balio txikiagoak izaten dituzten normalean, eta horrek piezen neke limitea probetena baino %20 txikiagoak izatea ekarri dezake, piezen logS-logN kurba behertuz hurrengo irudian erakusten den bezala.



Irudia 5: Probeta eta Piezaren S-N kurba orokortuak.

Piezaren balioak hauek izango dira:

$$\sigma_e = C_{e, mod} * \sigma_e'$$

$$\sigma_g = C_{g, mod} * \sigma_g'$$

$$N_e = C_{N, mod} * N_e'$$

Non σ_e' , σ_g' eta N_e' probeten entseguetan lortutako datuak diren.

$C_{e,mod}$ eta $C_{g,mod}$ koefizienteak beste koefiziente batzuen biderkaduraren emaitza dira, koefiziente bakoitza probeta eta piezen ezaugarrien arteko ezberdintasunak duen eragina kuantifikatuko duelarik.

$$C_{g,mod} = \prod_{i=1}^d c_{gi}$$

$$C_{e,mod} = \prod_{i=1}^d c_{ei}$$

d izanik kalkuluan kontuan hartuko diren koefizienteen kantitatea.

Koefiziente modifikatzaileek honako ezaugarri hauek hartzen dituzte kontuan:

- Zimurtasuna: c_s
- Lan egiteko era: c_t
- Piezaren tamaina: c_d
- Fidakortasuna: c_R
- Temperatura: c_T
- Soldadura: c_W
- Tratamendu mekanikoak: c_m
- Talkak: c_k
- Fretting-a: c_f
- Bestelako efektuak: c_v

Marinen formulak tentsio egoera alerno ezberdinetan lan egiten duten piezen neke limitea eskuratzeari ahalbidetzen du, probeten lortutako neke limiteen balioetatik abiatuz:

$$\sigma_e = c_{se} * c_{te} * c_{de} * c_{Re} * c_{Te} * c_{We} * c_{me} * c_{ke} * c_{fe} * c_{ve} * \sigma_e'$$

$$\sigma_g = c_{sg} * c_{tg} * c_{dg} * c_{Rg} * c_{Tg} * c_{Wg} * c_{mg} * c_{kg} * c_{fg} * c_{vg} * \sigma_g'$$

8.1.6. Batz besteko tentsioen eragina:

Batz besteko tentsioak σ_m , tentsio alernoetara σ_e gainezartzeak, materialek nekearekiko duten portaeran eragina dute. Tentsioaren batz besteko osagaia trakziozkoa bada, positiboa, materialak tentsioa alernoekiko duen erresistentziaren murrizpena gertatuko da, S-N kurba beheratuz. Konpreziozkoa bada ordea, guztiz

kontrakoa gertatuko da.

Aldagai berrien definiziorako terminologia eta balio espezifikoak erabiliko dira:

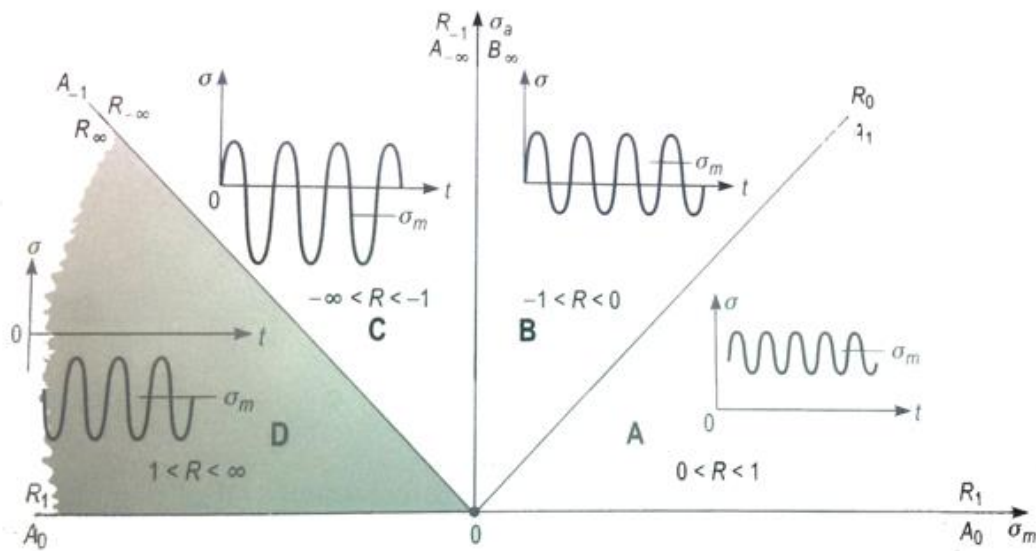
$$A = \frac{\sigma_a}{\sigma_m}$$

$$\Delta\sigma = \sigma_{max} - \sigma_{min}$$

A "anplitude arrazoia" eta $\Delta\sigma$ "tentsio tartea" izanik. σ_{min} negatiboa bada, balio hori beharrez $\sigma_{min} = 0$ izango dela suposatuko da, izan ere, tentsio negatiboek, konpreziozkoak, arrakalen hedapena oztopatzen dutelako eta nekearen ikuspegitik efektu onuragarria izango da.

σ_m nulua ez den kasuetarako entsegu berriak egitea beharrezkoa da, ezin daitekeelako σ_m nulua den tentsio alternoko entseguetatik lortutako emaitzekin erlaziorik lortu. Entsegatu daitezkeen kasu posibleak Haigh-en diagraman adierazten dira eta bertan 4 zona ezberdinu daitezke. D zonan teorikoki ez da hutsegiterik emango, zonalde honetan piezak konpresiopean lan egingo duelako sortuko diren arrakalak hedatzea oztopatuz:

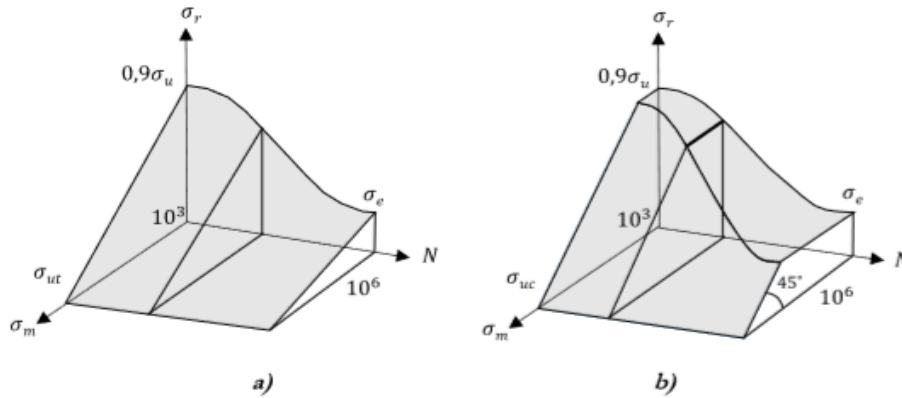
$$A = \frac{\sigma_r}{\sigma_m} \quad R = \frac{\sigma_{min}}{\sigma_{max}}$$



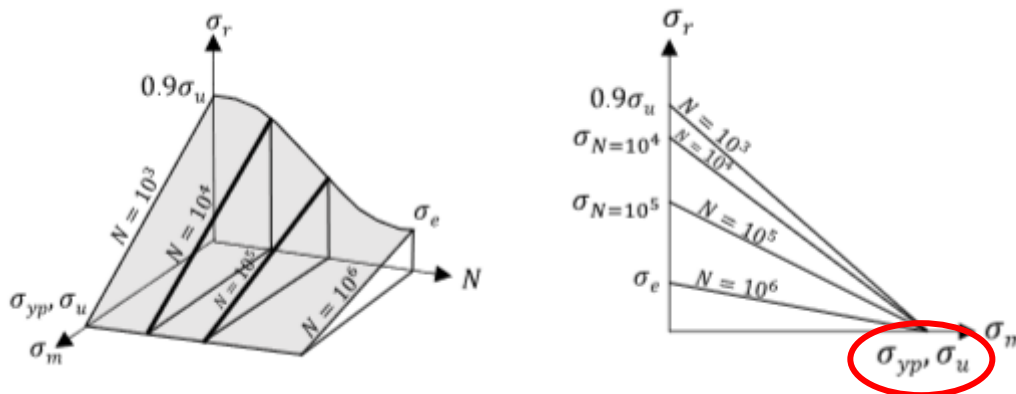
Irudia 6: Tentsio egoera ezberdinak erakusten dituen irudia.

8.1.7. Trakziozko bataz besteko tentsioentzako irizpideak:

Entseguetatik lortutako kurben trakzio aldeko ($\sigma_m > 0$) kurba zuzenetara hurbildu daitezke kalkuluak errazteko helburuarekin. Material motaren arabera hurbilpen metodoak edo irizpideak ezberdinak izango dira eta gainera, material mota berarentzako ere irizpide anitz aurkitu daitezke.

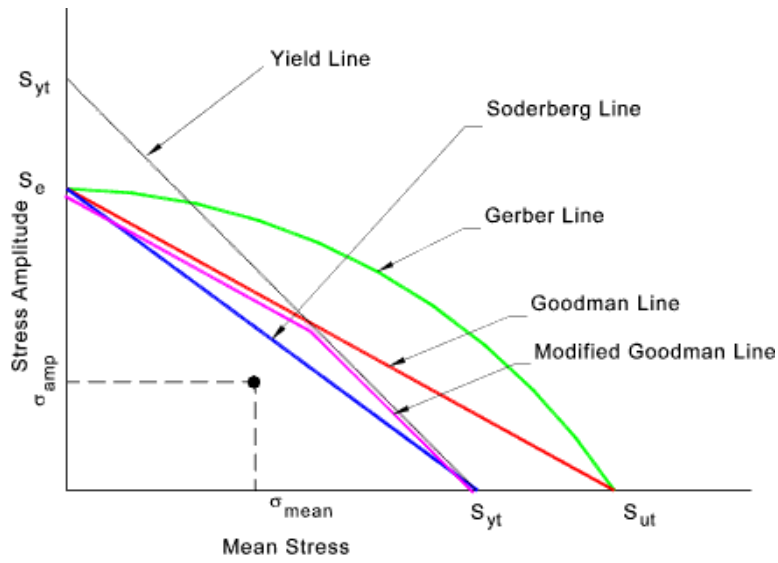


Irudia 7: Materialen harikor(a) eta hauskorretan(b) diagrama tridimentsionalak.



Irudia 8: Soderberg(σ_{yp}) eta Goodman(σ_u)-en irizpideak material harikorretan.

Trakzio alderako Goodman eta Soderberg-en irizpideak erabili ohi dira, triangeluen antzekotasunean oinarrituko direlarik. Hala ere, badira beste hainbat irizpide, hala nola, Gerber, Morrow eta Dietmann, 9 irudian aurkezten den moduan.



Irudia 9: Trakzio zonako irizpideen adierazpen grafikoa.

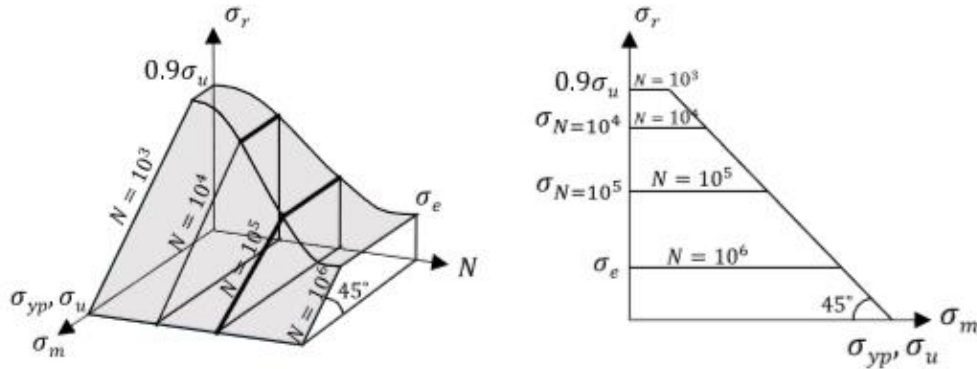
Goodman-en irizpidea hurrengoa izanik:

$$\frac{\sigma a}{\sigma e} + \frac{\sigma m}{\sigma ut} = 1$$

Soderberg-ek isurpen bidezko hutsegiteak saihesten dituen lerro zuzenaren ekuazioa planteatu zuen:

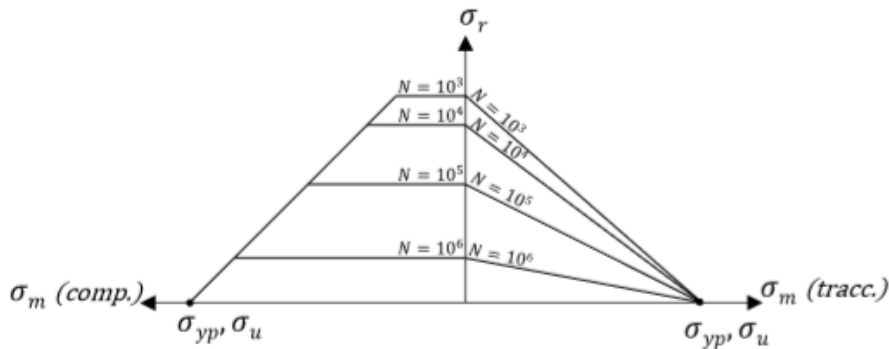
$$\frac{\sigma a}{\sigma e} + \frac{\sigma m}{\sigma yp} = 1$$

8.1.8. Konpresiozko batuz besteko tentsioaren eragina eta Haigh-en diagrama: Konpresiozko aldean diagraman bi zonalde bereizi daitezke: Alde batetik, lerro horizontala den atala, non $\sigma_N = \sigma_r$ eta beste alde batetik, 45° -tara dagoen lerroa, hutsegite estatikoa adierazten duena eta bere ekuazioa $\sigma_{yp} = \sigma_m + \sigma_r$ material harikorrentzat eta $\sigma_u = \sigma_m + \sigma_r$ material hauskorrentzat izango den.



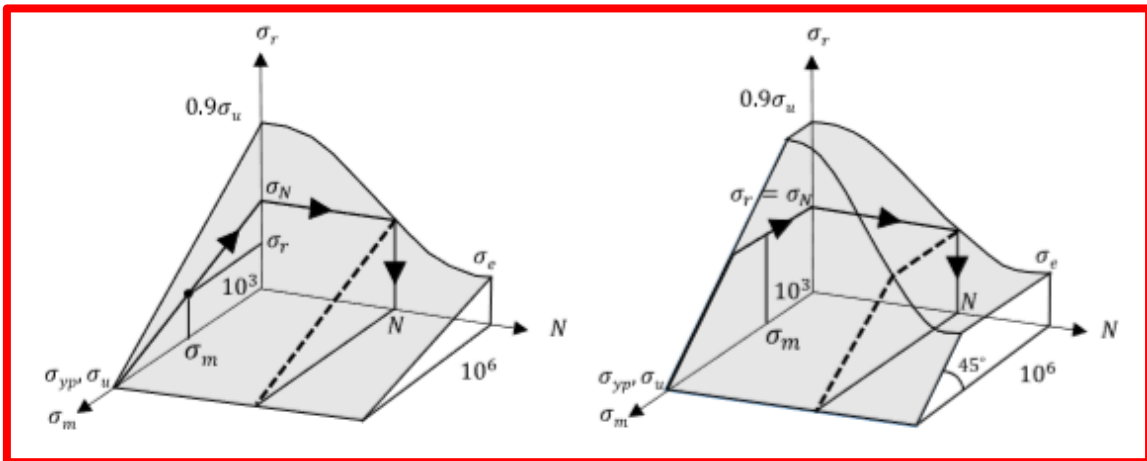
Irudia 10: konpresio zonako tentsio ezberdinetarako zuzen hurbilduak. Ezkerraldean, tridimentionala eta eskumaldean, bidimentionala.

Trakziozko eta konpresiozko digramak batuta, Haigh-en diagrama hurbildua lortu daiteke:

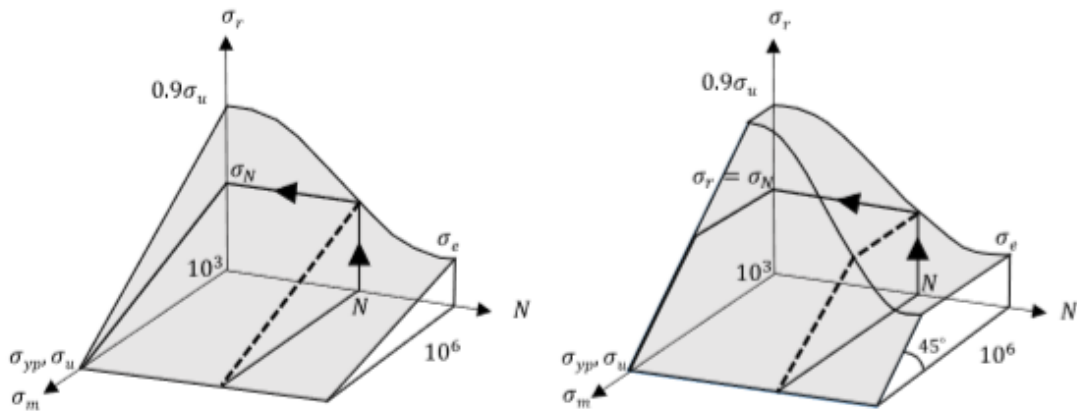


Irudia 11: Haigh-en diagrama hurbildua.

Neke analisisian bi problema ezberdin ebatzi daitezke: Alde batetik, tentsio egoera batetik piezak zenbat N zikloen kalkulua 12.irudian agertzen den moduan eta bestetik, N ziklo batetik abiatuz tentsio egoera kalkulatzeko 13.irudian erakusten den moduan, bigarren honek infinitu emaitza posible dituelarik. Programa honetan lehendabizi aipatutako problema mota ebatziko da:



Irudia 12: Material motaren arabera 1. problema mota, hutsegitea emango deneko N ziklo kopuruaren adierazpen grafikoa. Ezkerraldean, material harikorrentzako adierazpena eta eskumaldean, material hauskorrena.

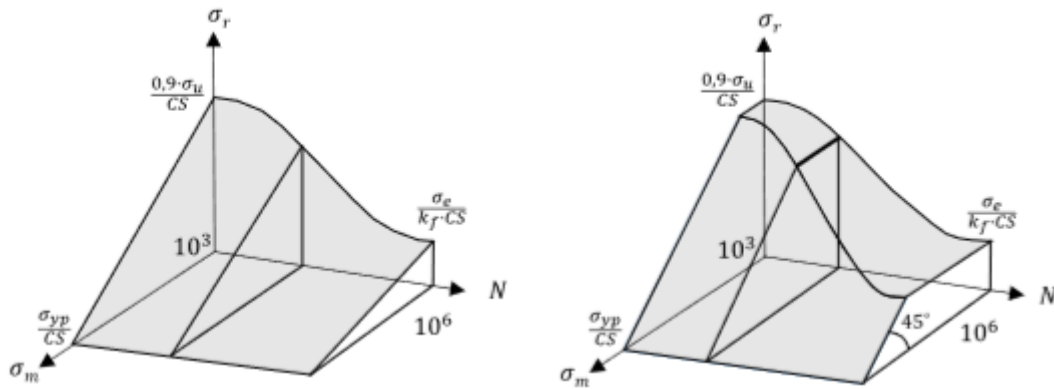


Irudia 13: Material motaren arabera 2. problema mota, hutsegitea emango deneko tentsioaren adierazpen grafikoa. Ezkerraldean, material harikorrentzako adierazpena eta eskumaldean, material hauskorrena.

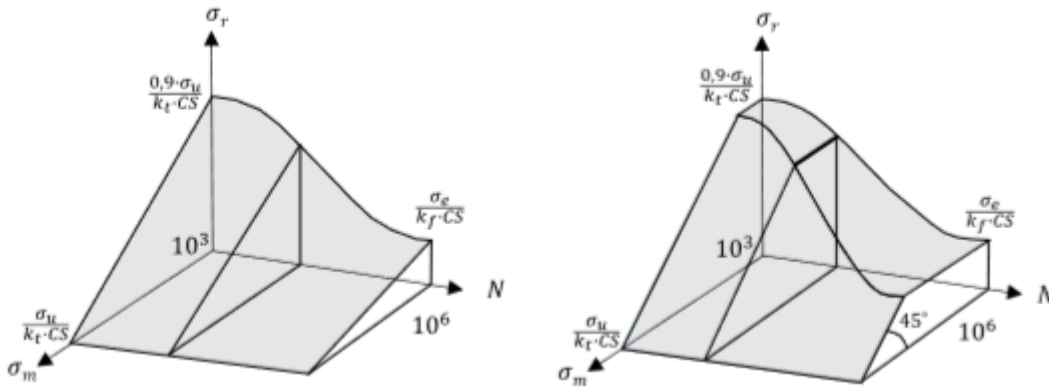
8.1.9. Tentsio kontzentrazioen efektua eta segurtasun koefizientea:

Tentsio kontzentrazioak zonak piezetan ez-jarraitasun geometrikoak daudenean ematen dira, eta kasu hauetan benetako tentsioak tentsio nominalak baino handiagoak dira eta horregatik, kurbaren minorizazioa egitea beharrezkoa da. Koefizientearen balioa kalkulatzeko hainbat taula eta metodo erabiltzen dira, piezaren forma eta lan egoeraren arabera izanik.

Beste alde batetik, segurtasun koefizientea erabiltzeak neke analisirako erabilitako sinplifikazio guztien eraginak kontuan hartzeko erabiltzen dira eta tentsio kontzentrazioekin gertatzen zen moduan hemen ere kurba minoratzea beharrezkoa izango da.



Irudia 14: Segurtasun koefizienteak kurbetan duen eraginaren adierazpen grafikoa. Ezkerraldean, material harikorrentzat eta eskumaldean, material hauskorrentzat.



Irudia 15: Tentsio kontzentrazio koefizienteak kurbetan duen eraginaren adierazpen grafikoa. Ezkerraldean, material harikorrentzat eta eskumaldean, material hauskorrentzat.

8.1.10. Tentsio multiaxiala:

Tentsio multiaxialen kasurako, ezin daitezke aurretik aipatutako tentsio egoera uniaxialeko metodoak zuzenean erabili baina badaude tentsio egoera baliokideak erabiliz metodo hauek erabiltzeko aukera.

Tentsio egoera multiaxialak ebazteko hainbat metodo badaude ere, hala nola, enpirikoak (espezifikoak eta klasikoak), enfoke globalekoak, plano kritikoan oinarritutakoak eta hausturaren mekanika, programa honetan metodo klasikoak soilik aztertuko dira.

Tentsio egoera multiaxial sinpleak ebazteko metodo klasikoien barnean 2 ezberdinu daitezke: SM eta MS metodoak, Soderberg-Mises eta Mises-Soderberg, hurrenez hurren, tentsio egoera baliokideak lortzeko erabilitako metodoen ordenari erreferentzia eginez. Bi metodoek emaitza ezberdinak eskaintzen dituztenez ondoren erakusten den 5.taularen arabera bata edo bestea erabiliko da:

PROBLEMA MOTA		METODOA
Iraupen infiniturako batz besteko bi tentsioak positiboak direnean	Preferentziarik ez badago	MS edo SM
	Kontserbakorra ez den metodoa	SM
Iraupen infiniturako batz besteko tentsioa bat edo biak positiboak direnean	Preferentziarik ez badago	MS edo SM
	Kontserbakorra ez den metodoa	MS
Iraupen finiturako batz besteko bi tentsioak positiboak direnean	Preferentziarik ez badago	MS edo SM
	Kontserbakorra ez den metodoa	SM
Iraupen finiturako batz besteko tentsioa bat edo biak positiboak direnean		MS

Taula 4: SM eta MS metodoak zein kasutarako erabili behar diren adierazten duen taula.

Aipatu beharra dago MS metodoa SM metodoa baino kontserbakorragoa dela eta horregatik, lortutako tentsio estatiko uniaxial baliokideak SM metodoan lortzen direnak baino handiagoak izango dira eta ondorioz, piezaren bizi iraupena txikiagoa izango da.

8.1.10.1. Soderberg-Mises:

Egoera multiaxialeko tentsio nagusiak ezagunak diren kasutik abiatuta, bakoitzaren batz besteko eta parte alternoa alegia:

$$\sigma_{1m}, \sigma_{1r}, \sigma_{2m}, \sigma_{2r}, \sigma_{3m}, \sigma_{3r}$$

Lehendabizi Soderberg-en irizpidea erabiltzen da tentsio nagusi estatiko baliokideak lortzeko kontuan izanik tentsio nagusi bakoitzaren batz besteko zatiaren ikurra, 3 kasu ezberdin aurkitu daitezke:

$$\sigma_m > 0 \text{ bada} \rightarrow \sigma_{eeq} = \sigma_m + \sigma_r * \sigma_{yp} / \sigma_e$$

$$\sigma_m < 0 \text{ eta } \sigma_r > \sigma_m \rightarrow \sigma_{eeq} = \sigma_r * \sigma_{yp} / \sigma_e$$

$$\sigma_m < 0 \text{ eta } \sigma_r < \sigma_m \rightarrow \sigma_{eeq} = \sigma_m - \sigma_r$$

Aurreko ekuazioak tentsio nagusi bakoitzarentzat aplikatu beharko lirateke. Ondoren, Von Mises-en irizpidea erabiltzen da tentsio egoera estatiko multiaxialaren baliokide den egoera estatiko uniaxiala lortzeko:

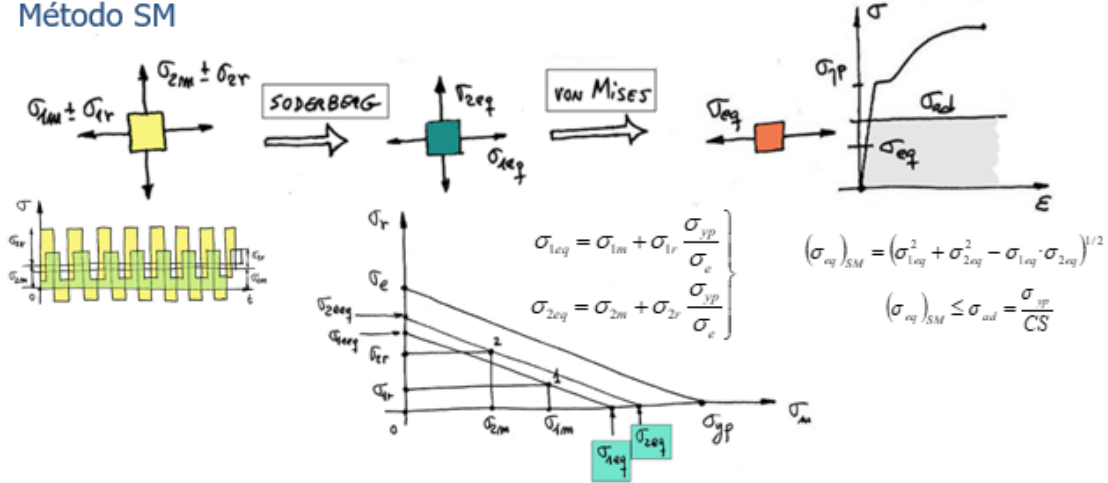
$$\sigma_{eqSM} = \sqrt{0.5 * ((\sigma_{1eeq} - \sigma_{2eeq})^2 + (\sigma_{1eeq} - \sigma_{3eeq})^2 + (\sigma_{2eeq} - \sigma_{3eeq})^2)}$$

Azkenik, lortutako tentsio uniaxial estatiko baliokidea piezaren isurpen tentsioarekin konparatu egiten da eta, segurtasun koefizienterik erabiltzekotan, kalkuluan kontuan harturik, suposatuz ez dela hutsegiterik emango tentsio baliokidea txikiagoa den kasurako:

$$\sigma_{eqSM} \leq \sigma_{yp} / c_s \text{ bada, ez da hutsegiterik emango}$$

Hutsegiterik ematen ez bada, piezak bizitza infinitua edukiko du. Hutsegitea ematen bada ordea, piezak momenturen batean nekegatik huts egingo du. Hutsegitea emango den momentuan igarotako ziklo kopurua kalkulatzeko lehendabizi σN tentsio baliokidea kalkulatu behar da eta ondoren, Basquin-en kurba erabilita N ziklo kopurua kalkulatu daiteke. Beheko irudian SM metodoaren sekuentzia erakusten da.

■ Método SM



Irudia 16: SM metodoaren sekuentzia.

8.1.10.2. Mises-Soderberg:

SM metodoaren alderantzizko prozesua erabiltzen da kasu honetan. Lehenik, Von Mises-en irizpidea erabiliz tentsio egoera uniaxiala lortzen da, dagokion parte alternoa eta batatz bestekoarekin:

$$\sigma_{eqm} = \sqrt{0.5 * ((\sigma_{1m} - \sigma_{2m})^2 + (\sigma_{1m} - \sigma_{3m})^2 + (\sigma_{2m} - \sigma_{3m})^2)}$$

$$\sigma_{eqr} = \sqrt{0.5 * ((\sigma_{1r} - \sigma_{2r})^2 + (\sigma_{1r} - \sigma_{3r})^2 + (\sigma_{2r} - \sigma_{3r})^2)}$$

Ondoren, Soderberg-en irizpidea aplikatuz tentsio egoera estatiko baliokidea lortu egiten da:

$$\sigma_{eqm} > 0 \text{ bada} \rightarrow \sigma_{eqMS} = \sigma_{eqm} + \sigma_{eqr} * \sigma_{yp} / \sigma_e$$

$$\sigma_{eqm} < 0 \text{ eta } \sigma_{eqr} > \sigma_{eqm} \rightarrow \sigma_{eqMS} = \sigma_{eqr} * \sigma_{yp} / \sigma_e$$

$$\sigma_{eqm} < 0 \text{ eta } \sigma_{eqr} < \sigma_{eqm} \rightarrow \sigma_{eqMS} = \sigma_{eqm} - \sigma_{eqr}$$

Azkenik, SM metodoaren konparaketa bera egiten da hutsegitea gertatuko den ala ez jakiteko, lortutako tentsio uniaxial estatiko baliokidea piezaren isurpen tentsioarekin

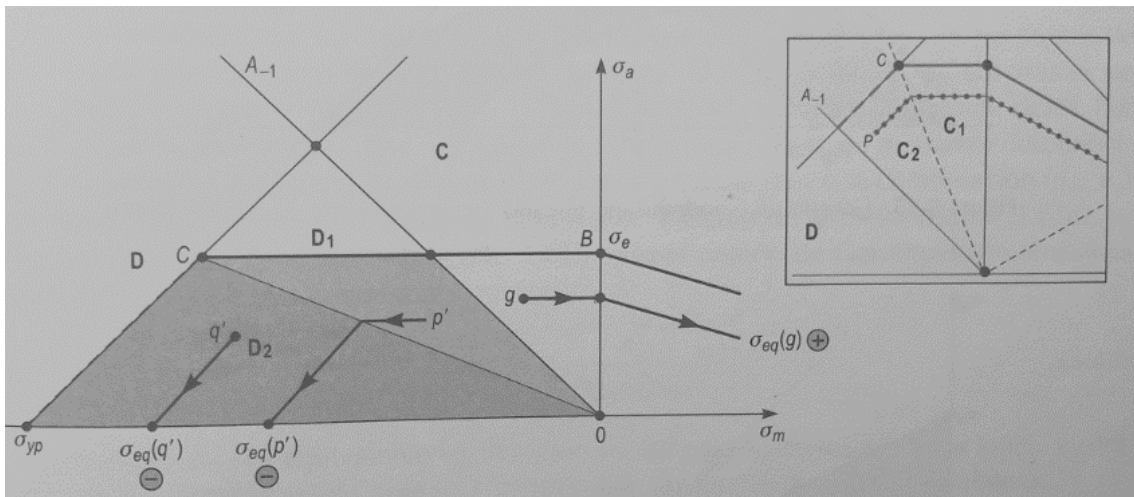
konparatuz, tentsio baliokidea txikiagoa den kasurako ez da hutsegiterik emango:

$$\sigma_{eqSM} \leq \sigma_{yp}/c_s \text{ bada, ez da hutsegiterik emango}$$

Hutsegiterik ematen ez bada, piezak bizitza infinitua edukiko du. Hutsegitea ematen bada ordea, piezak momenturen batean nekegatik huts egingo du. Hutsegitea emango den momentuan igarotako ziklo kopurua kalkulatzeko lehendabizi σN tentsio baliokidea kalkulatu behar da eta ondoren, Basquin-en kurba erabilia N ziklo kopurua kalkulatu daiteke.

8.1.10.3. Ikurren garrantzia:

SM metodoaren kasurako, Soderberg-en irizpidea aplikatzean lortzen den tentsio estatikoaren ikurrak garrantzia izango du ondoren aplikatuko diren kalkuluetarako. Beheko irudiko D1 eta D2 zonaldeko tentsio egoeretatik lortutako tentsio egoera estatiko baliokideek ikur negatiboa eramango dute, gainontzekoek berriz, positiboa.



Irudia 17: Tentsio egoeraren zonaldearen arabera tentsio egoera estatiko baliokidearen adierazpen grafikoa.

Zonaldearen arabera Tentsio baliokide estatikoak balio ezberdin bat edukiko du:

$$D2 \text{ zonan} \rightarrow \sigma_{eq} = |\sigma_m| + \sigma_r$$

$$C \text{ eta } D1 \text{ zonetan} \rightarrow \sigma_{eq} = \frac{\sigma_{yp}}{\sigma_e} \sigma_r$$

$$\sigma_m > 0 \text{ denerako} \rightarrow \sigma_{eq} \text{ Soderberg-en irizpidea erabiliz kalkulatu}$$

MS metodoa aplikatzean ordea, ikurraren garrantzia Von Mises-en irizpidea aplikatzean kontuan hartuko da. Tentsio baliokidearen ikurra positiboa izango baldin eta hurrengo ekuazio betetzen bada:

$$\sigma_1 + \sigma_2 + \sigma_3 > 0$$

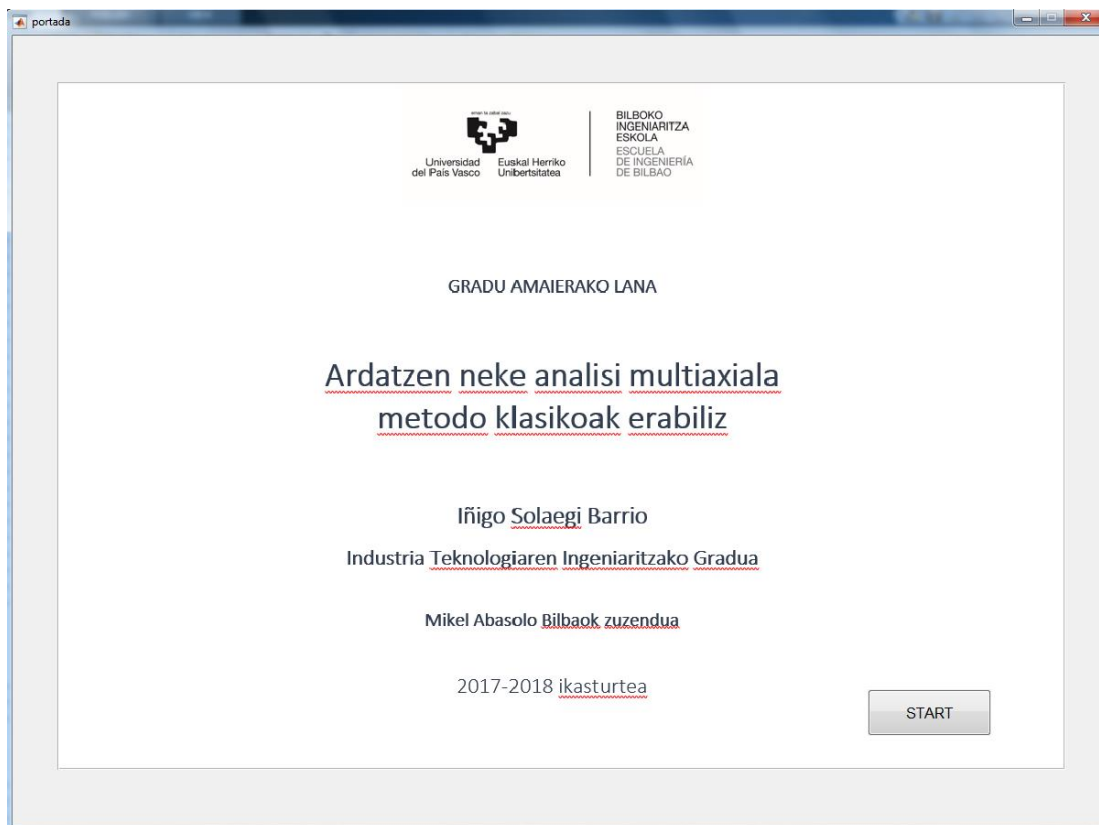
Ikurra negatiboa bada berriz:

$$\sigma_1 + \sigma_2 + \sigma_3 < 0$$

Hiru tentsio nagusien batuketa zero bada, hau da, tentsio egoera hidrostatikoa bada, Von Mises-en irizpideak huts egingo du eta horregatik ezin izango dira metodo klasikoak erabili, enfoke globalekoak edo plano kritikoan oinarritutako metodoak erabili beharko dira.

8.2. PROGRAMAREN DESKIRBAPENA

8.2.1. PORTADA



Programa ireki bezain laster portada irekiko da. Bertan programaren izena, egile eta zuzendariaren izena agertuko dira. *START* botoia sakatuz programan aurrera egin daiteke datuen lehiatila irekiz.

8.2.2. DATUAK

DATUAK SARTU:

NEKE KURBA :

Segurtasun koefizientea:

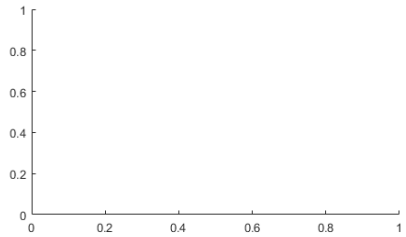
Ne :

Ng :

Piezaren neke limitea, σ_e (MPa):

σ_g (MPa):

NEKE KURBA SORTZEKO CLICK EGIN



MATERIALAREN PROPIETATE ESTATIKOAK:

Hautura tentsioa, σ_u (MPa):

Isurpen tentsioa, σ_{yp} (MPa):

GOMETRIA :

Ardatzaren erradioa (mm):

KARGAK :

Bataz besteko bihurtzailea (Nmm) :

Bihurtzaile alternoa (Nmm) :

Bataz besteko makurtzailea (Nmm) :

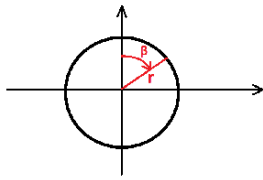
Makurtzaile alternoa (Nmm) :

Momentuen maiztasuna(Hz):

AZTERTU NAHI DEN PUNTUA :

r-ren balioa (mm, ikusi beheko irudia):

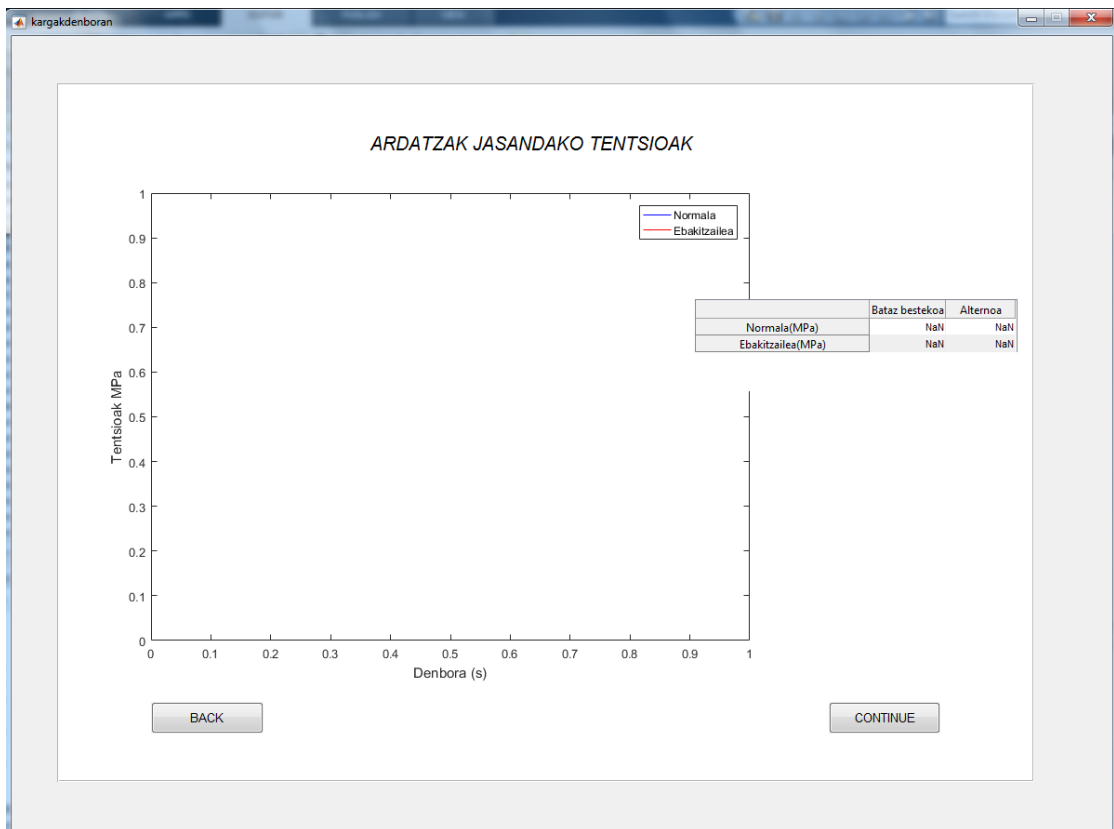
Theta-ren balioa (graduetan, ikusi beheko irudia):



KALKULATU

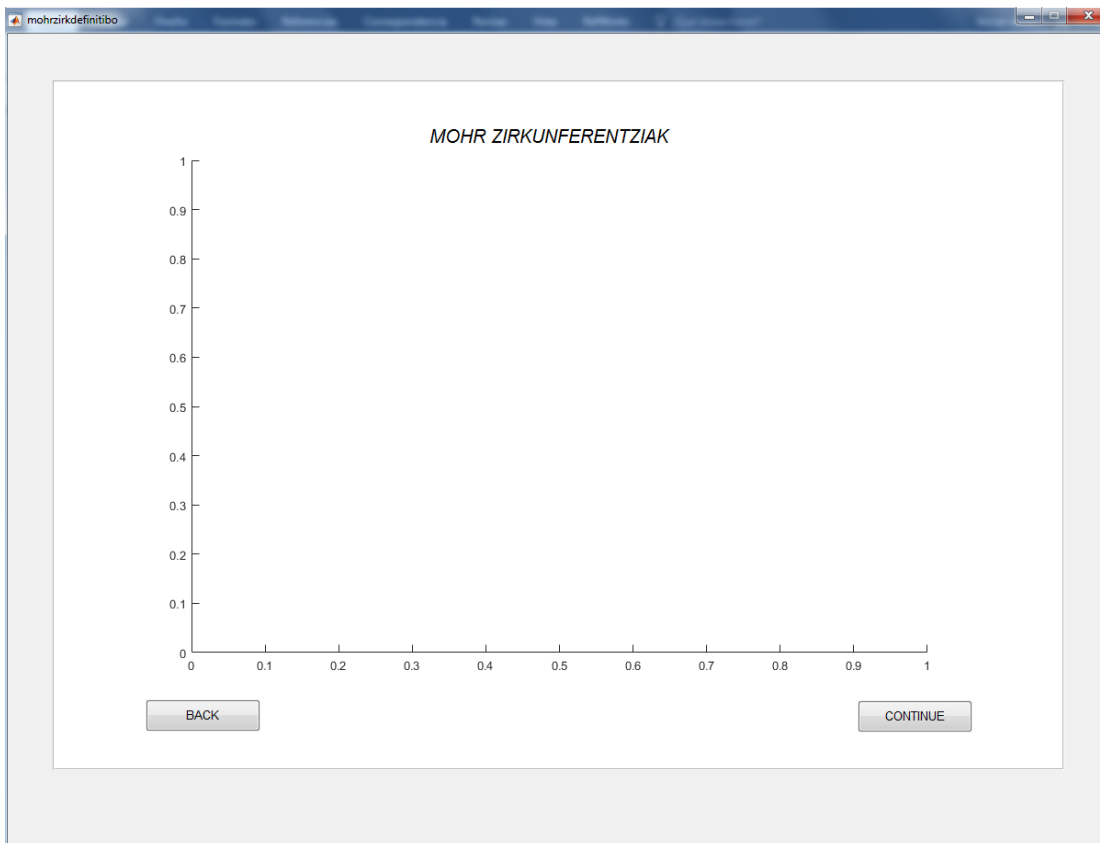
Datuen lehiatilan programaren erabiltzaileak eskatutako datuak sartuko ditu neke analisia egitea posible izan dadin. Lehenik eta behin neke kurba osatzeko datuak sartuko dira eta behin hori eginda, *neke kurba sortzeko click egin* botoia sakatzean neke kurba grafikoki irudikatuko da. Materialaren propietate estatikoak, geometria eta kargen datuak dagokien lekuan eta unitatekeekin sartuko dira. Azkenik, aztertu nahi den puntuaren datuak sartuko dira irudian agertzen diren erreferentziak erabiliz. Behin datu guztiak sartuta **KALKULATU** botoia sakatuz programaren hurrengo lehiatila irekiko da.

8.2.3. TENTSIOAK



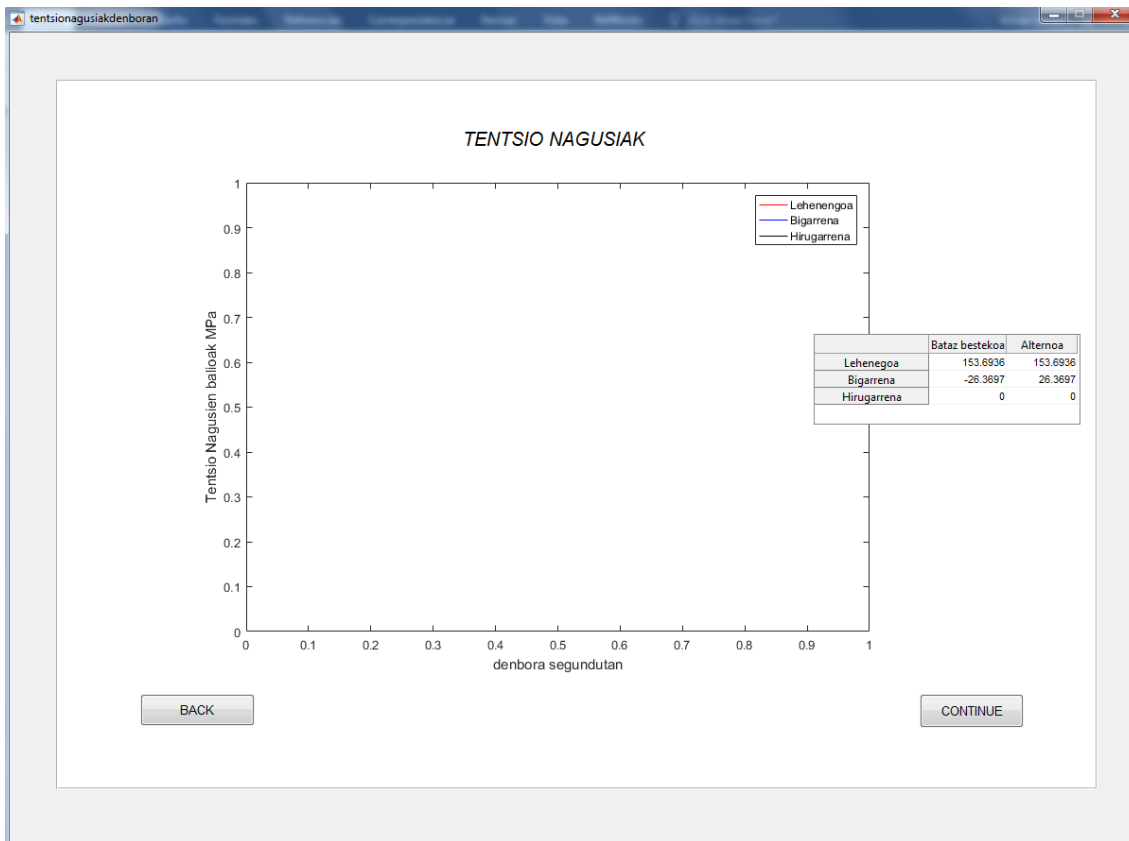
Lehiatila honetan tentsio normala eta ebakitzailea denboran nola aldatzen diren grafikoki erakusten dira. Eskumako taulan tentsioen batza besteko eta zati alternoaren balio numerikoak erakustiko dira. *BACK* botoia sakatzean aurreko lehiatila irekiko da eta *CONTINUE* botoia sakatzean ordea, programaren hurrengo lehiatila irekiko da.

8.2.4. MOHR-en ZIRKUNFERENTZIAK



Lehiatila hau tentsio lehiatilan *CONTINUE* botoia sakatzean irekiko da. Bertan Mohr-en zirkunferentziak irudikatuko dira, 2 tentsio egoera adierazgarrienenak, alde batetik, kolore gorritz tentsio maximoko egoeraren zirkunferentziak eta bestetik, tentsio minimoko egoeraren zirkunferentziak. *BACK* botoia sakatzean aurreko lehiatila irekiko da eta *CONTINUE* botoia sakatzean ordea, programaren hurrengo lehiatila irekiko da.

8.2.5. TENTSIO NAGUSIAK



Lehiatila hau Mohr-en zirkunferentziak erakusten diren lehiatilaren ostean irekiko da. Lehiatila honetan tentsio nagusiak denboran zehar hartuko dituzten balioak grafikoki adieraziko dira. Eskumako taulan berriz, 3 tentsio nagusien batuz besteko zatiaren eta zati alfernoaren balio numerikoak erakutsio dira. *BACK* botoia sakatzean aurreko lehiatila irekiko da eta *CONTINUE* botoia sakatzean ordea, programaren hurrengo lehiatila irekiko da.

8.3. PROGRAMA ERABILIZ PROBLEMA BATEN EBAZPENA

Problemaren datuak:

Piezaren neke limitea, $\sigma_e = 200\text{MPa}$

$\sigma_g = 900\text{MPa}$

$N_e = 1000000$ ziklo

$N_g = 1000$ ziklo

$CS(\text{segurtasun koefizientea}) = 1,5$

materialaren haustura tentsioa, $\sigma_u = 1100\text{MPa}$

materialaren isurpen tentsioa, $\sigma_{yp} = 800\text{MPa}$

Piezaren erradioa, $r = 10\text{mm}$

Momentu bihurtzailearen bataz besteko zatia, $B_m = 100000 \text{ Nmm}$

Momentu bihurtzailearen zati alternoa, $B_r = 100000 \text{ Nmm}$

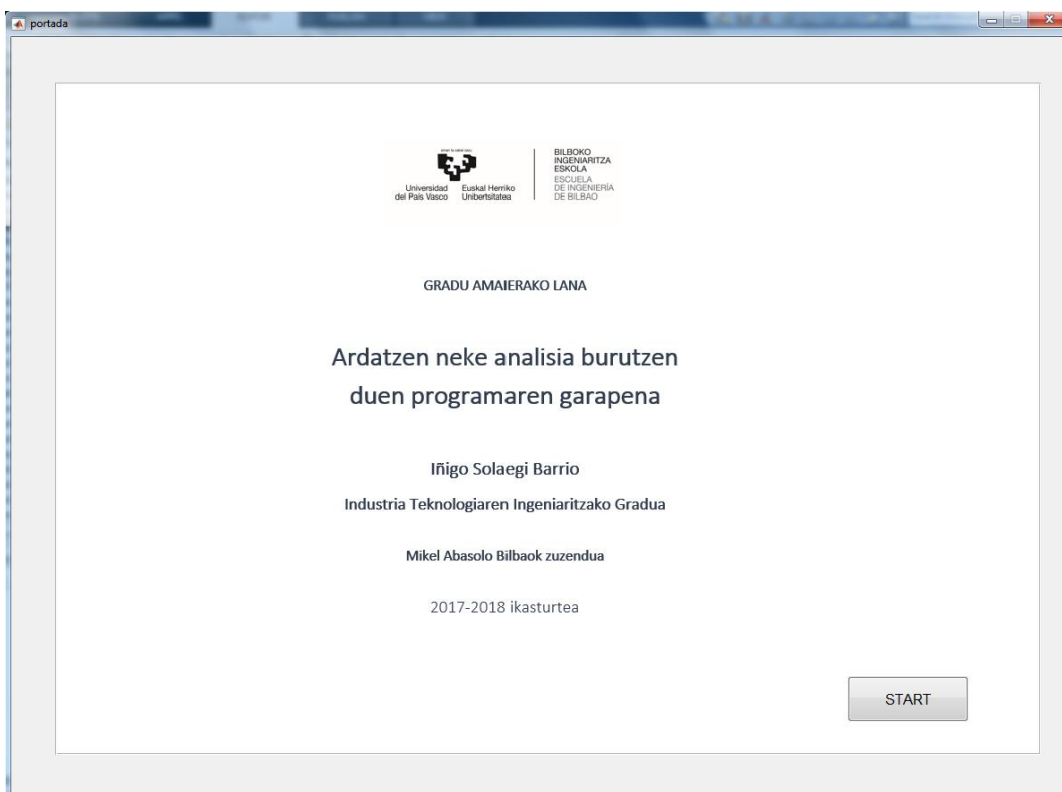
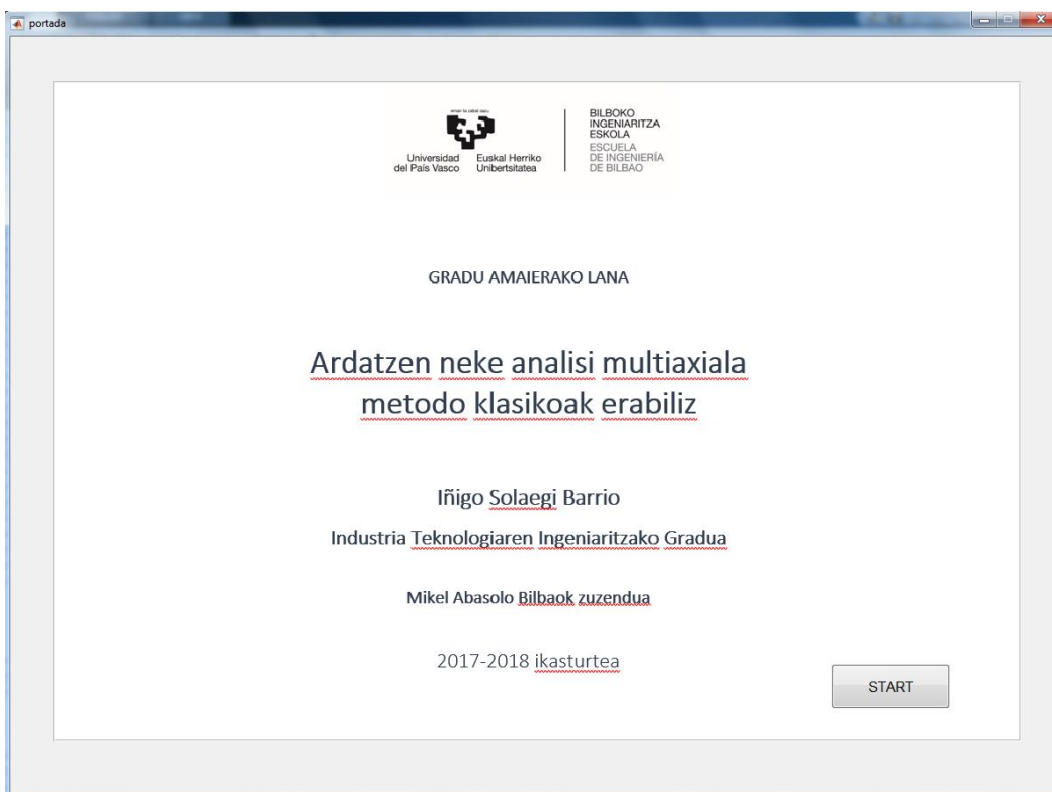
Momentu makurtzailearen bataz besteko zatia, $M_m = 100000 \text{ Nmm}$

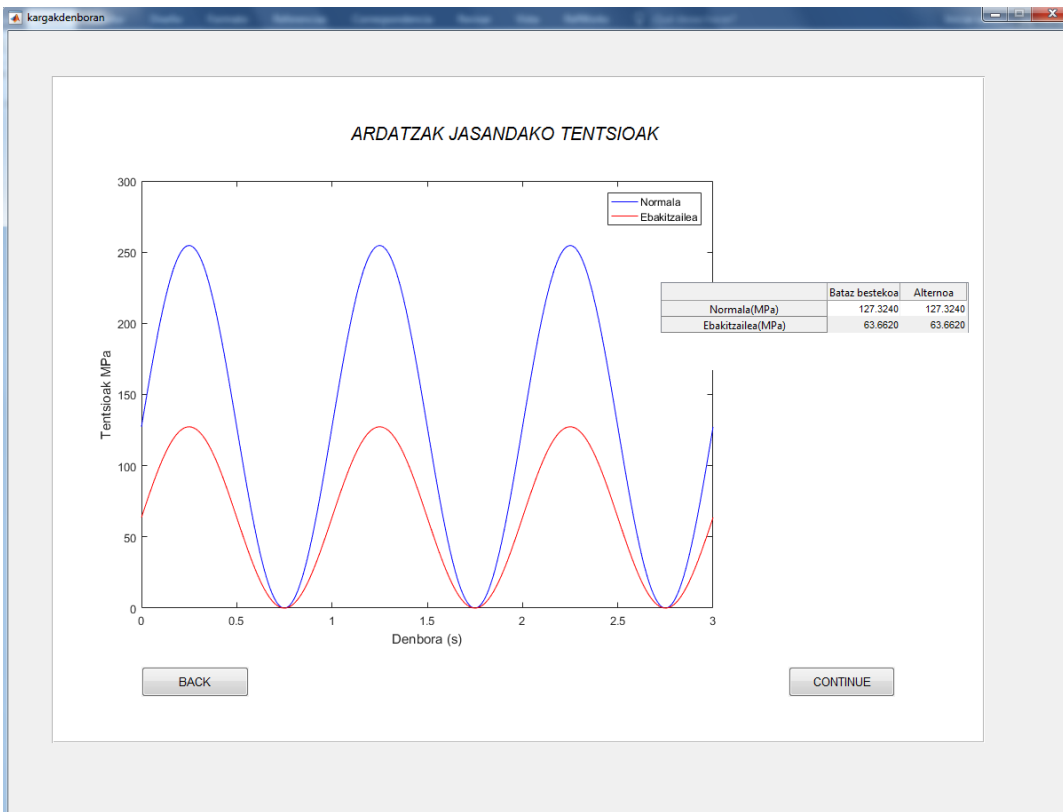
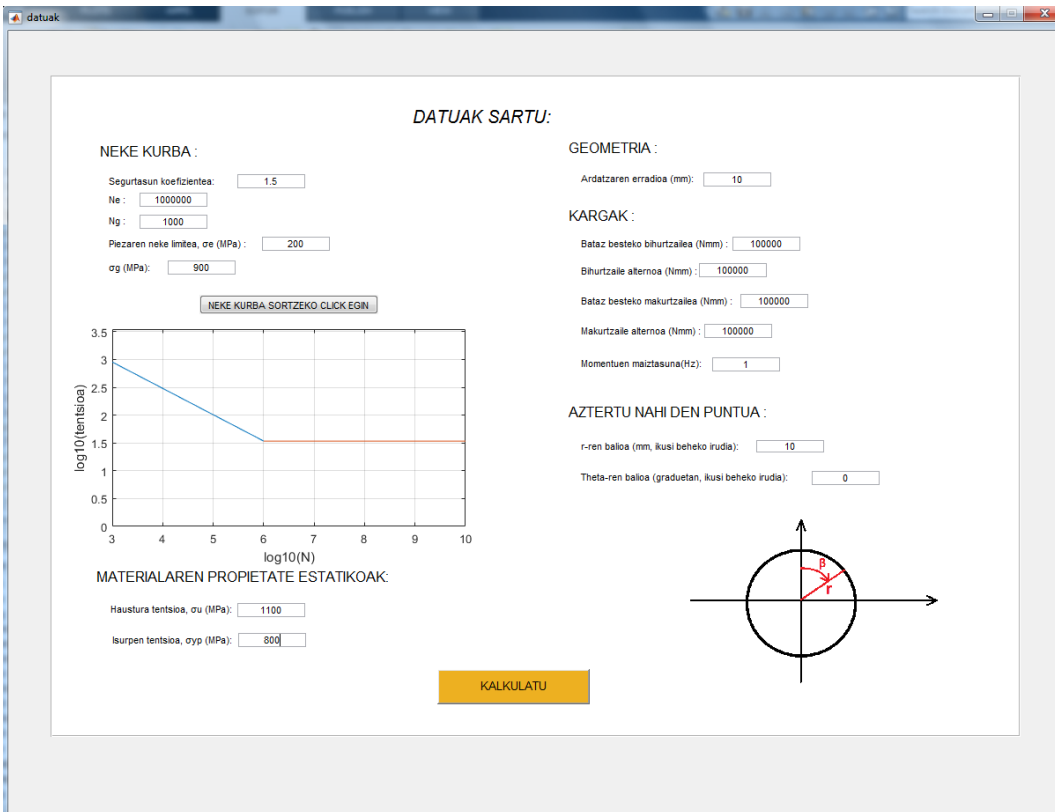
Momentu makurtzailearen zati alternoa, $M_r = 100000 \text{ Nmm}$

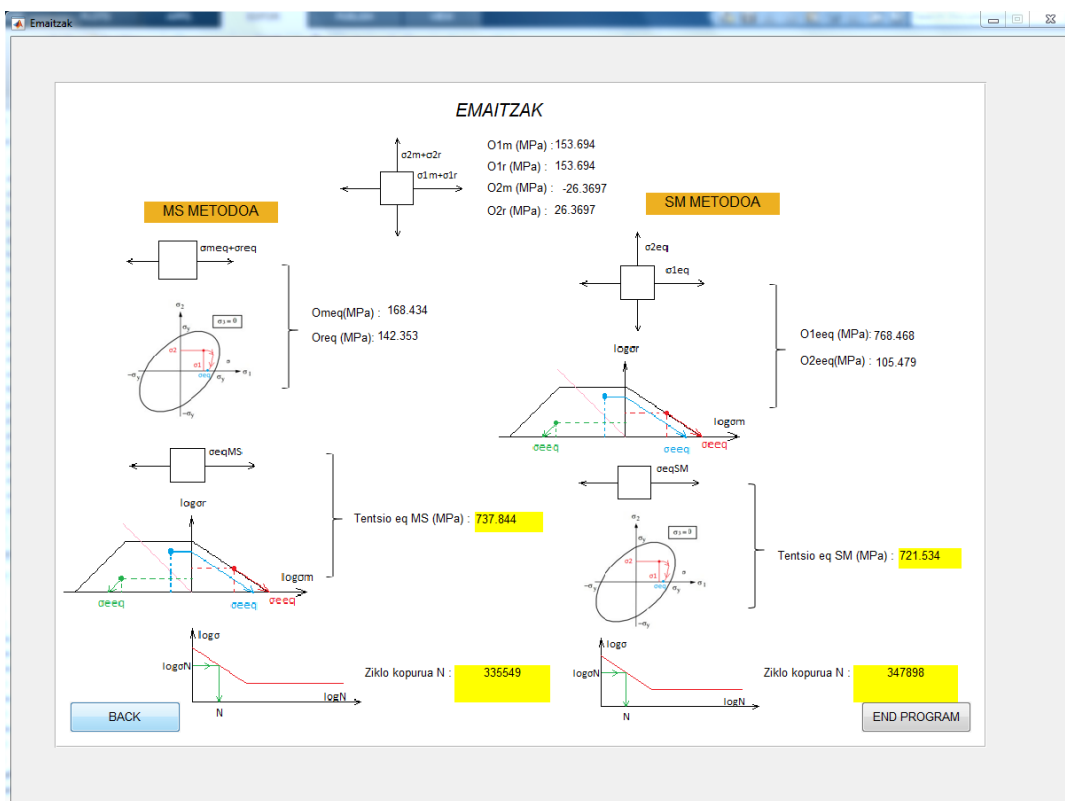
Aztertu nahi den puntuaren koordenatuak: $R = 10\text{mm}$, $\theta = 0$

R: Ardatzaren zentritik aztertu nahi den punturaino dagoen distantzia, norabide erradialean neurtuta.

Θ : Aztertu nahi den puntuaren eta norabide bertikaleko ardatzak osatzen duten angeluaren balio, graduetan neurtuta.





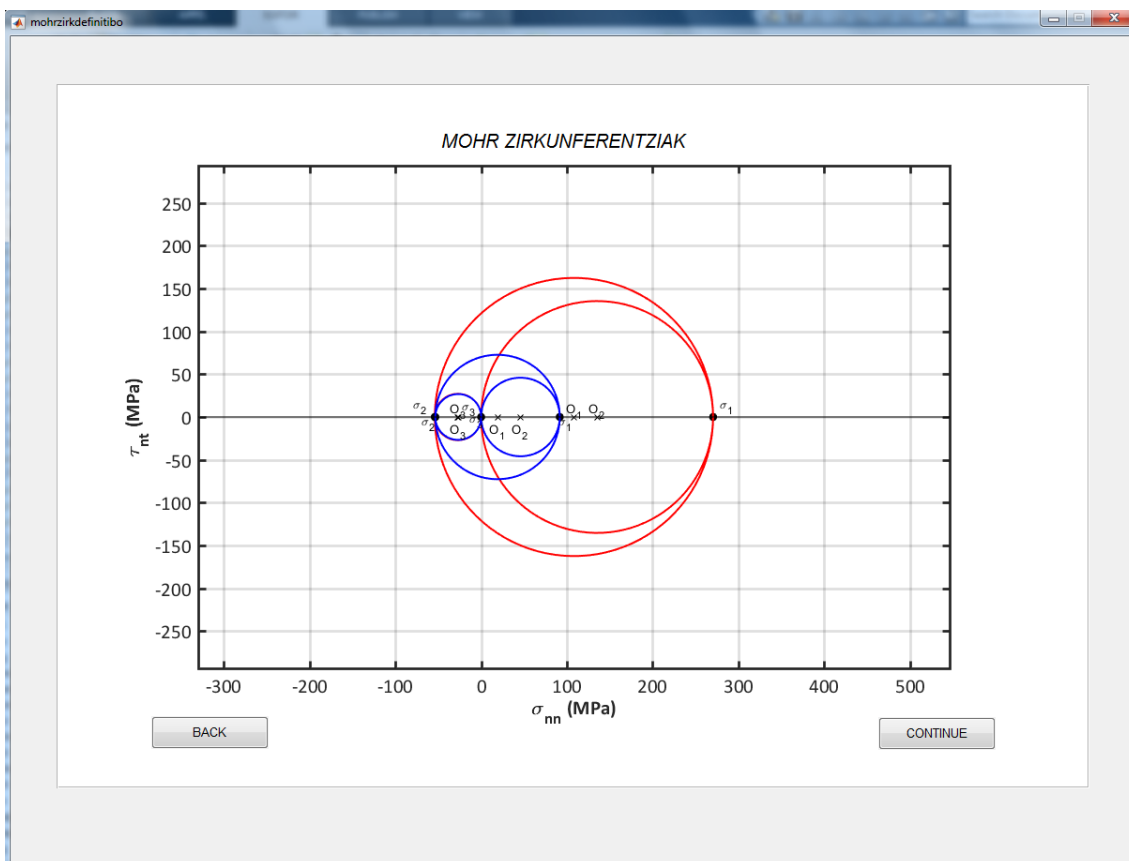


5 Taulan programatik lortutako emaitzak eta eskuzko kalkulua eginez (14.3. EANSKINAK: ESKUZKO ADIBIDEA) lortutako emaitzak konparatu egiten dira:

	ESKUZKO KALKULUA	PROGRAMA
Nms	335594,8	335549
σ_{eqMS}	737,83	737,844
Nsm	347928,15	347898
σ_{eqSM}	721,52	721,534

Taula 5: Emaitzen konparaketa

Mohr-en zirkunferentziak marraztean programak izatez tentsio maximo eta minimoen zirkunferentzia pareak erakusten ditu, tentsio nagusiak maximoak direneko unean zirkunferentziak gorriak direlarik eta tentsio minimoak direnean urdinez erakusten ditu. Kasu honetan karga pulsanteak aplikatu direnez tentsio nagusien balioak minimoak zero baliokoak izango dira eta ondorioz ez da zirkunferentziarik marraztuko. Beheko irudiak tentsio minimoak zero ez direneko kasu bat erakusten du:



9. METODOLOGIA

Atal honetan proiektu honetako programa sortzeko jarraitu den metodologia azalduko da. Aurretik azaldu den bezala, programa Matlab software-aren bidez garatuko da. Lehendabizi interfaze grafikoan programak edukiko duen diseinua eratuko da, hau da, programa erabiltzen duenak pantailan ikusiko duen diseinua eta ondoren, editorean interfaze grafikoan ezarritako irudiak, botoiak, static text-ak, edit textak... izango dituzten funtzioak programatuko dira baita neke analisia egiteko beharko diren kalkulu guztiak landuko dira. Programaren artxibo guztiak(.m eta .fig) eta irudiak karpeta berean egotea beharrezkoa izango da.

Interfaze grafiko editatzeko Matlab-en komandoen leihoan (ventana de comandos) *guide* hitza idatzi beharko da. Behin zabaldua dagoelarik programak eskaintzen dituen aukerekin programaren diseinua eraiki daiteke, hauen artean 5 erabiliko direlarik: “Panel”, programaren fondoak ezartzeko ; “Axes”, irudiak eta grafikoak erakusteko; “Static Text”, editatu ezin daitekeen testua idazteko eta emaitzak erakusteko; “Edit Text”, programaren erabiltzailek datuak sartzeko, editatu daitezkeelarik eta azkenik, “Push Botton”, botoiak. Botoi, kutxa, panel eta axes hauen diseinua aldatzeko *Property Inspector*-a zabaltu behar da, tamaina, kolorea, kokapena, letra mota etab. aldatu daitezkeelarik. Hala ere propietate garrantzitsuena izendapena izango da, “Tag” deritzona, izan ere, Matlab-en editoretik interfaze grafikoko botoi, kutxa, panel eta axes-ak erreferentziatzeko beharrezkoa izango baita. Ondoren, aukera bakoitzaren funtzioa azalduko da:

Panel:

Programako fondoak ezartzeko erabiliko da, zuri kolorekoak izanik guztiak.

Axes:

Aukera honekin grafikoak eta programakoak ez diren irudiak erakusteko erabiliko da., hala nola, tentsio normala eta ebakitzaileraren aldakuntza denboran adierazteko, mohr-en zirkuluak, Basquin-en kurba etab.

- Programakoak ez diren irudiak erakusteko:

Editorean idatzi beharreko kodea funtzio honen ondoren idatzi beharko da:

function *_Azpipogramarenizena*OpeningFcn(hObject, eventdata, handles, varargin)

Kodea:

1. **Axes(handles.*Erreferentzia*);**
 %Erreferentziatutako axes-ak eraldatzea ahalbidetuko du.
 2. **img=imread('Izena.extensioa');**
-

%Programaren kanpoaldeko irudia hartuko du.

3. **image(img);**

%Irudia erakusteko komandoa.

- Matlab-eko irudiak erakusteko hurrengo kodea erabilikoa:

Axes(handles.*Erreferentzia*);

%Erreferentziatutako axes-ak eraldatzea ahalbidetuko du.

Plot(*aldagaia*,*funtzioa*);

%Irudia erakusteko.

Edit text:

Kutxa hauekin programaren erabiltzaileak “datuak” azpipograman eskatzen diren datuak hartzeko erabiliko da. Kodigoa hurrengoa delarik:

1. ***AldagaiarenIzena*=get(handles.*EditKutxarenErreferentzia*, '*String*');**

%Erabiltzaileak edit kutxan sartutako datuak '*String*' mota aldagai moduan hartuko ditu.

2. ***Aldagaia*=str2double(*Aldagaia*);**

%Aldagaia String motatik Double motara bihurtuko dira programan zenbaki bezala erabiltzeko.

3. **assignin('base', '*Aldagaia*', *Aldagaia*)**

%Azken komando honekin aldagaiaren balioa Matlab-eko nukleloan gordeko da beste azpiprogrametan erabili ahal izateko.

Aipatu beharra dago Matlab-eko nukleloan gordetako aldagaiak beste azpiprogrametan erabiltzeko nukleotik atera beharko direla. Horretarako hurrengo komandoa erabili begar da:

***Aldagaia*=evalin('base', '*Aldagaia*');**

Push Botton:

Kutxa hauen bidez erabiltzaileak programan aurrera eta joateko aukera izango du ('Back' eta 'Continue' botoiak hurrenez hurren) baita grafikoak erakusteko funtzioa ere izango dute. Botoien funtzioak honela definitzen dira:

function *BotoiarenIzena_Callback*(hObject, eventdata, handles)

Botoia klikatzean funtzio honen ondoren programatzen den guztia exekutatu da. 'Back' eta 'Continue' botoien kasuan ondorengo kodea idatziko da, momentu honetan irekita dagoen azpiprograma ixteko eta nahi den azpiprograma irekitzeko:

close all;

clc;

IrekikoDenAzpiprograma

Static text:

Mota honetako kutxek programan bi funtzio nagusi eduki ditzakete, alde batetik, testua informatiboa idazteko, eta bestetik, neke analisiak emandako emaitzak erakusteko erabiliko dira.

Testu informatiboa daukaten kutxak interfasea diseinatzean editatuko dira, 'Property Inspector'-eko 'Edit Text' aukeran erakutsi nahi den testua idatziz.

Emaitzak erakusteko kutxak ordea editorean programatu beharko funtzio honen ostean:

function *Azpiprograma_OpeningFcn*(hObject, eventdata, handles, varargin)

eta komando hau erabiliz:

set(handles.*Erreferentzia*, 'String', *Aldagaia*)

'Static text'-en formatoa aldatzeko interfaze 'Property Inspector'-ak ematen dituen aukerak erabili behar dira, tamaina, kolorea eta letra mota aldatzeko, besteak beste.

10. AURREKONTUA

BARNE ORDUAK	ORDUAK	ORDUKO KOSTEA(€/h)	KOSTUA(€)
Ingeniari hasiberria	180	35	6300
Ingeniari aditua	35	55	1925
			8225

AMORTIZAZIOAK	ORDUAK	KOSTUA(€)	BIZITZA ERABILGARRIA*(ordu)	GUZTIZKOA(€)
Matlab programa	120	35	7300,00	0,575342466
Microsoft office programak	50	250	7300,00	1,712328767
Ordenagailua	170	700	7300,00	16,30136986
			*5 urtez egunean 4 ordu erabilia	18,5890411

KOSTUAK	GUZTIZKOA(€)
Idazteko materiala	15
Lana gordetzeko CD-ak	30
	45

AURREKONTUA	KOSTUA
Barne-orduak	8225
Amortizazioak	18,59
Kostuak	45
Azpitotala	8288,59
Ez-usteak(%5)	414,43
	8703,02

11. LANEN DESKRIBAPENA. GANTT.

L1: Informazioa eskuratu.

L1.1: Neke analisiaren inguruko ezagutzak eskuratu..

Lehenik eta behin, neke analisiaren inguruko informazio bilatzea beharrezkoa izango da, klasean ikasitakoaz gainera informazio iturri^[1] ezberdinetatik ezagutzak eskuratuz.

Iraupena: 8 egun

L1.2: Softwarea aukeratu.

Garatuko nahi programa sortzeko software egokienaren aukeraketa egitea, eskuratzeko erreztasuna, prezioa eta batez ere eskainiko dituen aukerak kontuan hartuta.

Iraupena: 1 egun

L1.3: Softwareak eskaintzen dituen aukera eta komandoen azterketa.

Behin softwarea aukeratuta, eskaini ditzakeen aukera guztiak aztertuko dira, baita izan ditzakeen erabilera ezberdinak eta komandoen ikasketa egitea.

Iraupena: 10 egun

L2: Alternatiben analisia .

Aukeratu den programak dituen alternatiben analisia eta konparaketa egitea.

Iraupena: 2 egun

L3: Programaren garapena.

Ardatzen neke analisia burutzen duen programa sortzea.

L3.1: Portada.

Programak edukiko duen portada egitea, hau da, irekitzean agertuko den aurreneko azpiprograma egitea.

Iraupena: 1 egun

L3.2: Datuak eskuratzeko azpiprograma.

Neke analisia burutzeko ezinbestekoak izango diren datuak jasoko dituen azpiprograma garatuko da. Datuak eskuratzear gain Basquin-en kurba irudikatuko da.

Iraupena: 3 egun

L3.3: Tentsio normala eta ebakitzaila denboran erakusten dituen azpiprograma.

Ardatzak denboran zehar jasango dituen tentsio normala eta ebakitzaila irudikatuko dira, bakoitzaren balio maximoa eta minimoa taula baten bidez adieraziz.

Iraupena: 5 egun

L3.4: Mohr-en zirkunferentziak.

Tentsio nagusien kalkulua egingo da eta ondoren, balio maximoen eta minimoen Mohr-en zirkunferentziak marraztuko dira.

Iraupena: 10 egun

L3.5: Tentsio nagusiak denbora erakusteko azpiprograma.

Tentsio nagusiek denboran zehar edukiko duten bilakaera grafiko baten bidez erakutsiko duen azpiprograma garatuko da, balio maximoak eta minimoak taula baten bidez erakutsiz.

Iraupena: 3 egun

L4.6: Erantzunen azpiprograma.

Azpiprograma honetan SM eta MS programatuko dira eta urratsez urrats metodo bakoitzak lortuko dituen emaitzak erakutsiko dira, pauso bakoitzean irudi adierazleak erabiliz.

Iraupena: 15 egun

L5: Adibide ezberdinen eskuzko eta proiektu honetan sortutako programa bidezko kalkuluak egitea.

Eskuzko nahiz programa bidez kalkuluak egitea eta emaitzak konparatzea.

Iraupena: egun 2

L6: Emaitzak aztertzea.

Lortutako emaitzen konparaketa eta azterketa egitea eta ondorioak ateratzea.

Iraupena: egun 1

L7: Proiektua idaztea.

Proiektuak eskatzen dituen atal guztien erredakzioa, beharrezkoa den informazio guztia azalduz.

Iraupena: 20 egun

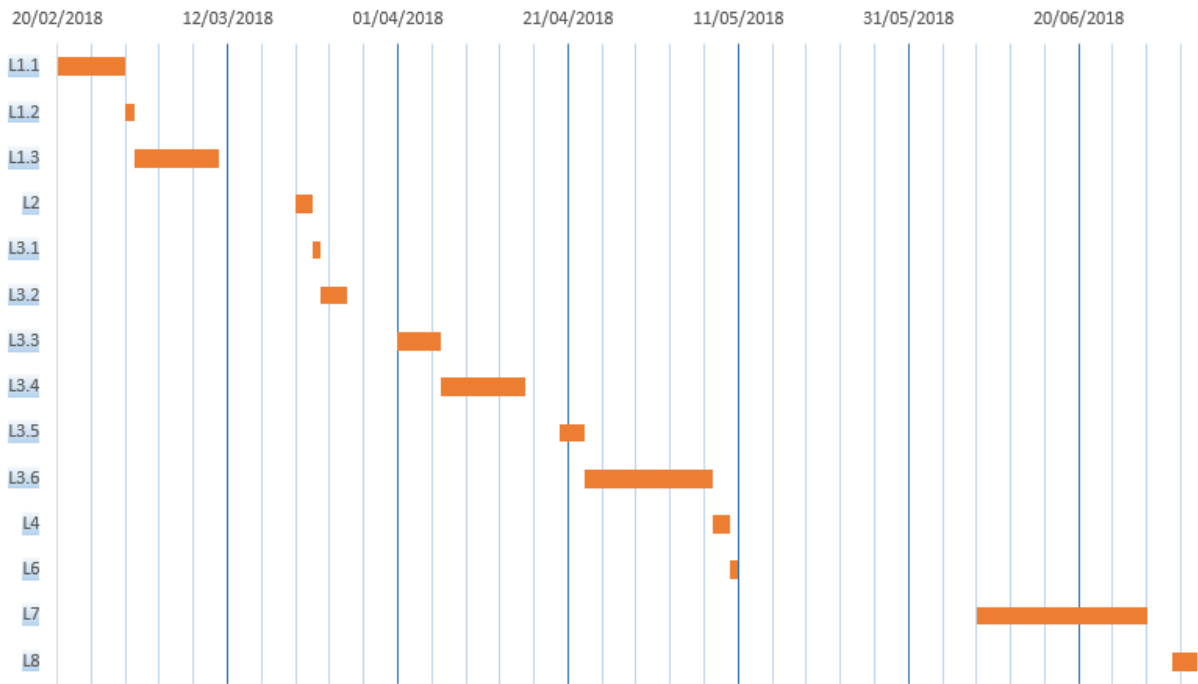
L8: Aurkezpena prestatzea.

Proiektuaren defentsa prestatzea, aurkezpenerako materiala prestatuz eta hitzaldia entsegatuz.

Iraupena: 3 egun



GANTT DIAGRAMA



12. ONDORIOAK

Proiektu honetan sortutako neke programa erabilia eta eskuzko kalkuluetatik lortutako emaitzekin konparatuta, neke analisi multiaxialerako eraginkorra eta egokia dela ondorioztatu daiteke.

Aurreko ataletan ebatzitako eskuzko kalkulua neketsua eta luzea izan daitekeela ikusi daiteke, programa honen garapena justifikatuz. Programa sortzeko ere hainbat egun behar izan badira ere, behin sortuta neke analisi multiaxiala segundotan burutuko du inolako akatsik egin gabe. Beraz esan daiteke programa honen bidez epe luzera denbora aurreztuko dela, kalkuluak zehazki burutuko direla, neke analisia pauso pauso argi aurkeztuko dela eta emaitzak era argian eta ordenatuan aurkeztuko dira. Gainera, ardatzak jasandako tentsioak eta tentsio nagusiak denboran nola aldatuko diren erakusten duten grafikoak aurkeztuko dira baita tentsio nagusien balio maximoen eta minimoen Mohr-en zirkunferentziak ere pantailaratuko dira segundo eskas batzuen buruan. Nahiz eta programaren erabiltzaileak neke analisian aditua ez izan, ez du inolako arazorik programa erabiltzeko, sinplea eta argia baita.

Programak nahiko abantaila baditu ere, pare bat desabantaila ere baditu. Alde batetik, programaren formatua mugatua da, datuak era espezifikoan eskatzen dira eta programaren leihatilak beti ordena berean aurkezten dira. Beste alde batetik, programa bera sortzeko neke analisiaren eta Matlab softwarearen inguruko ezagutza handia eduki behar da eta egun nahiko behar izan dira hura osatzeko. Gainera, neke analisiak burutzeko metodo klasikoak erabiliko dira soilik eta tentsio egoera hidrostatiakoetan huts egiten dutela kontuan hartu beharra dago, eta ardatzen neke analisia soilik burutu daiteke.

Aurreko ataletan aipatu izan den bezala, programa hau irakaskuntzara nahiz enpresa mundura bideratuta egon daiteke, irakaskuntza munduan ardatzen inguruko neke analisiak burutzeko eta enpresa munduan ordea, ardatzen aurrezimentsionaketa egiteko erraminta egokia izanik.

Azkenik, aipatu beharra dago programa aurrera eramateko erabili den softwarea Matlab izan da, neke analisi espezifikoak egiteko elementu finituen programak baino askoz merkeagoa eta erabiltzeko errazagoa da baina mugatuagoa ere bai.

Atal honetan aipatutako guztia aztertuta ondorioztatu daiteke proiektu honetan sortutako metodo klasikoetan oinarritutako neke analisi multiaxialerako programa efizientea, egokia, zehatza eta merkea dela. Hala ere, bere mugak ere baditu, ardatzen neke analisia soilik burutzen baitu eta tentsio egoera hidrostatiakoan hust egiten baitu.

13. BIBLIOGRAFIA

Rafael Avilés, (2015), *Métodos de cálculo de fatiga para ingeniería*, España.

Barragán Guerrero, D. O., (2008), *Manual de Interfaz Gráfica de Usuario en Matlab Parte I*, Ecuador.

Carrera Amuriza, A. R. eta Martínez Nebreda, M., Departamento de Matemáticas Aplicadas ETSIB (2004), *Introducción a Matlab Y la creación de interfaces gráficas*, Bilbao, España.

14. ERANSKINAK

14.1. ERANSKINAK: KODEA.

14.1.1. PORTADA

```
function varargout = portada(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @portada_OpeningFcn, ...
                  'gui_OutputFcn',  @portada_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before portada is made visible.
function portada_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to portada (see VARARGIN)

% Choose default command line output for portada
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = portada_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

[img,mapa]=imread('portada.PNG','PNG');
```

```
imshow(img);  
colormap(mapa)
```

```
% --- Executes on button press in startprogram.  
function startprogram_Callback(hObject, eventdata, handles)  
close all;  
clc;  
datuak
```

14.1.2. DATUAK AZPIPROGRAMA

```

function varargout = datuak(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @datuak_OpeningFcn, ...
                  'gui_OutputFcn',  @datuak_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before datuak is made visible.
function datuak_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for datuak
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = datuak_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

function isurpententsioa_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function isurpententsioa_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function hausturatentsioa_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.

```

```
function hausturatentsioa_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function erradioa_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function erradioa_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function cs_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function cs_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function mbb_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function mbb_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function ma_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function ma_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function bbb_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function bbb_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end
```

```
function ba_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function ba_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function maiztasuna_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function maiztasuna_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function rbalioa_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function rbalioa_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function nekelimitea_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function nekelimitea_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function sigmag_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function sigmag_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Ne_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function Ne_CreateFcn(hObject, eventdata, handles)
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function Ng_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function Ng_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end
```

```
function theta_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function theta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end
```

```
%Aztertuko den puntuaren irudia erakusteko:
img=imread('theta.png');
imshow(img);
```

```
% --- Executes on button press in nekekurbabotoi.
function nekekurbabotoi_Callback(hObject, eventdata, handles)
```

```
%neke kurba sortu eta irudikatu:
```

```
Oe=get(handles.nekelimitea, 'String');
Og=get(handles.sigmag, 'String');
Ne=get(handles.Ne, 'String');
Ng=get(handles.Ng, 'String');
cs=get(handles.cs, 'String');
```

```
Oe=str2double(Oe);
Og=str2double(Og);
Ne=str2double(Ne);
Ng=str2double(Ng);
cs=str2double(cs);
```

```
Ne=log10(Ne);
Ng=log10(Ng);
Oe=log10(Oe);
Og=log10(Og);
```

```
Oeprima=(Oe/(cs));
```

```
x1=Ng:0.05:Ne;
x2=Ne;
x3=10;
f1=@(t) (Oeprima-Og)/(Ne-Ng)*(t-Ng)+Og;
f2= Oeprima;
axes(handles.nekekurba);
plot(x1, f1(x1))
```

```

axis([Ng 10 0.9*Oe 1.2*Og])
grid on
hold on
plot([x2,x3],[f2,f2])
axis([Ng 10 0 1.2*Og])
xlabel('log10(N)')
ylabel('log10(tentsioa)')
hold off

% --- Executes on button press in kalkulubotoia.
function kalkulubotoia_Callback(hObject, eventdata, handles)

%Aurreko ekuazioetan lortzen diren aldagaiak matlaben basean gordeko
dira
%assignin komandoa erabiliz:
%Aurretik sartutako datuak hartzeko "get" komandoa erabiliko da,
string
%motan gordeko direlarik:
Oyp=get(handles.isurpententsioa,'String');
Ou=get(handles.hausturatentsioa,'String');
r=get(handles.erradioa,'String');
CS=get(handles.cs,'String');
Tb=get(handles.bbb,'String');
Ta=get(handles.ba,'String');
Mm=get(handles.mbb,'String');
Ma=get(handles.ma,'String');
f=get(handles.maiztasuna,'String');
R=get(handles.rbalioa,'String');
theta=get(handles.theta,'String');
Oe=get(handles.nekelimitea,'String');
Og=get(handles.sigmag,'String');
Ne=get(handles.Ne,'String');
Ng=get(handles.Ng,'String');

%String motako datuak double motako datuetara pasatuko dira
"str2double" komandoaren bitartez:
Oyp=str2double(Oyp);
Ou=str2double(Ou);
r=str2double(r);
CS=str2double(CS);
Tb=str2double(Tb);
Ta=str2double(Ta);
Mb=str2double(Mm);
Ma=str2double(Ma);
f=str2double(f);
R=str2double(R);
theta=str2double(theta);
Oe=str2double(Oe);
Og=str2double(Og);
Ne=str2double(Ne);
Ng=str2double(Ng);
w=2*pi*f;

%aztertuko den puntuaren altueraren kalkulua:
theta=theta*2*pi/360;
y=R*cos(theta);

```

```
%sektzio zirkular trinko baten inertzi momentua
I=0.25*pi*(r^4);
%sektzio zirkularraren inertzi momentu polarra
Ip=2*I;
%tentsio ebakitzaille medioa
Taum=(Tb*R)/Ip;
%tentsio ebakitzaille alternoa
Taur=(Ta*R)/Ip;
%tentsio normal medioa
Om=(Mb*y)/I;
%tentsio normal alternoa
Or=(Ma*y)/I;

%Aldagaiak Matlab-eko nukleoan gordetzeko:
assignin('base','Oyp',Oyp);
assignin('base','Ou',Ou);
assignin('base','Ne',Ne);
assignin('base','Ng',Ng);
assignin('base','r',r);
assignin('base','CS',CS);
assignin('base','Taum',Taum);
assignin('base','Taur',Taur);
assignin('base','Om',Om);
assignin('base','Or',Or);
assignin('base','w',w);
assignin('base','Og',Og);
assignin('base','Oe',Oe);
assignin('base','y',y);
clear all;
close all;
clc;
kargakdenboran
```

14.1.3. TENTSIOAK

```

function varargout = kargakdenboran(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @kargakdenboran_OpeningFcn, ...
                  'gui_OutputFcn',  @kargakdenboran_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before kargakdenboran is made visible.
function kargakdenboran_OpeningFcn(hObject, eventdata, handles,
varargin)

% Choose default command line output for kargakdenboran
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = kargakdenboran_OutputFcn(hObject, eventdata,
handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

%Azpiprograma honetan beharrezkoak izango diren aldagaiak Matlab-eko
nukleotik
%hartzeko:
Om=evalin('base','Om');
Or=evalin('base','Or');
Taum=evalin('base','Taum');
Taur=evalin('base','Taur');
w=evalin('base','w');

%Tentsio normala eta ebakitzaila irudikatzeko:
T=2*pi/w;
t=linspace(0,3*T,1000);
tensionormala=@(t) Or*sin(w*t)+Om;

```

```
tentsioebakitzalea=@(t) Taur*sin(w*t)+Taum;

plot(t,tentsionormala(t),'b');
hold on
plot(t,tentsioebakitzalea(t),'r');
hold off

xlabel('Denbora (s)')
ylabel('Tentsioak MPa')
legend('Normala','Ebakitzalea')

%Taula betetzeko:
batazbeste={Om Taum};
alterno={Or Taur};
taula=[batazbeste' alterno'];
set(handles.uitable1,'data',taula);

% --- Executes on button press in atzerabotoia.
function atzerabotoia_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;
datuak

% --- Executes on button press in continuebotoia.
function continuebotoia_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;
mohrzirkdefinitibo
```

14.1.4. MOHR-en ZIRKUNFERENTZIAK

```

function varargout = mohrzirkdefinitibo(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @mohrzirkdefinitibo_OpeningFcn,
                  ...
                  'gui_OutputFcn',   @mohrzirkdefinitibo_OutputFcn,
                  ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before mohrzirkdefinitibo is made visible.
function mohrzirkdefinitibo_OpeningFcn(hObject, eventdata, handles,
varargin)

% Choose default command line output for mohrzirkdefinitibo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = mohrzirkdefinitibo_OutputFcn(hObject, eventdata,
handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

%Beharrezkoak diren aldagaiak programako nukleotik hartzeko:
Om=evalin('base','Om');
Or=evalin('base','Or');
Taum=evalin('base','Taum');
Taur=evalin('base','Taur');

%Tentsio normal eta ebakitzaille maximoak kalkulatzeko:
Omax=Om+Or;
Omin=Om-Or;
Taumax=Taum+Taur;
Taumin=Taum-Taur;

%Tentsio nagusi maximo eta minimoen kalkulua:
Tmax=[Omax Taumax 0; Taumax 0 0; 0 0 0];

```

```

Tmin=[Omin Taumin 0; Taumin 0 0; 0 0 0];
tmax=eig(Tmax);
tmin=eig(Tmin);
S1max=tmax(1);
S2max=tmax(2);
S3max=tmax(3);
S1min=tmin(1);
S2min=tmin(2);
S3min=tmin(3);
Smax=[S1max,S2max,S3max];
Smin=[S1min,S2min,S3min];

%S1 beti tentsio handiena izan dadin, konbenioz:
if (max(Smax)>0)
  for i=1:1:3
    if (Smax(i)~=0 && Smax(i)~=max(Smax))
      S2max=Smax(i);
    end
  end
  S1max=max(Smax);
  S3max=0;
else
  for i=1:1:3
    if (Smax(i)~=0 && Smax(i)~=min(Smax))
      S1max=Smax(i);
    end
  end
  S2max=min(Smax);
  S3max=0 ;
end

if (max(Smin)>0)
  for i=1:1:3
    if (Smin(i)~=0 && Smin(i)~=max(Smin))
      S2min=Smin(i);
    end
  end
  S1min=max(Smin);
  S3min=0;
else
  for i=1:1:3
    if (Smin(i)~=0 && Smin(i)~=min(Smin))
      S1min=Smin(i);
    end
  end
  S2min=min(Smin);
  S3min=0;
end

%Aldagaiak Matlab-eko nukleoan gordetzeko:
assignin('base','S1max',S1max);
assignin('base','S2max',S2max);
assignin('base','S3max',S3max);
assignin('base','S1min',S1min);
assignin('base','S2min',S2min);
assignin('base','S3min',S3min);

%Mohr-en zirkunferentzien zentroen posizio maximoa eta minimoa

```



```

kalkulatzeko:
z1max=(S1max+S2max)/2;
z2max=(S1max+S3max)/2;
z3max=(S2max+S3max)/2;
zentroakmax=[z1max,z2max,z3max];

z1min=(S1min+S2min)/2;
z2min=(S1min+S3min)/2;
z3min=(S2min+S3min)/2;
zentroakmin=[z1min,z2min,z3min];

%Erradioen balio maximoaren eta minimoaren kalkulua:
r1max=(S1max-S2max)/2;
r2max=(S1max-S3max)/2;
r3max=(S2max-S3max)/2;
rmax=[r1max,r2max,r3max];

r1min=(S1min-S2min)/2;
r2min=(S1min-S3min)/2;
r3min=(S2min-S3min)/2;
rmin=[r1min,r2min,r3min];

%Tentsio nagusi maximoen zirkunferentziak irudikatzeko:

%Lehenengo zirkunferentzia:
thetal=linspace(0,2*pi,100);
rho1=ones(1,100)*r1max;
[x1,y1]=pol2cart(thetal,rho1);
x1max=x1+z1max;
y1=y1;
plot(x1max,y1,'r-','LineWidth',1.5);
axis square
hold on
plot(z1max,0,'kx','MarkerFaceColor','k');
hold on
plot(z1max+r1max,0,'ko','MarkerFaceColor','k');
hold on
plot(z1max-r1max,0,'ko','MarkerFaceColor','k');
hold on
axis equal

text(z1max,0.3*min(abs(rmax)),'O_{1}',...
     'HorizontalAlignment','center','FontSize',10)
text(z1max+1.1*r1max,r1max/12,'\sigma_{1}',...
     'HorizontalAlignment','center','FontSize',10)
text(z1max-1.1*r1max,r1max/12,'\sigma_{2}',...
     'HorizontalAlignment','center','FontSize',10);
set(gca,'LineWidth',2,'FontSize',16,'FontWeight','normal','FontName','
Calibri');

%Bigarren zirkunferentzia:
theta2=linspace(0,2*pi,100);
rho2=ones(1,100)*r2max;
[x2,y2]=pol2cart(theta2,rho2);
x2max=x2+z2max;
y2=y2;
plot(x2max,y2,'r-','LineWidth',1.5);

```

```

axis square
hold on
plot(z2max,0,'kx','MarkerFaceColor','k');
hold on
plot(z2max-r2max, 0, 'ko','MarkerFaceColor','k');
hold on
axis equal

text(z2max,0.3*min(abs(rmax)), 'O_{2}',...
     'HorizontalAlignment','center','FontSize',10)
text(z2max-1.1*r2max,r2max/12, '\sigma_{3}',...
     'HorizontalAlignment','center','FontSize',10);
set(gca, 'LineWidth',2, 'FontSize',16, 'FontWeight','normal', 'FontName',
'Calibri');

%Hirugarren zirkunferentzia:
theta3=linspace(0,2*pi,100);
rho3=ones(1,100)*r3max;
[x3,y3] = pol2cart(theta3,rho3);
x3max=x3+z3max;
y3=y3;
plot(x3max,y3,'r-','LineWidth',1.5);
axis square
hold on
plot(z3max,0,'kx','MarkerFaceColor','k');
hold on
axis equal

text(z3max,0.3*min(abs(rmax)), 'O_{3}',...
     'HorizontalAlignment','center','FontSize',10);
set(gca, 'LineWidth',2, 'FontSize',16, 'FontWeight','normal', 'FontName',
'Calibri');

plot([min(Smax)-1.7*abs(max(rmax)), max(Smax)+1.7*abs(max(rmax))],[0,
0], 'k-');
hold on
set(get(gca, 'XLabel'), 'String', '\sigma_{nn}
(MPa)', 'FontSize',16, 'FontWeight', 'bold', 'FontName', 'Calibri');
set(get(gca, 'YLabel'), 'String', '\tau_{nt}
(MPa)', 'FontSize',16, 'FontWeight', 'bold', 'FontName', 'Calibri');
hold on

%Tentsio nagusi maximoen zirkunferentziak irudikatzeko:

%Lehenengo zirkunferentzia:
theta1=linspace(0,2*pi,100);
rho1=ones(1,100)*r1min;
[x1,y1] = pol2cart(theta1,rho1);
x1min=x1+z1min;
y1=y1;
plot(x1min,y1,'b-','LineWidth',1.5);
axis square
hold on
grid on
plot(z1min,0,'kx','MarkerFaceColor','k');

```

```
hold on
plot(z1min+r1min, 0, 'ko', 'MarkerFaceColor', 'k');
hold on
plot(z1min-r1min, 0, 'ko', 'MarkerFaceColor', 'k');
hold on
axis equal

text(z1min, -0.6*min(abs(rmin)), 'O_{1}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10)
text(z1min+1.1*r1min, -r1min/12, '\sigma_{1}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10)
text(z1min-1.1*r1min, -r1min/12, '\sigma_{2}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10);
set(gca, 'LineWidth', 2, 'FontSize', 16, 'FontWeight', 'normal', 'FontName', '
Calibri');

%Bigarren zirkunferentzia:
theta2=linspace(0, 2*pi, 100);
rho2=ones(1, 100)*r2min;
[x2, y2] = pol2cart(theta2, rho2);
x2min=x2+z2min;
y2=y2;
plot(x2min, y2, 'b-', 'LineWidth', 1.5);
axis square
hold on
plot(z2min, 0, 'kx', 'MarkerFaceColor', 'k');
hold on
plot(z2min-r2min, 0, 'ko', 'MarkerFaceColor', 'k');
hold on
axis equal

text(z2min, -0.6*min(abs(rmin)), 'O_{2}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10)
text(z2min-1.1*r2min, -r2min/12, '\sigma_{3}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10);
set(gca, 'LineWidth', 2, 'FontSize', 16, 'FontWeight', 'normal', 'FontName', '
Calibri');

%Hirugarren zirkunferentzia:
theta3=linspace(0, 2*pi, 100);
rho3=ones(1, 100)*r3min;
[x3, y3] = pol2cart(theta3, rho3);
x3min=x3+z3min;
y3=y3;
plot(x3min, y3, 'b-', 'LineWidth', 1.5);
axis square
hold on
plot(z3min, 0, 'kx', 'MarkerFaceColor', 'k');
hold on
axis equal

text(z3min, -0.6*min(abs(rmin)), 'O_{3}', ...
     'HorizontalAlignment', 'center', 'FontSize', 10);
set(gca, 'LineWidth', 2, 'FontSize', 16, 'FontWeight', 'normal', 'FontName', '
Calibri');
```

```
plot([min(Smin)-1.7*abs(max(rmin)), max(Smin)+1.7*abs(max(rmin))],[0,0], 'k-');
hold on
set(get(gca,'XLabel'),'String','\sigma_{nn} (MPa)', 'FontSize',16,'FontWeight','bold','FontName','Calibri');
set(get(gca,'YLabel'),'String','\tau_{nt} (MPa)', 'FontSize',16,'FontWeight','bold','FontName','Calibri');

% --- Executes on button press in continuebotoia.
function continuebotoia_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;
tentsionagusiakdenboran

% --- Executes on button press in backbotoia.
function backbotoia_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;
kargakdenboran
```

14.1.5. TENTSIO NAGUSIAK

```

function varargout = tentsionagusiakdenboran(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @tentsionagusiakdenboran_OpeningFcn, ...
                  'gui_OutputFcn',  @tentsionagusiakdenboran_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before tentsionagusiakdenboran is made visible.
function tentsionagusiakdenboran_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for tentsionagusiakdenboran
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = tentsionagusiakdenboran_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

%Matlabeko basetik beharrezko datuak hartu:
S1max=evalin('base','S1max');
S1min=evalin('base','S1min');
S2max=evalin('base','S2max');
S2min=evalin('base','S2min');
S3max=evalin('base','S3max');
S3min=evalin('base','S3min');
w=evalin('base','w');

%Tentsio nagusien bataz bestekoa eta parte alternoa:
O1r=0.5*(S1max-S1min);
if (O1r>=0)
    O1r=O1r;
else

```

```

    O1r=-O1r;
end
O1m=0.5*(S1max+S1min);

O2r=0.5*(S2max-S2min);
if (O2r>=0)
    O2r=O2r;
else
    O2r=-O2r;
end
O2m=0.5*(S2max+S2min);

O3r=0.5*(S3max-S3min);
O3m=0.5*(S3max+S3min);

assignin('base','O1r',O1r);
assignin('base','O1m',O1m);
assignin('base','O2r',O2r);
assignin('base','O2m',O2m);
assignin('base','O3r',O3r);
assignin('base','O3m',O3m);

batazbeste={O1m O2m O3m};
alterno={O1r O2r O3r};
taula=[batazbeste' alterno'];
set(handles.uitable1,'data',taula);

%tentsio nagusien grafikatuko dira denboran zehar nola aldatzen diren
%ikusteko:
T=2*pi/w;
t=linspace(0,3*T,1000);
TentsioNagusi1=@(t) O1r*sin(w*t)+O1m;
TentsioNagusi2=@(t) O2r*sin(w*t)+O2m;
TentsioNagusi3=@(t) O3r*sin(w*t)+O3m;
plot(t,TentsioNagusi1(t),'r');
hold on
plot(t,TentsioNagusi2(t),'b');
hold on
plot(t,TentsioNagusi3(t),'k');
hold off
xlabel('denbora segundutan')
ylabel('Tentsio Nagusien balioak MPa')
legend('Lehenengoa','Bigarrena','Hirugarrena')

batazbeste={O1m O2m O3m};
alterno={O1r O2r O3r};
taula=[batazbeste' alterno'];
set(handles.uitable1,'data',taula);

% --- Executes on button press in back.
function back_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;

```

mohrzirkdefinitibo

```
% --- Executes on button press in continueprogram.  
function continueprogram_Callback(hObject, eventdata, handles)  
clear all;  
close all;  
clc;  
Emaitzak
```

14.1.6. EMAITZAK

```
function varargout = Emaitzak(varargin)
```

```

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Emaitzak_OpeningFcn, ...
                  'gui_OutputFcn',  @Emaitzak_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Emaitzak is made visible.
function Emaitzak_OpeningFcn(hObject, eventdata, handles, varargin)

% Choose default command line output for Emaitzak
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = Emaitzak_OutputFcn(hObject, eventdata, handles)

% Get default command line output from handles structure
varargout{1} = handles.output;

%Programa nagusitik datuak hartzeko:
O1m=evalin('base','O1m');
O2m=evalin('base','O2m');
O3m=evalin('base','O3m');
O1r=evalin('base','O1r');
O2r=evalin('base','O2r');
O3r=evalin('base','O3r');
Oyp=evalin('base','Oyp');
Oe=evalin('base','Oe');
cs=evalin('base','CS');
Og=evalin('base','Og');
Ne=evalin('base','Ne');
Ng=evalin('base','Ng');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%-----MS metodoa-----%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1. Von mises1-en irizpidea erabiliz Oeqm eta Oeqr kalkulatu:
Oeqm=sqrt(((O1m-O2m)^2+(O1m-O3m)^2+(O2m-O3m)^2)*0.5);
Oeqr=sqrt(((O1r-O2r)^2+(O1r-O3r)^2+(O2r-O3r)^2)*0.5);

```



```

%tentsio baliokideen ikurra zehazteko:
if (O1m+O2m+O3m>=0)
    Oeqm=Oeqm;
else
    Oeqm=-Oeqm;
end

%2. soderberg1-en irizpidearekin OeqMS kalkulatu:
if (Oeqm>=0)
    OeqMS=Oeqm+Oeqr*(Oyp/Oe);
elseif (Oeqm<0 && Oeqr>=abs(Oeqm))
    OeqMS=Oeqm*(Oyp/Oe);
elseif (Oeqm<0 && Oeqr<abs(Oeqm))
    OeqMS=Oeqm-Oeqr;
end

%Hutsegitea gertatuko den ala ez kalkulatu, huts egiterik ez
gertatzekotan piezak bizitza
%infinetua edukiko duela adieraziko da eta htsegitea gertatzen bada
berri, hutsegitea
%zenbatgarren zikloan emango den kalkulatu da. Tentsio baliokidea Og
baino handiagoa bada,
%LCF(ziklo baxuetako nekea) zonan egongo garela adieraziko da eta
estimatutako kalkuluak
%ezin izango dira kontuan hartu SM eta MS metodoak HCF zonan soilik
erabili daitezkeelako:

if (OeqMS<=Oyp/cs)
    Nms='Bizitza infinitua';
    axes(handles.snkurba1);
    img=imread('snkurbainf.PNG');
    imshow(img);
elseif (OeqMS>Oyp/cs && OeqMS<=Og)
    if (Oeqm>=0)
        ON=Oyp*Oeqr/(Oyp-Oeqm);
    elseif (Oeqm<0 && Oeqr>=abs(Oeqm))
        ON=Oeqm;
    else
        ON=Oeqm/2;
    end

    Oeprima=Oe/(cs);
    logaritmoN=log10(Ne)-(log10(ON)-log10(Oeprima))*(log10(Ne)-
log10(Ng))/(log10(Og)-log10(Oeprima));
    Nms=10^logaritmoN;
    if (Nms>1000000)
        Nms=999999;
    else
        Nms=Nms;
    end
    axes(handles.snkurba1);
    img=imread('snkurba.PNG');
    imshow(img);

else
    axes(handles.snkurba1);
    img=imread('lcf.PNG');
  
```

```

    imshow(img);
    Nms='Ziklo baxuak!(ON>Og)';
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%-----SM metodoa:-----%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%1. Soderberg-en irizpidea erabiliko da tentsio nagusi baliokide
estatikoak kalkulatzeko:
if (O1m>=0)
    O1eeq=O1m+O1r*(Oyp/Oe);
elseif (O1m<0 && O1r>=abs(O1m))
    O1eeq=O1r*(Oyp/Oe);
elseif (O1m<0 && O1r<abs(O1m))
    O1eeq=O1m-O1r;
end

if (O2m>=0)
    O2eeq=O2m+O2r*(Oyp/Oe);
elseif (O2m<0 && O2r>=abs(O2m))
    O2eeq=O2r*(Oyp/Oe);
elseif (O2m<0 && O2r<abs(O2m))
    O2eeq=O2m-O2r;
end

if (O3m>=0)
    O3eeq=O3m+O3r*(Oyp/Oe);
elseif (O3m<0 && O3r>=abs(O3m))
    O3eeq=O3r*(Oyp/Oe);
elseif (O3m<0 && O3r<abs(O3m))
    O3eeq=O3m-O3r;
end

%2. Von mises1-en irizpidea erabiliz tentsio uniaxial estatiko
baliokidea lortuko da:
OeqmSM=sqrt(0.5*((O1eeq-O2eeq)^2+(O1eeq-O3eeq)^2+(O2eeq-O3eeq)^2));
%tentsio baliokidearen ikurra zehazteko:
if (O1eeq+O2eeq+O3eeq>=0)
    OeqmSM=OeqmSM;
else
    OeqmSM=-OeqmSM;
end

%MS metodoaren hutsegite analisi berdina:
if (OeqmSM<=Oyp/cs)
    Nsm='Bizitza infinitua';
    axes(handles.snkurba2);
    img=imread('snkurbainf.PNG');
    imshow(img);
elseif (OeqmSM>Oyp/cs && OeqmSM<=Og)

    if (O1m>=0)
        ON1=Oyp*O1r/(Oyp-O1m);
    elseif (O1m<0 && O1r>=abs(O1m))
        ON1=O1r;
    elseif (O1m<0 && O1r<abs(O1m))
        ON1=O1r/2;
    end

```

```

guide

if (O2m>=0)
    ON2=Oyp*O2r/(Oyp-O2m);
elseif (O2m<0 && O2r>=abs(O2m))
    ON2=O2r;
elseif (O2m<0 && O2r<abs(O2m))
    ON2=O2r/2;
end
ON3=0;

ON=sqrt(0.5*((ON1-ON2)^2+(ON1-ON3)^2+(ON2-ON3)^2));

Oeprima=Oe/(cs);
logaritmoN=log10(Ne)-(log10(ON)-log10(Oeprima))*(log10(Ne)-
log10(Ng))/(log10(Og)-log10(Oeprima));
Nsm=10^logaritmoN;
if (Nsm>1000000)
    Nsm=9999999;
else
    Nsm=Nsm;
end
axes(handles.snkurba2);
img=imread('snkurba.PNG');
imshow(img);

else
    axes(handles.snkurba2);
    img=imread('lcf.PNG');
    imshow(img);
    Nsm='Ziklo baxuak!(ON>Og)';
end

%Eaitzak dagokien kutxetan erakusteko:
set(handles.O1m,'String',O1m);
set(handles.O2m,'String',O2m);
set(handles.O1r,'String',O1r);
set(handles.O2r,'String',O2r);
set(handles.Omeq,'String',Oeqm);
set(handles.Oreq,'String',Oeqr);
set(handles.Nms,'String',Nms);
set(handles.MSbaliokide,'String',OeqMS);
assignin('base','Nms',Nms);
assignin('base','OeqMS',OeqMS);

set(handles.O1eeq,'String',O1eeq);
set(handles.O2eeq,'String',O2eeq);
set(handles.Nsm,'String',Nsm);
set(handles.MSbaliokide,'String',OeqmSM);
assignin('base','Nsm',Nsm);
assignin('base','OeqmSM',OeqmSM);

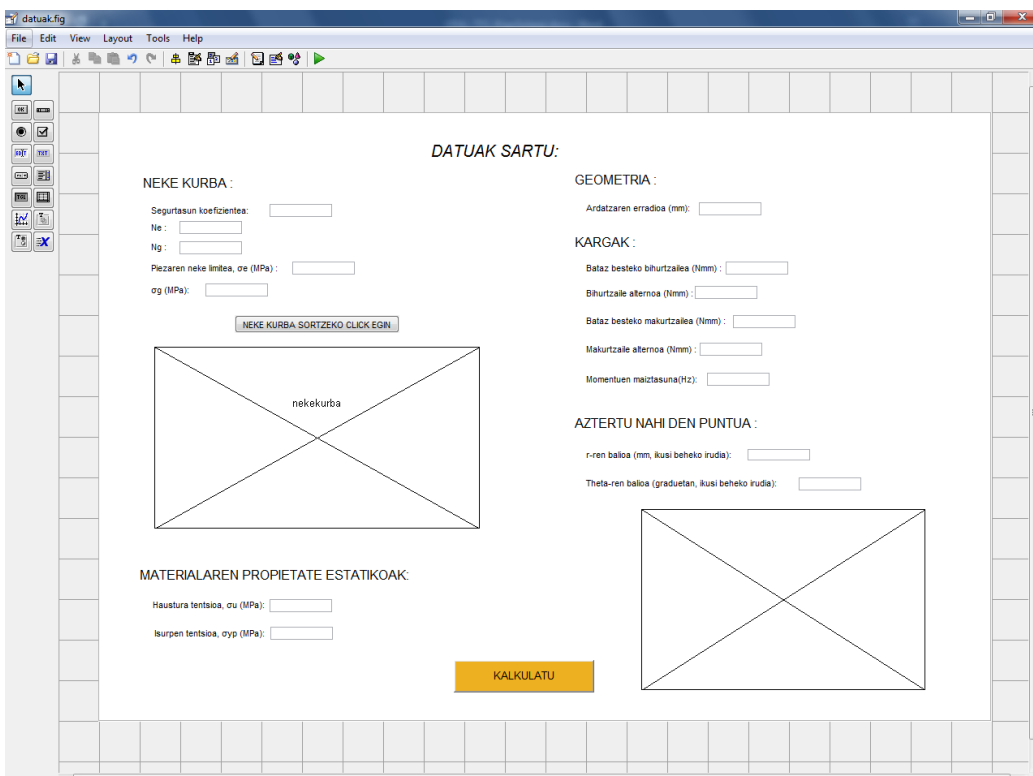
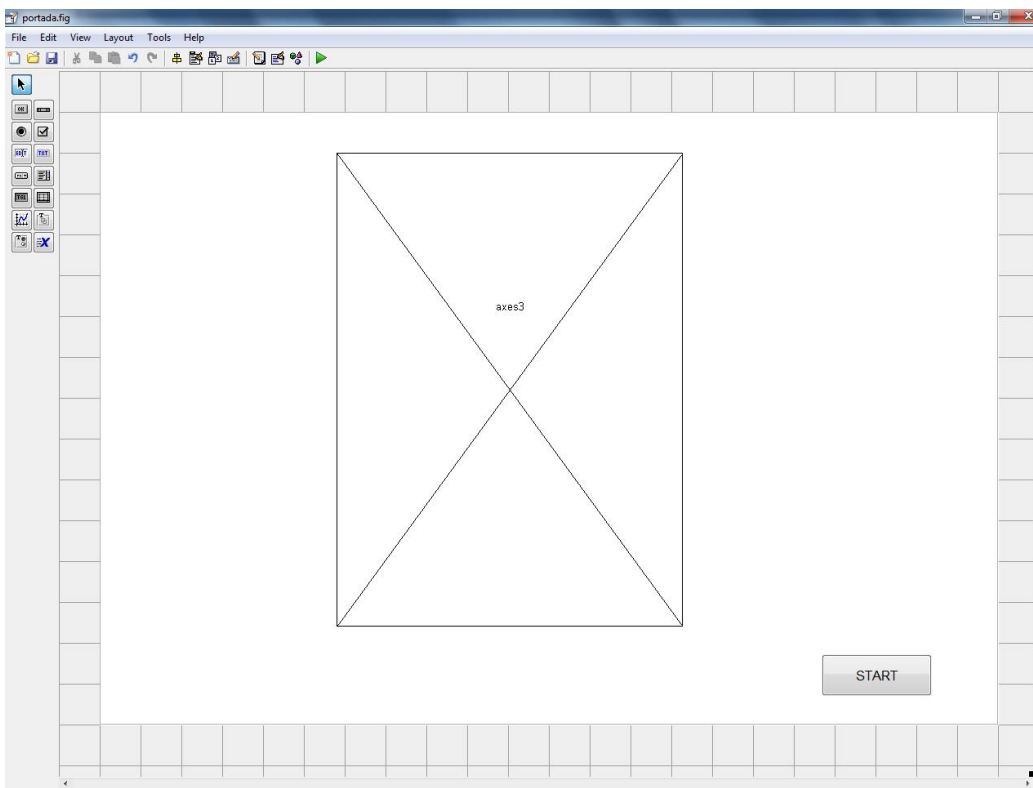
%Irukiak erakusteko:
axes(handles.sm2);
img=imread('sm2.PNG');
  
```

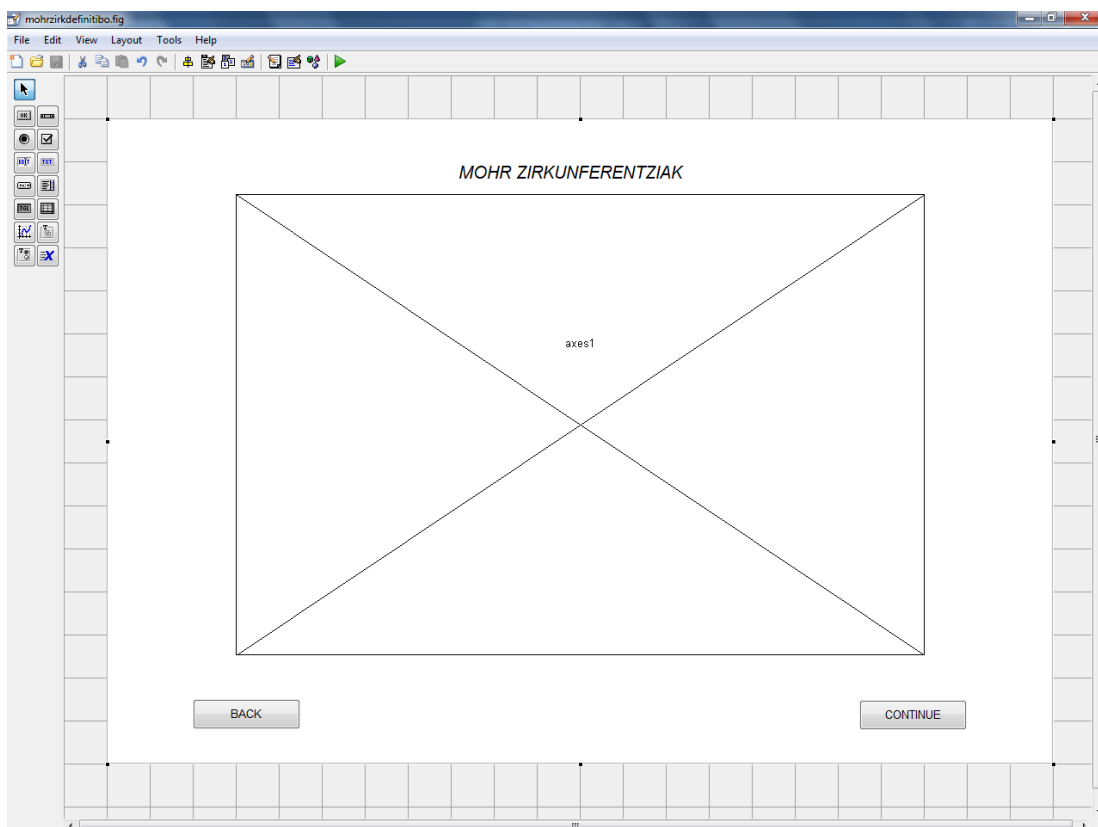
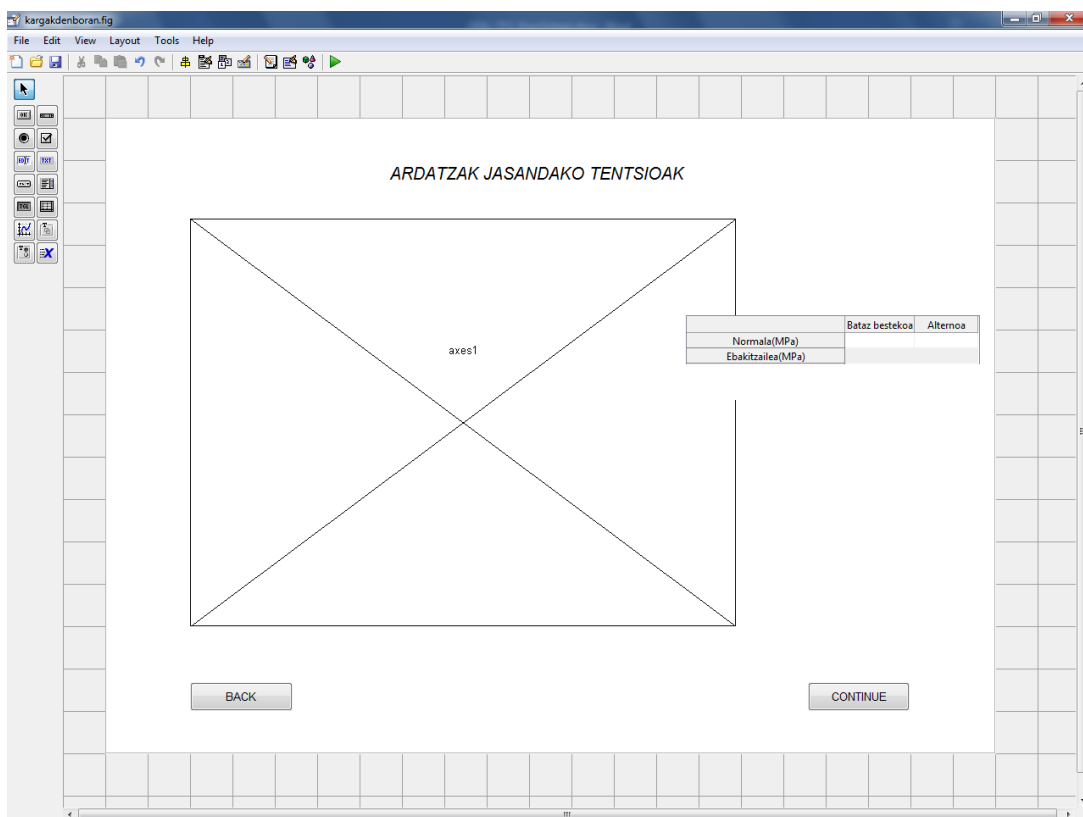
```
imshow(img);
axes(handles.sm3);
img=imread('sm3.PNG');
imshow(img);
axes(handles.ms1);
img=imread('sm1.PNG');
imshow(img);
axes(handles.ms2);
img=imread('ms2.PNG');
imshow(img);
axes(handles.ms3);
img=imread('ms3.PNG');
imshow(img);
axes(handles.giltza1);
img=imread('giltza.PNG');
imshow(img);
axes(handles.giltza2);
img=imread('giltza.PNG');
imshow(img);
axes(handles.giltza3);
img=imread('giltza.PNG');
imshow(img);
axes(handles.giltza4);
img=imread('giltza.PNG');
imshow(img);
axes(handles.soderberg1);
img=imread('soderberg.PNG');
imshow(img);
axes(handles.soderberg2);
img=imread('soderberg.PNG');
imshow(img);
axes(handles.mises1);
img=imread('mises.PNG');
imshow(img);
axes(handles.mises2);
img=imread('mises.PNG');
imshow(img);

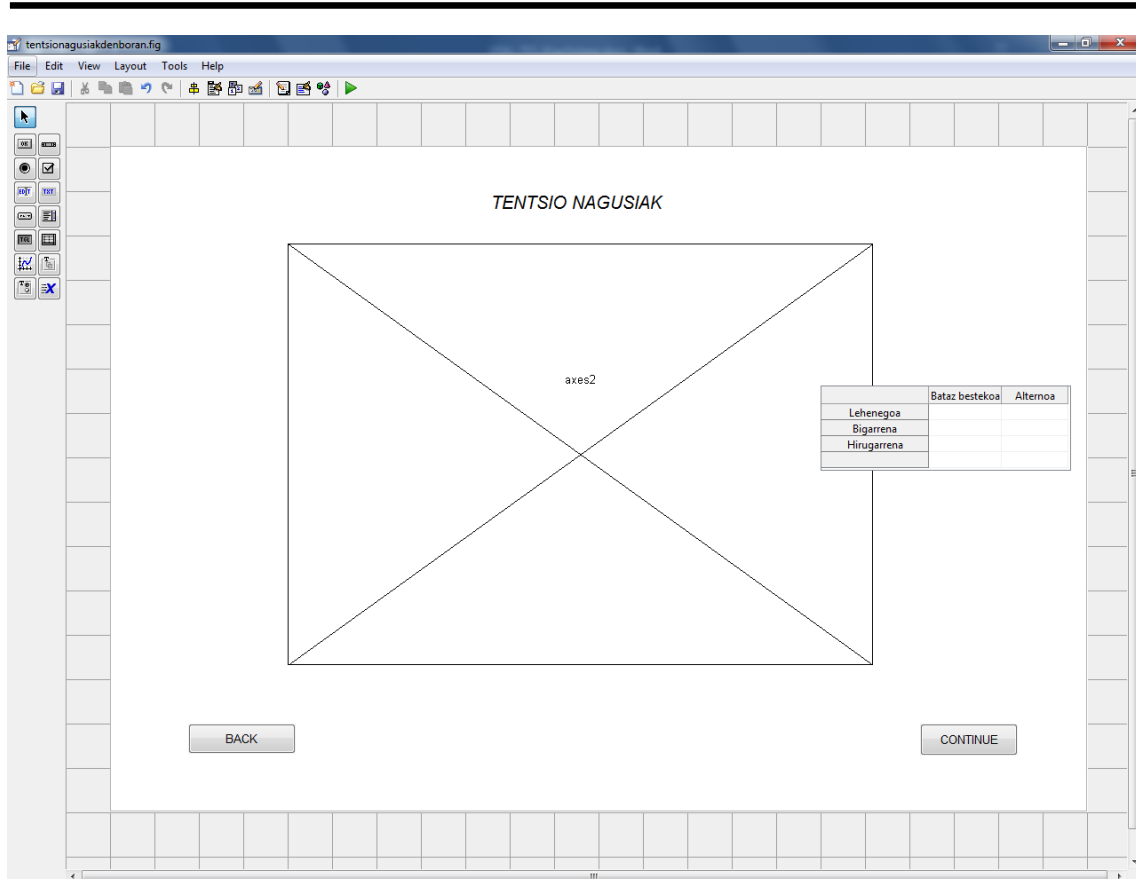
% --- Executes on button press in Endprogram.
function Endprogram_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;

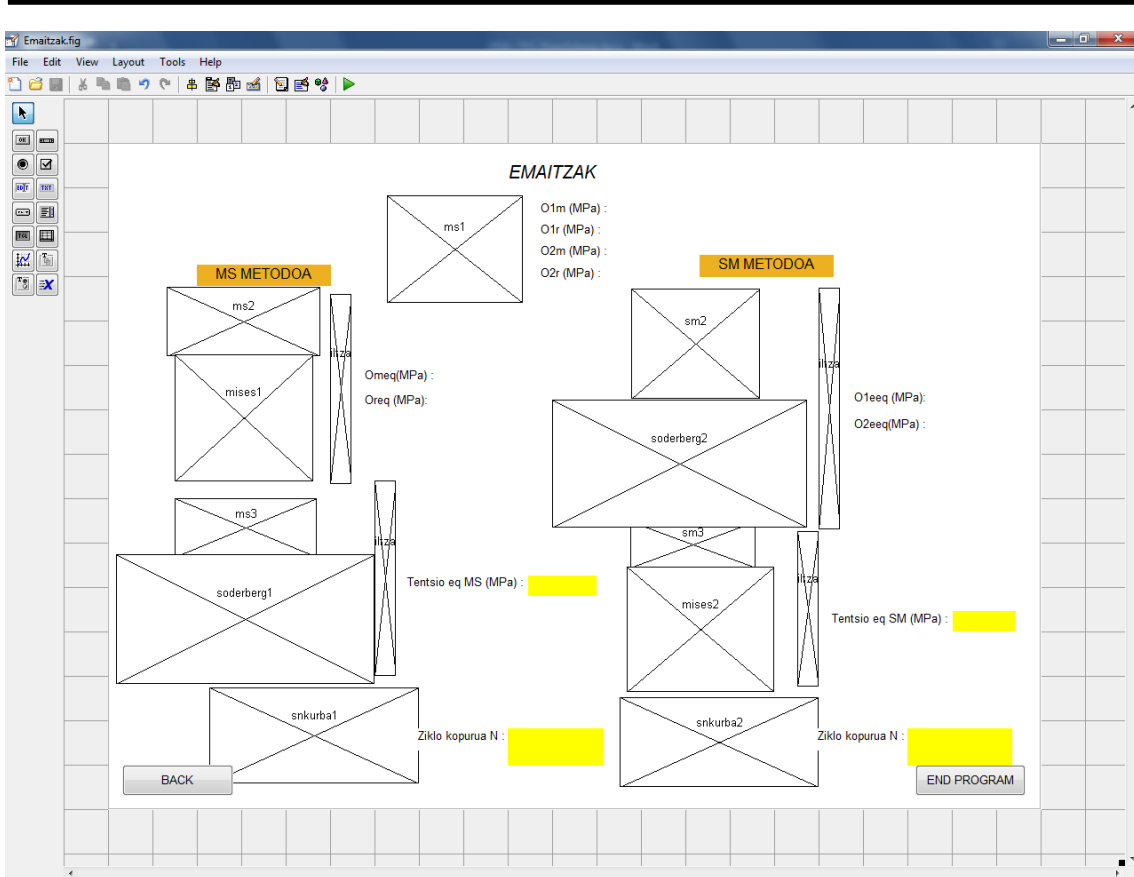
% --- Executes on button press in backbotoia.
function backbotoia_Callback(hObject, eventdata, handles)
clear all;
close all;
clc;
tentsionagusiakdenboran
```

14.2. ERANSKINAK: INTERFAZE GRAFIKOA.









14.3. ERANSKINAK: ESKUZKO ADIBIDEA.

Problemaren datuak:

Piezaren neke limitea, $\sigma_e = 200MPa$

$\sigma_g = 900MPa$

$N_e = 1000000$ ziklo

$N_g = 1000$ ziklo

$CS(\text{segurtasun koefizientea}) = 1,5$

materialaren haustura tentsioa, $\sigma_u = 1100MPa$

materialaren isurpen tentsioa, $\sigma_{yp} = 800MPa$

Piezaren erradioa, $r = 10mm$

Momentu bihurtzailearen bataz besteko zatia, $B_m = 100000 Nmm$

Momentu bihurtzailearen zati alternoa, $B_r = 100000 Nmm$

Momentu makurtzailearen bataz besteko zatia, $M_m = 100000 Nmm$

Momentu makurtzailearen zati alternoa, $M_r = 100000 Nmm$

Aztertu nahi den puntuaren koordenatuak: $R = 10mm, \theta = 0$

R: Ardatzaren zentritik aztertu nahi den punturaino dagoen distantzia, norabide erradialean neurtuta.

θ : Aztertu nahi den puntuaren eta norabide bertikaleko ardatzak osatzen duten angeluaren balio, graduetan neurtuta.

Lehenik eta behin ardatzaren inertzia momentuak eta jasaten duten tentsio normal eta ebakitzzaileak kalkulatu dira:

$$I = \pi * r^4 / 4 = 2500\pi$$

$$I_p = 2 * I = 5000\pi$$

Tentsio ebakitzzailea:

$$\tau_m = B_m * R / I_p = 63,6619 MPa$$

$$\tau_r = B_r * R / I_p = 63,6619 MPa$$

Tentsio normala:

$$\sigma_m = \frac{Bm * y}{I} = 127,3239 \text{ MPa}$$

$$\sigma_r = \frac{Bm * y}{I} = 127,3239 \text{ MPa}$$

non y , azertu nahi den puntuaren ardatzaren zentrotik norabide bertilean dagoen distantzia den.

Ondoren, tentsio nagusiak kalkulatu dira Elastikotasun teorema aplikatuz:

$$\sigma_{1m} = 153,69 \text{ MPa}$$

$$\sigma_{1r} = 153,69 \text{ MPa}$$

$$\sigma_{2m} = -26,370 \text{ MPa}$$

$$\sigma_{2r} = 26,370 \text{ MPa}$$

$$\sigma_{3m} = 0 \text{ MPa}$$

$$\sigma_{3r} = 0 \text{ MPa}$$

Orain, lehendabizi, SM metodoa aplikatu da tentsio baliokidea kalkulatzeko, horretarako Soderberg-en irizpidea erabiliz tentsio estatiko baliokideak kalkulatu dira:

$$\sigma_{1m} > 0 \text{enez} \rightarrow \sigma_{1eeq} = \sigma_{1m} + \sigma_{1r} * \frac{\sigma_{yp}}{\sigma_e} = 768,45 \text{ MPa}$$

$$\sigma_{2m} < 0 \text{ eta } \sigma_{2r} = \sigma_{2r} \text{enez} \rightarrow \sigma_{2eeq} = \sigma_{2r} * \frac{\sigma_{yp}}{\sigma_e} = 105,48 \text{ MPa}$$

$$\sigma_{3eeq} = 0$$

Azkenik, Von Mises-en irizpidearekin tentsio uniaxial estatiko baliokidea kalkulatu da baita hutsegitea emango denean igarotako ziko kopurua:

$$\sigma_{eqSM} = \sqrt[2]{0,5 * ((\sigma_{1eeq} - \sigma_{2eeq})^2 + (\sigma_{1eeq} - \sigma_{3eeq})^2 + (\sigma_{2eeq} - \sigma_{3eeq})^2)}$$

$$\sigma_{eqSM} = 721,52 \text{ MPa}$$

Tentsio baliokidea $> \sigma_{yp}/\sigma_s$ enez, piezak bizitza finitua edukiko du:

$$\sigma_{1m} > 0 \text{enez} \rightarrow \sigma_{N1} = \sigma_{yp} * \frac{\sigma_{1r}}{\sigma_{yp} - \sigma_{1m}} = 190,2368 \text{ MPa}$$

$$\sigma_{2m} < 0 \text{ eta } \sigma_{2r} = \sigma_{2r} \text{ denez} \rightarrow \sigma_{N2} = \sigma_{2r} = 26,370 \text{ MPa}$$

$$\sigma_{N3} = 0 \text{ MPa}$$

$$\sigma_N = \sqrt[2]{0,5 * ((\sigma_{N1} - \sigma_{N2})^2 + (\sigma_{N1} - \sigma_{N3})^2 + (\sigma_{N2} - \sigma_{N3})^2)} = 178,52 \text{ MPa}$$

Ziklo kopurua kalkulatu da Basquin-en kurba triangeluen antzekotasuna erabiliz:

$$\frac{\log(\sigma_g) - \log(\sigma^e/c_s)}{\log(N_e) - \log(N_g)} = \frac{\log(\sigma_N) - \log(\sigma^e/c_s)}{\log(N_e) - \log(N_{sm})}$$

$$N_{sm} = 347928,15$$

Orain MS metodoa erabiliz kalkuluak errepikatuko dira aurreko datu berdinak erabiliz.
Tentsio nagusiak:

$$\sigma_{1m} = 153,69 \text{ MPa}$$

$$\sigma_{1r} = 153,69 \text{ MPa}$$

$$\sigma_{2m} = -26,370 \text{ MPa}$$

$$\sigma_{2r} = 26,370 \text{ MPa}$$

$$\sigma_{3m} = 0 \text{ MPa}$$

$$\sigma_{3r} = 0 \text{ MPa}$$

Lehenik, Von Mises-en irizpidea erabiliko da tentsio egoera uniaxial baliokide bat lortzeko:

$$\sigma_{meq} = \sqrt[2]{0,5 * ((\sigma_{1m} - \sigma_{2m})^2 + (\sigma_{1m} - \sigma_{3m})^2 + (\sigma_{2m} - \sigma_{3m})^2)}$$

$$\sigma_{req} = \sqrt[2]{0,5 * ((\sigma_{1r} - \sigma_{2r})^2 + (\sigma_{1r} - \sigma_{3r})^2 + (\sigma_{2r} - \sigma_{3r})^2)}$$

$$\sigma_{1m} + \sigma_{2m} + \sigma_{3m} > 0 \rightarrow \sigma_{meq} = +168,43 \text{ MPa}$$

$$\sigma_{req} = 142,35 \text{ MPa}$$

Ondoren Soderberg-en irizpidea aplikatuko da:

$$\sigma_{meq} > 0 \text{ denez, } \rightarrow \sigma_{eqMS} = \sigma_{meq} + \sigma_{rer} * \frac{\sigma_{req}}{\sigma_{yp} - \sigma_{meq}} = 737,83 \text{ MPa}$$

$$\sigma_{eqMS} = 737,83 \text{ MPa}$$

Tentsio baliokidea $> \sigma_{yp}/cs$ denez, piezak bizitza finitua edukiko du eta hutsegitea emango den momentuan igarotako ziklo kopurua kalkulutako da, horretarako, lehenik σN kalkulatu da:

$$\sigma_{meq} > 0 \text{ denez, } \rightarrow \sigma N = \sigma_{yp} * \frac{\sigma_{req}}{\sigma_{yp} - \sigma_{meq}} = 180,31 \text{ MPa}$$

Ziklo kopurua kalkulatzeko:

$$\frac{\log(\sigma g) - \log(\sigma^e/cs)}{\log(Ne) - \log(Ng)} = \frac{\log(\sigma N) - \log(\sigma^e/cs)}{\log(Ne) - \log(Nms)}$$

$$Nms = 335594,8$$