

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Marijus Sebeckis

**OBJEKTŲ FIZIKINIŲ POVEIKIŲ MODELIAVIMO
NAUDOJANT BAIGTINIŲ ELEMENTŲ METODĄ TYRIMAS
IR PRITAIKYMAS MOBILIEMS ĮRENGINIAMS**

Baigiamasis magistro projektas

Vadovas
Dr. Šarūnas Pakevičius

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas
Doc. dr. Tomas Blažauskas

OBJEKTŲ FIZIKINIŲ POVEIKIŲ MODELIAVIMO
NAUDONAT BAIGTINIŲ ELEMENTŲ METODĄ TYRIMAS
IR PRITAIKYMAS MOBILIESIEMS ĮRENGINIAMS

Baigiamasis magistro projektas
Programų sistemų inžinerija (621E16001)

Vadovas
Dr. Šarūnas Packevičius

Recenzentas
Prof. Gintaras Palubeckis

Projektą atliko
Marijus Sebeckis

KAUNAS, 2015



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Marius Sebeckis

(Studento vardas, pavardė)

Programų sistemų inžinerija, 621E16001

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Objektų fizikinių poveikių modeliavimo naudojant spyruoklių metodą tyrimas ir pritaikymas mobiliems įrenginiams“

AKADEMINIO SĄŽININGUMO DEKLARACIJA

20 15 m. Gegužės 22 d.
Kaunas

Patvirtinu, kad mano **Marijaus Sebeckio** baigiamasis projektas tema „Objektų fizikinių poveikių modeliavimo naudojant baigtinių elementų metodą tyrimas ir pritaikymas mobiliems įrenginiams“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

TURINYS

1. ĮVADAS	5
1.1. Tyrimo sritis ir aktualumas	9
1.2. Darbo tikslas.....	10
1.3. Tyrimo objektas	10
1.4. Darbo uždaviniai.....	10
2. ANALIZĖ	11
3. BAIGTINIŲ ELEMENTŲ METODAS.....	23
4. PROJEKGINĖ DALIS	33
4.1. Projektavimo metodika	33
4.2. Architektūriniai reikalavimai.....	33
4.5. Sistemos statinis vaizdas.....	43
5. TYRIMO DALIS	44
5.1. Tikslas	44
5.2. Tyrimui naudojamų programų metrikų aprašymas	44
5.3. Tyrimui naudojamos programinės įrangos aprašymas.....	44
5.4. Rezultatai	46
5.5. Rekomendacijos	48
6. EKSPERIMENTINĖ DALIS.....	49
6.1. Tyrimo aplinka.....	49
6.2. Grafinis deformacijos kokybės įvertinimas	51
6.3. Deformacijos algoritmo greitaveikos tyrimas	55
7. IŠVADOS.....	58
11. LITERATŪRA.....	59
12. TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	62

PAVEIKSLĖLIŲ SĄRAŠAS

<i>Pav. 2.1</i>	<i>Konstrukcijos atsparumo įvertinimo schema</i>	11
<i>Pav. 2.2</i>	<i>Geometrinio objekto veikiamos išorinės jėgos: aktyviosios M_i, F_i, T, f ir aktyviosios: F_r, M_r</i>	13
<i>Pav. 2.3</i>	<i>Išorinių jėgų apkrovos kitimo atvejai</i>	14
<i>Pav. 2.4</i>	<i>Esminių įtempių grafinis vaizdavimas</i>	16
<i>Pav. 2.5</i>	<i>Esminių įtempių (a) ir oktaedrinių įtempių (b) koordinacių sistemos</i>	16
<i>Pav. 2.6</i>	<i>Tūrinio apkrovų veikiamas kūnas</i>	17
<i>Pav. 2.7</i>	<i>Paviršinių apkrovų veikiamo kūno pusiausvyra</i>	18
<i>Pav. 2.8</i>	<i>Plyšio plitimas geometriniuose objektuose: 1 – plyšio plitimas y ašimi; 2 – plyšio plitimas x ašimi; 3 – plyšio plitimas z ašimi</i>	20
<i>Pav. 3.1</i>	<i>Konstrukcijos elemento mazgus veikiančios jėgos</i>	27
<i>Pav. 4.1</i>	<i>Automobilio sudaužymo žaidimo panaudos atveju diagrama</i>	34
<i>Pav. 4.2</i>	<i>Sistemos paketų diagrama</i>	43
<i>Pav. 5.1</i>	<i>SourceMonitor v3.5 programinės įrangos išvaizda</i>	45
<i>Pav. 5.2</i>	<i>Bendra viso projekto metrikų suvestinė</i>	46
<i>Pav. 5.3</i>	<i>Atskirų projekto failų metrikos</i>	46
<i>Pav. 5.4</i>	<i>Bendra projekto rezultatų suvestinė</i>	47
<i>Pav. 5.5</i>	<i>Kiviat metrikų grafikas</i>	47
<i>Pav. 6.1</i>	<i>Eksperimentinė aplinka</i>	49
<i>Pav. 6.2</i>	<i>Eksperimentiniams duomenimis simuliuoti naudojamas geometrinis objektas</i>	51
<i>Pav. 6.3</i>	<i>Geometrinio objekto konstrukcijos elementų dingimas po kontakto</i>	52
<i>Pav. 6.4</i>	<i>Galinio bamperio konstrukcijos elementų poslinkis po kontakto su kitu geometriniu objektu</i>	52
<i>Pav. 6.5</i>	<i>Baigtinių elementų tinklelio trūkimas po deformacinės jėgos poveikio nagrinėjamai sričiai</i>	53
<i>Pav. 6.6</i>	<i>Geometrinio objekto konstrukcijos rezultatai po geometrinės sferos objekto nukritimo ant nagrinėjamo objekto paviršiaus</i>	54
<i>Pav. 6.7</i>	<i>Geometrinio objekto konstrukcijos rezultatai po sąveikos su reljefu</i>	55
<i>Pav. 6.8</i>	<i>Sistemos spartos tyrimas simuliuojant rezultatus Windows Phone emuliatoriumi</i>	56
<i>Pav. 6.9</i>	<i>Grafiniai laiko ir iteracijų kiekio priklausomybės rezultatai</i>	57

LENTELIŲ SARAŠAS

Lentelė 5.1 Naudojamų metrikų aprašymas	44
Lentelė 6.1 Iteracijų kiekio įtaka duomenų apdorojimo laikui.....	56

Sebeckis, M. „Objektų fizikinių poveikių modeliavimo naudojant baigtinių elementų metodą tyrimas ir pritaikymas mobiliesiems įrenginiams“ magistrinio baigiamasis projektas, vadovas dr. Šarūnas Pakevičius; Kauno technologijos universitetas, informatikos fakultetas, programų sistemų inžinerijos katedra.

Kaunas, 2015. 62 psl.

SUMMARY

With the development of mobile phone platforms and systems using numerical methods, it opens wide opportunities building a trustworthy engineering construction projects. Exploring models of real structures requires major computing resources costs, so it is important to find quick quality solutions. However, most of the engineering design area facing challenges where correct solution is not known. For this challenges are using approximate methods of calculation, which finding the most appropriate solution.

For several decades, the prevailing numerical methods, using engineering design computing technologies is the finite element method. This approach dealing with one of the difficulties, which affects the calculation results in an implicit dependence on the selected finite element mesh. Although this area is examined for decades, but the finite element generation mesh is one of the most widely investigated area.

Designing the model of the construction, is important to obtain reliable calculation results, calculation errors in assessing the specific case of the calculation. Qualitative evaluation of finite element solutions, can be used finite element strategies. This method is an iterative process managed solution from pre-defined tolerances and the various definitions of errors of assessment procedures. Using finite element strategies can be qualitatively assess the results obtained, which are sufficiently accurate models describing the structure in question, at different external influences.

Using finite element strategies and their application at issue, mechanical construction models are popular in computational mechanics and mechanical engineering challenges, extending the possibilities of numerical experiments.

1. ĮVADAS

1.1. Santrauka

Tobulėjant mobiliųjų telefonų platformoms bei skaitinius metodus naudojančioms sistemoms, atsiveria plačios galimybės kurti patikimus inžinerinių konstrukcijų projektus. Tyrinėjant realių modelių konstrukcijas reikalingos didelės skaičiavimo resursų sąnaudos, todėl ypač svarbu greitai rasti kokybiškai įvertintus projektų sprendinius. Tačiau dažniausiai inžinerinio projektavimo srityje susiduriama su uždaviniais, kurių tikslus sprendinys nėra žinomas ir tokiems uždaviniams taikomi apytiksliai skaičiavimo metodai, randantys tinkamiausią sprendinį.

Keletą dešimtmečių tarp vyraujančių skaitinių metodų, naudojančių inžinerinių konstrukcijų skaičiavimo technologijas, yra baigtinių elementų metodas. Šio metodo taikyme susiduriame su vienu iš sunkumų, kuris pasireiškia neišreikštine skaičiavimo rezultatų priklausomybe nuo pasirinkto baigtinių elementų tinklo. Nors ši sritis yra nagrinėjama jau dešimtmečius, tačiau baigtinių elementų tinklo generavimas yra vienas tarp plačiausiai nagrinėjamų sričių.

Projektuojant modelių konstrukcijas svarbu išgauti patikimus skaičiavimo rezultatus, įvertinant skaičiavimo paklaidas konkrečiu skaičiavimo atveju. Kokybiškai įvertinus baigtinių elementų sprendinius, galima taikyti prisitaikančiąsias baigtinių elementų strategijas. Šis metodas yra iteracinis procesas, valdomas iš anksto apibrėžtomis sprendinio paklaidomis bei įvairiomis apibrėžtomis paklaidų vertinimo procedūromis. Taikant baigtinių elementų strategijas, galima kokybiškai įvertinti gaunamus rezultatus, kurie yra pakankamai tikslūs, aprašantys nagrinėjamus konstrukcijų modelius, esant įvairiems išoriniams poveikiams.

Taikant baigtinių elementų strategijas bei jų pritaikymą nagrinėjamiems mechaninių konstrukcijų modeliams yra populiarus skaičiuojamasis mechanikos ir mechanikos inžinerinių uždavinių, praplečiantis skaitinių eksperimentų galimybes.

1.2. Tyrimo sritis ir aktualumas

Rinkoje didėjant išmaniųjų mobiliųjų telefonų pasiūlai, jie sparčiai keičia įprastinius mobiliuosius telefonus. Nors ir senuosiuose mobiliuosiuose įrenginiuose yra svarbiausios telefono funkcijos, tačiau naujos technologijos leidžia vartotojams pateikti daugiau įdomių ir naudingų funkcionalumo galimybių. Viena iš patraukliausių vartotojams siūlomų periferinių funkcijų yra žaidimai.

Žaidimų paklausa mobiliuose telefonuose pasitvirtino jau ankstyvoje mobiliųjų telefonų rinkoje. Nors mobiliųjų telefonų ekranai turėjo ribotą dydį, spalvų raišką ir labai ribotus skaičiavimo resursus, tai netrukdė vartotojams pasiūlyti žaidybinio tipo programėles, pritaikytas mobiliesiems įrenginiams. Vienas iš pirmųjų, pačių primityviausių, siūlomų žaidimų vadinosi „*Gyvatėlė*“. Tačiau vystantis technologijoms ir plečiantis mobiliųjų įrenginių galimybėms, dabar galima naudoti žaidimų karkasus ir sukurti naujus, kur kas įdomesnius žaidimus, kurie rinkoje gali susilaukti didelio susidomėjimo.

Išanalizavus šiuo metu rinkoje esančius žaidimus, magistrinio darbo objektui pasirinktas automobilių lenktynių žaidimų tipas dėl jų populiarumo ir asmenių interesų. Analizuojant konkurentų rinkoje esančius produktus pastebėta, kad jų žaidimams trūksta grafinio realistiškumo, kurį planuojame sukurti suteikdami geometriniams objektams vizualines deformacijas. Tokio geometrinio funkcionalumo žaidimų rinkoje dar pakankamai mažai, todėl darome prielaidą, kad šis sprendimas pasiteisins ir bus įdomus tiek kuriant šią sistemą, tiek ją naudojant vartotojui.

1.3. Darbo tikslas

Pagrindinis darbo tikslas - sukurti lenktynių žaidimą, kuriame būtų realizuotas baigtinių elementų deformacijos algoritmas, leidžiantis greitai rasti kokybiškiausią sprendimą mechaniniuose deformacijos uždaviniuose. Ši sistema bus optimizuota mobiliams įrenginiams. Optimizacija apims deformacijos algoritmų bei grafikos apdorojimo sistemos pritaikymą ribotiems skaičiavimo resursams, kurių turi pakakti mobiliams įrenginiams su vidutinės spartos skaičiavimo resursais.

1.4. Tyrimo objektas

Darbe nagrinėjami netamprios ir nevienalytės mechaninės struktūros geometriniai objektai, kurie modeliuojami kaip trimačiai geometriniai objektai mobiliuosiuose įrenginiuose. Šie geometriniai objektai yra tam tikros formos. Darbe nagrinėjamas išorinių jėgų poveikis geometriniam objektui, sukeliantis deformacijos procesą.

1.5. Darbo uždaviniai

1. Apžvelgti baigtinių elementų metodo algoritmą, jo principus.
2. Ištirti geometrinių objektų deformacijos elgsenas bei jų procesiškumą.
3. Įvertinti skaičiavimo poreikius, reikalingus deformacijos apskaičiavimui, naudojant baigtinių elementų algoritmą.
4. Įvertinti geometrinių objektų deformacijos vizualizavimo tikslumą.
5. Optimizuoti baigtinių elementų algoritmą, kuris bus taikomas objekto deformacijos procese.

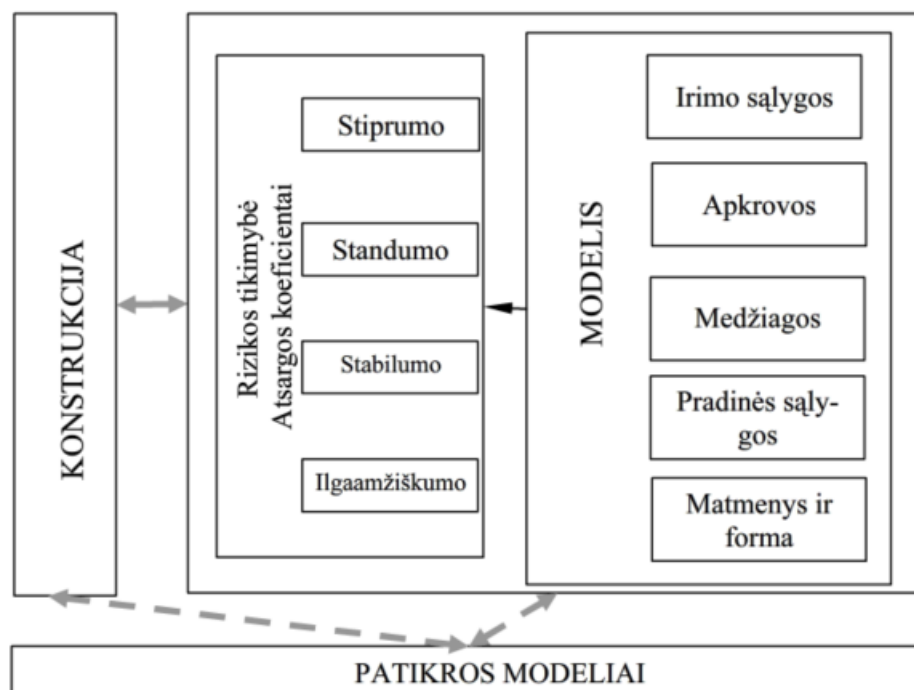
2. ANALIZĖ

Geometrinių objektų konstrukcijas veikiančios išorinės apkrovos jėgos ir dėl jų atsirandanti vidinė apkrovos sukelia konstrukcijų deformacijas – matmenų ir formos pokyčius. Medžiagų atsparumo savybės apibūdinamos deformavimo diagramomis ir ribinėmis būsenomis. Šios savybės priklauso nuo aplinkos ir apkrovos sąlygų.

Skaitiniai metodai plačiai naudojami sprendžiant geometrinių kūnų deformacijos uždavinius. Jie yra pagrįsti konstrukcijų geometrijos, medžiagų ir apkrovų skaidymu į mažesnes sritis. Šiose nagrinėjamuose srityse vykstantys reiškiniai išreiškiami naudojant paprastas matematinės funkcijas.

Reiškinų supaprastinimas leidžia atsirasti skaičiavimo paklaidoms. Paklaidų pobūdis priklauso nuo supaprastinimų laipsnio. Tačiau būtent nuo modeliavimo tikslumo priklauso uždaviniams spręsti reikalingo skaičiavimo kiekis, trukmė ir kaina.

Skaitinės deformuojamų kūnų mechanikoje uždavinių sprendimas susijęs su modelių sudėtingumo ir skaičiavimo paklaidų minimizavimu. Todėl sprendžiant šiuos uždavinius, svarbu surasti ekstremalias funkcijų vertes.



Pav. 2.1 Konstrukcijos atsparumo įvertinimo schema

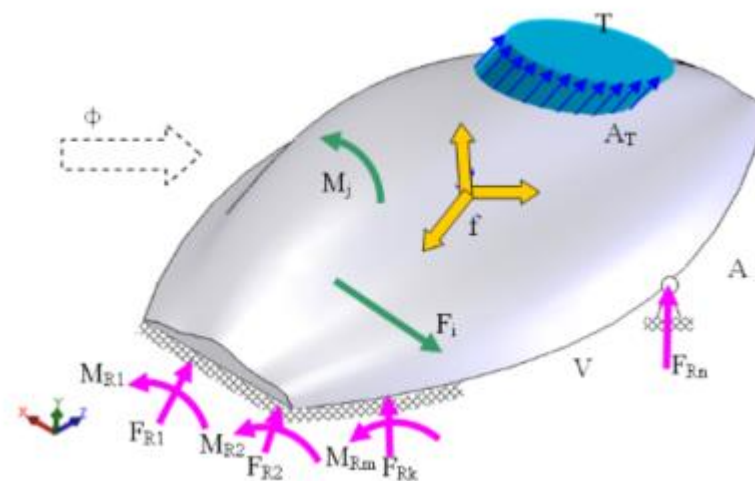
Geometrinio kūno konstrukcijos atsparumo modelis priklauso nuo jo matmenų ir formos. Pasirenkami sprendimo metodai, skaidymo pobūdis ir tankis, numatomos modelio patikros galimybės. Medžiagų modeliai pateikia priklausomybes, apibūdinančias medžiagų elgseną: nuo pasipriešinimo lėtai arba greitai besikeičiantis apkrovoms ir kitokiam išorės poveikiui iki pat jų ribinių būsenų – visiško pasipriešinimo išnykimo. Apkrovų modeliavimas yra susijęs su didele aibe atskirų arba sujungtų funkcijų į bendrą modelį, kuris nusako priklausomybę nuo apkrovų dydžio, pasiskirstymo ir kitimo per konkretų laiką. Pradinės sąlygos apibūdina objekto pradinę būseną ir aplinkos reakcijos pobūdį.

Projektuojamas konstrukcijos skaitinis modelis turėtų būti kuo paprastesnis, bet pakankamai aukšto tikslumo, atspindintis modeliuojamą objektą. Paruošimas gali būti sudėtingas dėl daugelio parametrų ir pasirinkčių, kurias reikia apibrėžti. Netikslus arba netinkamas jų įvertinimas yra viena iš dažniausiai pasitaikančių priežasčių.

Geometrinio objekto modelio matmenys, medžiagų savybės arba apkrovos yra atsitiktiniai parametrai ir jų neapibrėžtumai yra nusakomi tikimybinėmis charakteristikomis. Todėl konstrukcijos atsparumo analizės rezultatai yra atsitiktiniai dydžiai, o jų tikimybinės charakteristikos apibūdina konstrukcijos irimo tikimybę.

Apkrovos yra skirstomos į išorines ir vidines. Išorinės apkrovos yra tokios jėgos, kurių atsiradimo priežastys yra išorinės aplinkos veikiamos jėgos – sąveika su kitais kūnais ir fizikinių laukų poveikis. Išorinės aplinkos jėgos gali veikti ir konstrukcijos viduje kaip vidinis slėgis arba gravitacinės apkrovos. Vidinės apkrovos – tai gniuždymas ir įtempimai, kurie apibrėžiami pagal priimtą koordinačių sistemą. Išorinių ir vidinių apkrovų pusiausvyrą išreiškia statistikos ir dinamikos lygtimis.

Išorinės apkrovos jėgos atsiranda dėl konstrukcijos sąveikos su aplinka ir gali būti sutelktos viename taške arba pasiskirsčiusios išilgai linijos, plote ar tūryje. Jos gali būti statinės arba kintančios. Šilumos, elektromagnetinio lauko, radioaktyvių dalelių poveikiai medžiagoms taip pat gali būti suvokiami kaip apkrovos komponentai. Apkrovos, kurios atsiranda konstrukcijų įtvirtinimo vietose yra vadinamos reakcijomis. Išorinės jėgos veikiančios geometrinį objektinį kūną, kurio paviršiaus plotas A , o tūris V parodytas žemiau esančiame paveikslėlyje.



Pav. 2.2 Geometrinio objekto veikiamos išorinės jėgos: aktyviosios M_i , F_i , T , f ir aktyviosios: F_r , M_r

Čia aktyviosios apkrovos yra M_i , F_i , T , f , o reakcijos – F_r , M_r .

Apkrovos jėgos pasirinktoje koordinačių sistemoje išreiškiamos jų projekcijomis, kurios trimatėje koordinačių sistemoje galima išreikšti vektoriais:

$$f = [f_x, f_y, f_z]^T,$$

$$T = [T_x, T_y, T_z]^T,$$

$$F_i = [F_{xi}, F_{yi}, F_{zi}]^T,$$

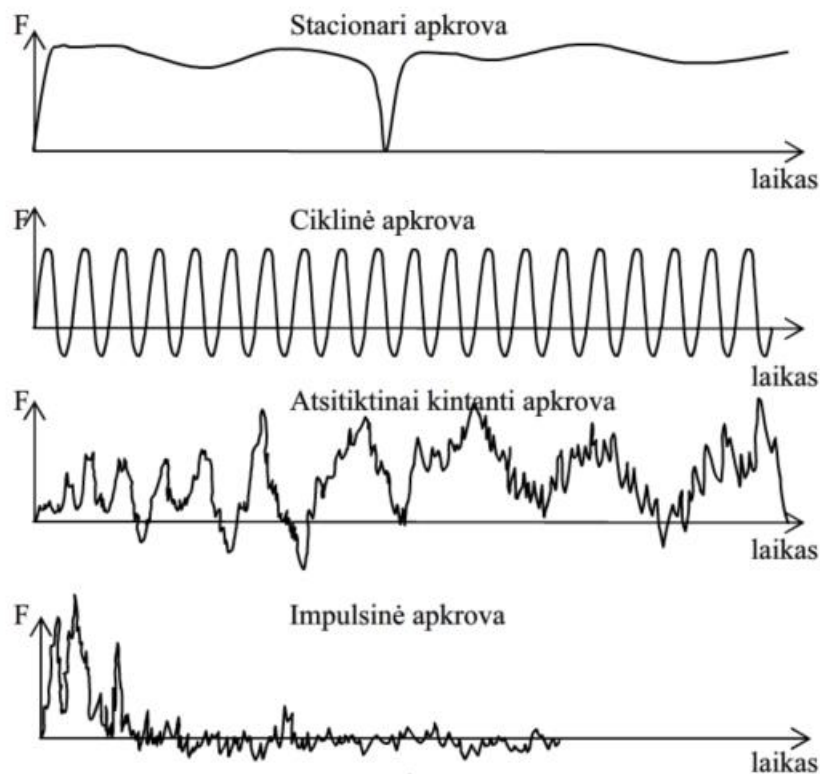
$$M_j = [M_{xj}, M_{yj}, M_{zj}]^T,$$

$$F_{Rn} = [F_{xn}, F_{yn}, F_{zn}]^T,$$

$$M_{Rm} = [M_{xm}, M_{ym}, M_{zm}]^T.$$

Sutelktosios arba taškinės apkrovos (F_i , M_i) veikia mažuose ploteliuose – guoliuose, ratų kontakto taškuose. Išskirstytosios apkrovos veikia didesnėse srityse – išilgai linijos, paviršiuje arba tūryje. Linijinės apkrovos gali būti krūvio veikiamą siją, paviršinės apkrovos jėgos – lėktuvo sparną veikiančios aerodinaminės jėgos ar panardinto kūno hidrostatinės jėgos, tūrinės apkrovos, atsirandančios dėl gravitacijos ar inertinio poveikio. Šių apkrovų pasiskirstymą galima apskaičiuoti taikant skaitinius metodus. Atsparumo skaičiavimuose išskirstytos apkrovos dažniausiai yra pakeičiamos sričių, kuriose jos veikia, atstojamųjų išskirstymu mazguose.

Statinės apkrovos yra jėgos, kurios nekinta arba kinta lėtai. Tai gali būti automobilio konstrukcijas veikiančios savojo ir gravitacinė apkrovos. Jas nustatant reikia atsižvelgti į konstrukcijų savuosius svyravimus ir medžiagų nuovargį. Dinaminis poveikis yra aktualus, kai apkrovimo greitis yra didesnis už trečdalį savųjų svyravimų greičio. Esant dideliame apkrovimo greičiui, kuris būdingas smūgiams ar sproginimams, būtina atsižvelgti į tai, kad medžiagų konstrukcinės savybės priklauso nuo deformacijos greičio. Išorinių apkrovos jėgų priklausomybių pasiskirstymo laike atvejai pateikti žemiau esančiame paveikslėlyje.



Pav. 2.3 Išorinių jėgų apkrovos kitimo atvejai.

Vidinės jėgos vadinamos įrašomis. Šios jėgos atsiranda dėl išorinių apkrovos jėgos lenkimo, sukimo. Vidinių jėgų pasiskirstymo intensyvumą apibūdina įtempimai. Atsižvelgiant į įtempimo jėgos kryptis, jie skirstomi į normalinius σ ir tangentinis τ :

$$\sigma = \{ \sigma_x, \sigma_y, \sigma_z, \tau_{yx}, \tau_{zx}, \tau_{xy} \}^T$$

Kintant koordinačių sistemos padėčiai, keičiasi ir įtempimų komponentų vertės. Koordinačių sistemoje galima rasti padėtį, kuriai esant tangentiniai įtempimai tampa lygūs nuliui. Nagrinėjamoje srityje įtempimai įgyja didžiausias vertes ir yra svarbiausiais įtempiniais:

$$\sigma_1 > \sigma_2 > \sigma_3.$$

Čia $\sigma_1 = \sigma_{max}$ ir $\sigma_3 = min$. Jie apskaičiuojami pagal:

$$\sigma^3 - I_1\sigma^2 + I_2\sigma - I_3 = 0,$$

Čia I_1, I_2, I_3 - įtempių invariantai:

$$I_1 = \sigma_1 + \sigma_2 + \sigma_3,$$

$$I_2 = \sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_z\sigma_x - \tau_{xy}^2 - \tau_{yz}^2 - \tau_{zx}^2,$$

$$I_3 = \sigma_x\sigma_y\sigma_z + 2\tau_{xy}\tau_{yz}\tau_{zx} - \sigma_x\tau_{xy}^2 - \sigma_y\tau_{yz}^2 - \sigma_z\tau_{zx}^2.$$

Svarbiausi įtempiai $\sigma_1, \sigma_2, \sigma_3$ išreiškiami įtempių žvaigžde:

$$\sigma_1 = \frac{I_1}{3} + c \cos \Theta,$$

$$\sigma_2 = \frac{I_1}{3} + c \cos \left(\Theta - \frac{2\pi}{3} \right),$$

$$\sigma_3 = \frac{I_1}{3} + c \cos \left(\Theta + \frac{2\pi}{3} \right),$$

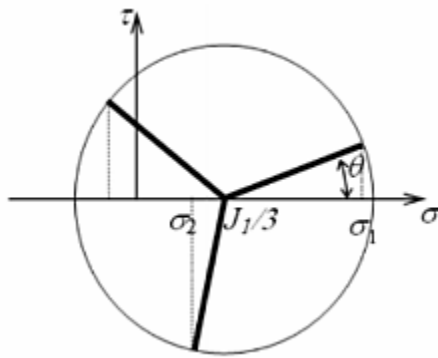
kur

$$a = \frac{I_1^2}{3} - I_2,$$

$$b = -2 \left(\frac{I_1}{3} \right)^3 - \frac{I_1 I_2}{3} - I_3,$$

$$c = 2 \sqrt{\frac{a}{3}},$$

$$\Theta = \frac{1}{3} \arccos \left(-\frac{3b}{ac} \right).$$



Pav. 2.4 Esminių įtempių grafinis vaizdavimas

Esant visiems šios lygties sprendiniams nelygiems nuliui, įtempių būvis yra erdvinis. Esant vienam iš sprendinių lygiam nuliui, įtempių būvis yra plokščiasis. Esant dviem sprendiniams lygiams nuliui, įtempimas yra vienašis. Ašinės simetrijos masyvuose arba plonasieniuose kūnuose vienai ašiai įtempių būvis panašus į plokščiąją deformacijų būvį.

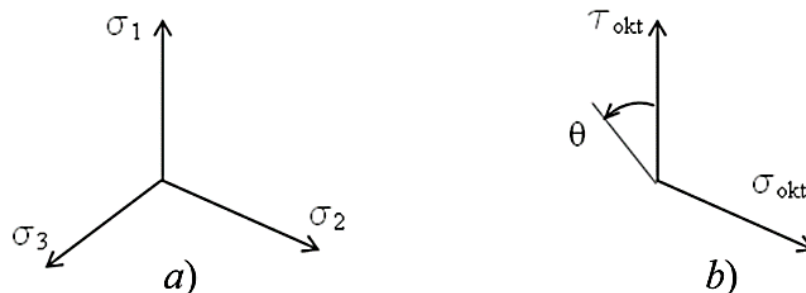
Grafinį įtempių būvį yra patogiau pavaizduoti trimis jo komponentais svarbiausių įtempių koordinačių sistemoje, kurios trimis kampais apibūdina įtempių būvį. Įtempių būvį galima pavaizduoti su oktaedrine plokštuma susijusioje cilindrinėje koordinačių sistemoje, kurios viena ašis yra normalinis oktaedrinis įtempis:

$$\sigma_{oct} = \left(\frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \right)$$

Antroji – Tangentinis oktaedrinis įtempis:

$$\tau_{oct} = \sqrt{\frac{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}{2}}$$

Trečioji – Įtempių būvio kampas Θ .



Pav. 2.5 Esminių įtempių (a) ir oktaedrinų įtempių (b) koordinačių sistemos

Svarbiausiųjų įtempių koordinatėse medžiagų būseną yra apibrėžiama įtempių segmente, kuriame $\sigma_1 > \sigma_2 > \sigma_3$. Oktaedrinėse įtempių koordinatėse galioja šios sąlygos: $\tau_{okt} > 0$ ir $0 < \varphi < \frac{\pi}{3}$.

Deformuojamų kūnų statikos lygtimi, nagrinėjama geometrinio kūno srityje, išreiškiama išorinių ir vidinių jėgų pusiausvyra:

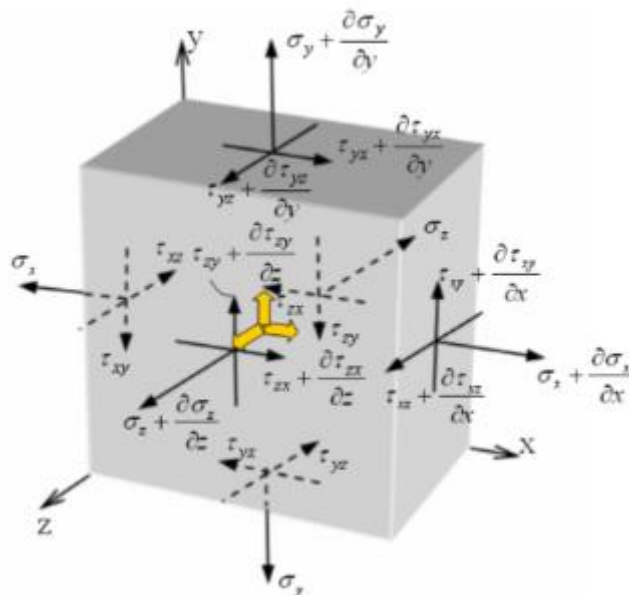
$$K \cdot Q = F$$

Čia $K \cdot Q$ – tampriojo deformavimo sukeltos vidinės jėgos. F – išorinės jėgos. K – standumo koeficientas, Q – deformuoto kūno poslinkių masyvas. Statinėms apkrovoms darant prielaidą, kad konstrukcijos savybės laikui bėgant nesikeičia.

Kūno išorinių ir vidinių apkrovų pusiausvyrai išreikšti naudojama dinamikos lygtis. Lygtyje yra įvertinamos inercijos, slopinimo ir vidinės jėgos.

$$M \cdot \ddot{Q} + C \cdot \dot{Q} + K \cdot Q = F$$

Čia $M \cdot \ddot{Q}$ – inercijos apkrova, $C \cdot \dot{Q}$ – slopinimo apkrova, K, M, C – kūno standumo, masės ir slopinimo charakteristikos, \ddot{Q}, \dot{Q}, Q – pagreičių, greičių ir poslinkių masyvai. Visi komponentai šioje lygtyje laikui bėgant gali kisti.



Pav. 2.6 Tūrinio apkrovų veikiamas kūnas

Kai kūnas veikiamas tūrinės išskirstytosios apkrovos f , tai medžiagos elemento pusiausvyra apibrėžiama:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + f_x = 0,$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + f_y = 0,$$

$$\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + f_z = 0,$$

Čia f_x, f_y, f_z – tūrinių išskirstytųjų apkrovų projekcijos.

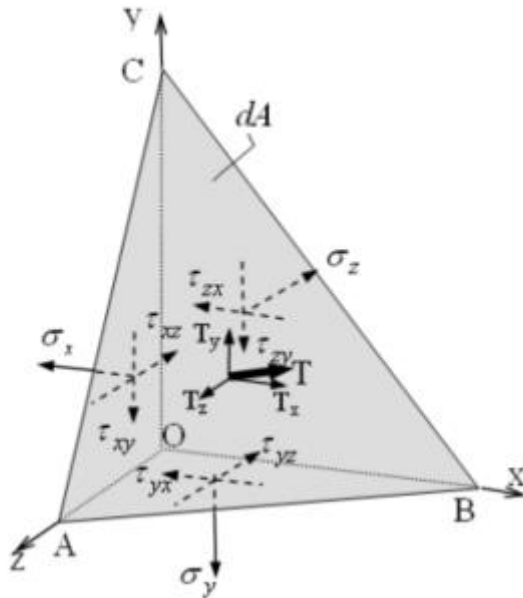
Jeigu paviršius, kurio normalė n , yra apkrautas išskirstytuoju krūviu T , tuomet medžiagos elemento OABC (žr.pav.2.7) pusiausvyros lygtis yra:

$$\sigma_x n_x + \tau_{yx} n_y + \tau_{zx} n_z = T_x,$$

$$\sigma_{xy} + \sigma_y n_y + \tau_{zy} n_z = T_y,$$

$$\tau_{xz} n_x + \tau_{yz} n_y + \sigma_z n_z = T_z;$$

Čia n_x, n_y, n_z – paviršiaus, slegiamas jėgos T , normalės projekcijos.



Pav. 2.7 Paviršinių apkrovų veikiamo kūno pusiausvyra

Pusiausvyros lygtys išreiškia išorines apkrovas ir įtempius. Įtempius išreikšus deformacijomis (tampriajai deformacijai naudojama Huko dėsnis), o deformacijas poslinkiais, gaunama statikos lygtis, kuri apibūdina apkrovų ir poslinkių tarpusavio priklausomybę.

Medžiagos deformuojamos veikiant jas apkrovomis ir įtempiais, todėl konstrukcijos praranda savo pradinę struktūrą. Dėl deformacijos atsiranda konstrukcijos taškų poslinkiai. Tam tikruose konstrukcijos vietose atsiranda tam tikro dydžio ir krypties poslinkiai. Deformacija apibūdinama kaip konstrukcijos matmenų ir formų pokyčiai, kuries priklauso nuo medžiagų savybių ir veikiančių įtempių.

Medžiagų struktūra, sandara, aplinkos veiksniai, apkrovimo jėgos įtakoja jų stiprumą, patvarumą, tamprumą plastiškumą, tįsumą. Deformuojamų medžiagų modeliai, pagal jų sandaros aprašymo pobūdį, gali būti skirstomi į fizikinius, inžinierinius-fizikinius ir inžinierinius. Fizikiniuose modeliuose medžiagos nagrinėjamos atomų ir molekulių sąveikos lygmenyje. Inžinieriniuose-fizikiniuose nagrinėja medžiagų grūdelių būseną. Inžinieriniuose – medžiaga laikoma vienalyte. Sprendžiant mechaninius kūno deformuojamų uždavinius, dažnai taikomi inžinieriniai medžiagų modeliai.

Konstrukcijos skaitinis modelis, dar vadinamas skaičiuojamąją schema, sudaromas siekiant minimaliomis sąnaudomis patikimai įvertinti konstrukcijos atsparumą. Skaitinis modelis – tai realios konstrukcijos matematinis modelis, kuriame apytikriai pavaizduotos svarbiausios funkcinės priklausomybės. Šios priklausomybės apibūdinamos remiantis esama teorinėmis ir praktine patirtimi, konstrukcijos savybėmis ir su jų neapibrėžtumais t.y. matmenimis, forma, apkrovomis, medžiagomis ir jų ribinėmis būsenomis.

Skaitinio modelio patikimumas parodo, kaip šis modelis atitinka realios geometrinio objekto konstrukcijos atsparumą, kai ji yra įprastos darbinės arba kritinės būsenos.

Konstrukcijos elementų geometrinės charakteristikos gali būti paprastos ir sudėtingos. Norint tiksliai jas perteikti skaitmeniniam modelyje, kartais reikia labai smulkiai suskaidyti nagrinėjama objekto sritį, o tai padidina sprendžiamų uždavinių apimtį, jiems išspręsti reikalingi didesni skaičiavimo resursai. Ruošiant skaitinius modelius ar sprendžiant analitiškai, verta atsisakyti kai kurių sudėtingų formų, neturinčių didesnės įtakos rezultatams.

Įvairūs fizikiniai defektai, veikiantys mechanikoje, apsunkina konstrukcijos elemento stiprumo skaičiavimą. Taikant tik klasikinius kietojo kūno mechaninius deformacijos metodus, gaunami rezultatai ne visada atspindi geometrinių objektų deformacijos rezultatus. Daugybė deformacinio pobūdžio mechaninių veiksnių privertė naujai įvertinti medžiagų stiprumą ir patikimumą. Taip atsirado nauja kryptis, analizuojanti kietojo kūno deformacijas mechanikoje – irimo mechanika.

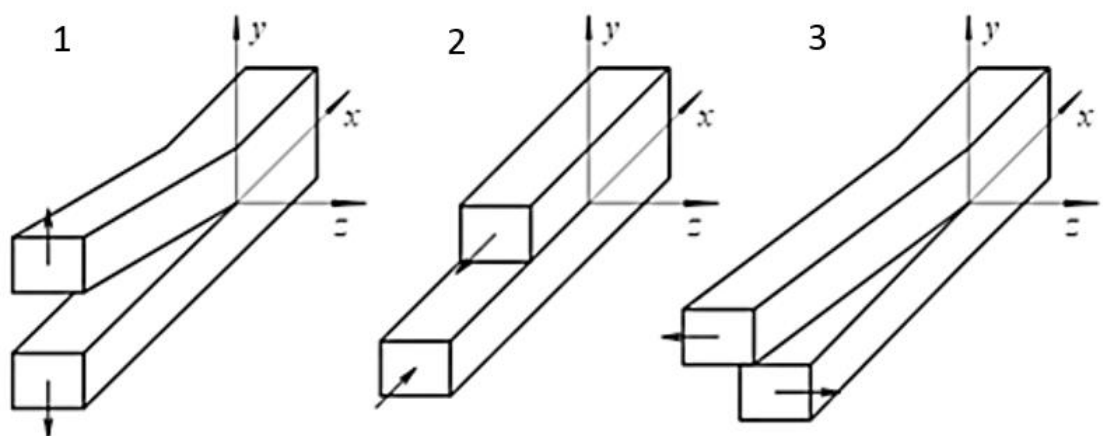
Irimo mechanikoje tikslas yra identifikuoti bet kokio kūno formos deformacijos sąlygas, veikiant kūną išorinėms jėgoms bei žinomoms apkrovoms. Irimo modelio identifikavimas galimas keliais aspektais:

1. įrimo modelio sudarymas, remiantis teorinėmis žiniomis, eksperimentiniais duomenimis;
2. pritaikant jau žinomus irimo modelius.

Realūs kūno irimo procesai priklauso nuo struktūros, cheminių savybių, defektų įtakos, apkrovos tipo, aplinkos įtakos ir kitų veiksnių.

Geometrinė konstrukcija nesuyra tik prasidėjus deformacijos procesui. Didelę laiko dalį eksploatacijos metu irimas nepereina į tiesioginį medžiagos plyšio plitimą dėl cheminių medžiagų savybių. Medžiagos išlaiko savo struktūrą ir geometrinės kūno formas. Kietojo kūno molekulės erdvėje išsidėsto tankiai ir tvarkingai. Laisvai judėti jos negali, todėl tik svyruoja apie pusiausvyros padėtį. Dėl šios priežasties kietieji kūnai išlaiko ne tik tūrį, bet ir formą. Tačiau esant didelėms deformacijos jėgoms, geometrinis objektas ima deformuotis dėl jį veikiančių padidėjusių išorinių jėgų. Deformacijos procese dėl veikiančių kūno jėgų geometriniam paviršiui, irimo procesas pereina į tiesioginį plyšio plytimą, todėl plyšys gali atsirasti geometriniam objektui.

Plyšį galima apibūdinti kaip geometrinio objekto plokštumos atskyrimą. Briauna apribojanti geometrinio objekto medžiagos paviršius vadinama plyšio kraštu. Yra trys pagrindiniai deformacijos formos tipai, apibūdinantys viršutinės ir apatinės plyšio paviršiaus persislinkimo atvejus.



Pav. 2.8 Plyšio plitimas geometrinuose objektuose: 1 – plyšio plitimas y ašimi; 2 – plyšio plitimas x ašimi; 3 – plyšio plitimas z ašimi.

Pirmojo plyšimo plitimo formoje konstrukcinių elementų paviršiaus perskiriamas ir juda y ašies atžvilgiu. Antrojo plyšio plitimo atveju jėga pereina į gniuždymo/ tempimo deformaciją. Trečiuoju atveju plyšimas atsiranda z ašies atžvilgiu.

Skirtingiems plyšio plitimo formoms bei deformacijoms aprašyti taikomas skirtingos matematinės įtempimo išraiškos.

Skaičiuojant geometrinių objektų, turinčių plyšių, konstrukcijos elementų stiprumą, svarbu nustatyti įtempimų ir deformacijų būseną, veikiant išoriniams apkrovos jėgoms ir atkreipiant didesnę dėmesį į plyšio viršūnes, nes joje prasideda irimas.

Pirmojo plyšio plitimo tipo apibendrinantis sprendinys gali būti išreikštas formule:

$$\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} f_{\alpha j}(\theta) + T \delta_{1i} \delta_{1j} + C_{ij2}(\theta) r^{\frac{1}{2}} + \dots + C_{ijn}(\theta) r^{\frac{n-1}{2}} + \dots$$

Deformacijoje ir poslinkyje naudojama išraiška:

$$\varepsilon_{ij} = \frac{K_I}{E\sqrt{2\pi r}} f_{\alpha j}(\theta) + \dots$$

Čia r ir θ – polinės koordinatės. $r^{-1/2}$ - asimptotinis sprendinys, kai r artėja į 0. K – įtempimo intensyvumo koeficientas. Šis įtempimo intensyvumo koeficientas priklauso nuo kūno su plyšiu matmenų, plyšio dydžio ir deformacijos tipo, taip pat nuo apkrovos dydžio, jo dimensijos $\text{MPa}\sqrt{m}$. T 0 pastovaus dydžio rodiklio, kurio dimensija atitinka įtempimus. Jis priklauso nuo apkrovos atvejo. Aukštesnio dydžius įprasta atmesti.

Jau irimo mechanikos pradžioje buvo kuriami įvairūs analitiniai metodai skirti irimo parametrų nustatymui bei pačio irimo procesui nagrinėjimui. Šie metodai skirti tradicinių formų objektų defektams nagrinėti. Tačiau realiame gyvenime susiduriame su įvairiais defektais, kurių skaičiavimams nėra analitinių sprendinių, todėl tenka ieškoti kitų sprendimo kelių.

Mechanikos irimo uždaviniams spręsti pritaikomus skaitinius metodus galima nagrinėti keliais aspektais. Analizuojant įtempimų bei deformacijų būsenų analizės uždavinius galima išnagrinėti sudėtingesnių geometrinių objekto sričių vietas su įvairiausių formų defektais. Įvertinti įvairias veikiančias apkrovos jėgas ir kraštinių sąlygas. Tuo tarpu teoriniai analitiniai medeliai taikytini tik ribotiems atvejams. Skaitiniai metodai leidžia tiksliau išanalizuoti įtempimų ir deformacijų būsenas defektų aplinkoje.

Aktualiausia skaitinių metodų taikymo sritis yra irimo proceso modeliavimas. Šis modeliavimas leidžia sekti plyšių plitimo eigą, jo trajektoriją. Eksperimentiniuose metoduose šių procesų stebėjimas yra kur kas sudėtingesnė procedūra. Plyšių plitimas yra netiesinis procesas. Kinta plyšio judėjimo trajektorija, kraštinės sąlygos ir panašiai. Skaitiniai metodai leidžia sąlygiškai nebrangiai sumodeliuoti irimo, deformacijos mechanikos uždavinius, taikant skaitinius metodų eksperimentus.

Vienas iš populiariausių ir aktualiausių skaitinių metodų taikomų irimo bei deformacijos procesui modeliuoti yra baigtinių elementų metodas(BEM). Be šio metodo galima taikyti ir spyruoklės/masės(angl. Spring/mass), masės tempimo(angl. Mass-tensor), deformuojamos juostos(angl. Deformable splines) bei kitus metodus.

3. BAIGTINIŲ ELEMENTŲ METODAS

Geometrinių objektų struktūra ir jų atliekamos deformacijos elgsenos išreiškimas funkcijomis yra dažniausiai gan sudėtingas procesas, kurį būtų galima tiksliai matematiškai aprašyti. Todėl skaičiavimams naudojamos ekvivalentinės skaičiuojamosios schemos arba tikslūs ar idealizuoti modeliai. Pasirinkus paprastą, tačiau pakankamai tikslų modelį, mažesnės bus skaičiavimo sąnaudos, lengviau bus spręsti optimizavimo uždavinius. Eksperimentiniu metu vėliau nustatoma ar pasirinkto modelio ir jo atlikti skaičiavimo rezultatai yra priimtini.

Ištobulėjus kompiuteriniams, skaičiavimo metodams, skaičiavimams jau galima panaudoti realios konstrukcijos brėžinį – kompiuterine grafika parengtą geometrinį modelį, nekuriant atskirai specialaus, tik dinaminiam skaičiavimams sudaryto idealizuoto modelio.

Kompiuterinis modeliavimas, kurio programų yra nemažai, tapo pagrindine įvairių mokslo ir inžinierinių sričių dalimi. Todėl dauguma mokslininkų ir inžinierių naudoja įvairias programas, pradedant nuo bazinių programavimo kalbų iki aukšto lygio paketų, naudojančių pažangius modeliavimo metodus. Viena iš baigtinių elementų metodo naudojimo sričių yra kompiuterinis modeliavimas. Juo paremtos baigtinių elementų programos yra neatsiejama inžinierinio projektavimo ir analizės dalis, palengvinanti naujų gaminių kūrimą ir projektavimą. Plėtojantis kompiuterinėms technologijoms, šios programos vis plačiau naudojamos modeliuojant. Iš visų projektavimo metodų, plačiausiai yra naudojamas baigtinių elementų metodas. Jis yra labiau ištobulintas ir pranašesnis už tokius metodus, kaip antrinių šaltinių metodas, baigtinių skirtumų metodas ar momentų metodas.

Vienas iš labiausiai paplitusių skaitinių metodų, skirtas deformacijos skaičiavimams atlikti ar mechaniniams konstrukcijoms modeliuoti, yra baigtinių elementų metodas (BEM). Tai skaitinis metodas, leidžiantis rasti apytiksles diferencialinių lygčių dalines išvestines ar integralinių lygčių sprendinius. Jis yra bene plačiausiai šiandienai taikomas metodas. BEM populiarumą lėmė jo tinkamumas įvairioms fizikos sritims. Jis yra laikomas vienu pažangiausių inžinierinių metodų.

BEM plačiausiai taikomas daugelio specialybių inžinierių skaičiuojamosios mechanikos metodas. Metodo idėja kilo skaičiuojant inžinerines konstrukcijas ir yra labai artima deformuojamų diskrečiųjų struktūrų mechanikai. Intuityvus diskretizavimas yra natūralus būdas, daugelį metų taikomas inžinierių, sudėtingiems objektams modeliuoti.

Matematinio požiūriu BEM yra apytikslinis metodas, skirtas diferencialinėms dalinių išvestinių lygtims spręsti (inžinieriškai skambantis – kontinualiosios struktūroms modeliuoti). Kontinuumo tapatinimas su diskrečiąja struktūra yra tik matematinės abstrakcijos rezultatas. Griežtų matematinių teiginių vienovė ir fizikinis bei inžinierinis metodo interpretavimas yra labai svarbūs ypatumai, lėmę jo paplitimą įvairiuose mokslo ir technikos srityse.

Šiuolaikinė BEM samprata labai plati. Šis metodas dažniau suvokiamas kaip skaičiavimo technologija ar universalus fizikinių ir inžinierinių objektų ar sistemų elgsenos analizės būdas. Pagal formaliąsias savybes BEM tampa šablonu, kuriame vyrauja formalūs euristiniai principai ir loginiai ryšiai, o matematinės priklausomybės tik nusako kiekybinius ryšius.

BEM algoritmas tinka spręsti daugelį sistemos uždavinių, kurie fizikiniu požiūriu yra skirtingi. Todėl nebereikia gilintis į daugelį tradicinių analizinių lygčių sprendimo metodų. Tai įtakojo BEM metodo paplitimą ir skysčių mechanikoje, jį plačiai taiko informatikai, matematikai, tiek ir fizikai.

Ir vis dėlto BEM realią vertę įgyja tik sukūrus atitinkamas jį realizuojančias kompiuterines programas ar mobiliąsias programėles. Programuoti reikia ne tik matematinius veiksmus, pravartu turėti baigtinių elementų tinklelio generavimo ir rezultatų apdorojimo bei vizualizavimo ir kitas programas. Mat sprendinio prasmę dažniausiai galima suvokti tik atvaizdavus jį grafiškai ekrane. Todėl BEM pagrindu yra sukurta daug universalių bei specializuotų kompiuterinių programų. Pasaulyje lyderiaujančias pozicijas užima šie universalūs BEM programiniai paketai: ABAQUS, ALGOR, ANSYS, COSMOSM, NASTRAN, ir kt.

Patį metodą patogiau suvokti ne kaip abstrakčiai matematinį, o susieti jį su konkrečių taikomųjų uždavinių klasėmis ir jų ypatumais. BEM algoritmas panašus įvairiuose fizinių sistemų uždaviniuose.

Teoriniu požiūriu BEM yra analogiškas kitiems skaitiniams metodams, tačiau, taikant techninėms problemų sritims spręsti, metodas pasižymi tam tikrais privalumais.

Baigtinių elementų metodo privalumai:

1. padeda projektuojant sudėtingus geometrinius objektus;
2. atliekant modelio skaičiavimus, galima koreguoti, tikslinti ir greitai atnaujinti skaičiavimo rezultatus;
3. plačiai taikomas įlinkių deformacijos skaičiavimui, dėl mažesnių laiko resursų sąnaudų, deformacijos procesams apskaičiuoti;
4. tinkamas neįprastoms apkrovimo situacijoms (pvz. staigus geometrinių objektų susidūrimas) generuoti;
5. leidžia sumodeliuoti objektų elgseną dideliame situacijų spektre.

Baigtinių elementų metodo trūkumai:

1. sunku įvertinti momentų pasiskirstymą;
2. reikalauja gerų modeliavimo žinių;
3. tikrinant sunku modelyje atrasti klaidų;
4. reikalingos tiek projektavimo žinios, tiek ir supratimas apie tinklo pasiskirstymą elementuose.

Kaip jau minėta, baigtinių elementų metodas yra skaitmeninis metodas, skirtas inžinieriniams projektavimui ir analizei. BEM metodo taikymas geometrinio objekto modeliavimui ir gaunamų rezultatų analizei daugiausiai priklauso nuo naudojamų skaičiavimo ir modeliavimo procedūrų. Nepriklausomai nuo metodo taikymo srities, skaitmeninis modeliavimas reikalauja išsamios informacijos apie nagrinėjamą erdvę ir jos elementus (geometriją, kraštines, ribines sąlygas, poveikius ir t.t.).

Baigtinių elementų metodo esmė – sritis, kurioje ieškomas sprendimas, yra sudalinama į baigtinius elementus (paprasčiausias geometrinės figūras). Uždaviniams spręsti dažniausiai yra naudojami trikampiai ir keturkampiai elementai. Baigtiniai elementai tarpusavyje yra sujungti mazguose. Elementų visuma sudaro baigtinių elementų tinklą. Šių elementų viduje yra užduodamas nežinomos funkcijos (poslinkių) kitimo dėsnis. Bet kurio elemento e taško poslinkiai aprašomi vektoriumi $u_e(x)$, gali būti išreikšti per mazginių taškų poslinkius u_e ir aproksimacijos matricą $[N_e(x)]$:

$$u_e(x) = [N_e(x)]u_e$$

Matricos $[N_e(x)]$ elementai sudaryti iš paprasčiausių išraiškų, kaip taisyklė, iš laipsninių polinomų. Polinomo eilė yra siejama su elemento mazgų skaičiumi. Jei elementas turi tik kampinius mazgus, tai $N_e(x)$ – sudaryta iš tiesinių funkcijų.

Remiantis poslinkių, o taip pat ir kitų dydžių elemente dėsniniais, sudaromos lygtys visai baigtinių elementų sistemai. Išskiriami šie skaičiavimo etapai:

1. atskirų elementų informacijos suformavimas;
2. atskirų elementų pagrindinių priklausomybių suformavimas;
3. atskirų elementų surinkimas į vieną modelį ir šio modelio sprendimas, nustatant pagrindinius nežinomuosius;
4. kitų dydžių apskaičiavimas pavieniuose elementuose.

Terminų išreikštinė (angl. explicit) arba neišreikštinė (angl. implicit) analizė nusako būdą konstrukcijos atsakui į dinaminis arba statinius poveikius apskaičiuoti.

Taikant išreikštinį būdą, randami sistemos pagreičiai ir pagal juos sprendžiama, kaip geometrinio objekto konstrukcijos mazgai pasislinks per labai trumpą laiką. Geometrinio objekto konstrukcijos mazgams pasislinkus, pagreičiai apskaičiuojami iš naujo, įvertinant pakitusius konstrukcijos poslinkius ir greičius, numatoma poslinkių reikšmės sekančio trumpo laiko intervalo bėgyje, ir procesas kartojamas. Dažniausiai nereikia spręsti lygčių sistemų, o vis naujos poslinkių reikmės apskaičiuojamos pagal rekurentines formules. Visi skaičiavimai remiasi užprogramuotomis analitinėmis išraiškomis ir nereikia atlikti sudėtingų lygčių sistemų sprendimo.

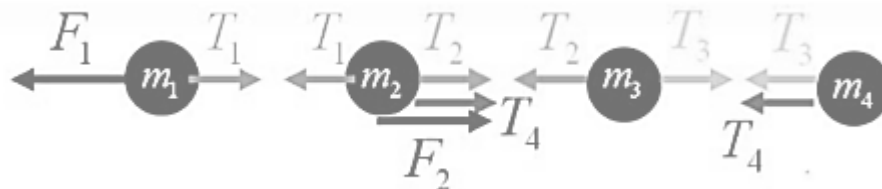
Taikant neišreikštinį būdą, pradžioje apskaičiuojama sistemos pusiausvyros padėtis praėjus tam tikram laiko intervalui nuo poveikio momento. Laiko intervalas gali būti didesnis. Taikant šį būdą visuomet renka spręsti algebrinių lygčių sistemą. Didelės lygčių sistemos sprendinį sudėtinga užrašyti analitinėmis išraiškomis, todėl taikomos lygčių sistemų skaitinio sprendimo programos arba funkcijos.

Geometrinio objekto deformacijos proceso analizę geriausiai atspindi spyruoklių konstrukcijos pavyzdys.

Geometrinio objekto konstrukcine struktūrą reikia suskaidyti į elementus taip, kad visi jie būtų aprašomi vienodo pavidalo lygtimis. Šiuo atveju konstrukcijos elementas yra spyruoklė. Elementu sujungimo taškai vadinami mazgais, kur kiekvienas jų turi sutelktąją masę. Kiekvieno mazgo numeris konstrukcijoje vadinamas globaliuoju numeriu. Nagrinėjant

vieną atskirą elementą, taikyti globaliuosius numerius netikslinga, nes šiuos atveju pageidautina gauti kiekvienam nagrinėjamojo tipo elementui bendras išraiškas. Išvedant pavienio elemento pusiausvyros lygtį, mazgams numeruoti laikomi lokalieji numeriai, kurie dažniausiai nesutampa su globaliaisiais.

Sudarant i-tojo elemento lygtis, pirmiausia reikia užrašyti pusiausvyros lygtis kiekvienam elemento mazgui. Pirmąjį mazgą veikia dvi jėgos: R_1^i , kur nagrinėjamąjį elementą veikia greta esantys elementai bei deformuotos spyruoklės tamprumo jėga $T_i = k_i(u_2^i - u_1^i)$, čia k_i – spyruoklės standumo koeficientas. Pagal antrą Niutono dėsnį, mazgo pusiausvyra galima, kai šių jėgų projekcijų į ašį suma lygi mazgo masės ir pagreičio sandaugai, t.y. $T_i + R_1^i = m_1^i a_1^i$. Užrašant 2-ojo mazgo pusiausvyros lygtį, reikia įvertinti, kad spyruoklės tamprumo jėgos projekcija yra priešingo ženklo, nes ištempta spyruoklė trauktų 2-ąjį mazgą.



Pav. 3.1 Konstrukcijos elemento mazgus veikiančios jėgos

Pav.2.1 pavaizduotos visų spyruoklių sukurtos tamprumo jėgos, veikiančios visus konstrukcijos mazgus. Užrašę kiekvienam mazgui 2-ąjį Niutono dėsnį gausime:

$$\begin{cases} m_1 a_1 = -F_1 + T_1 \\ m_2 a_2 = F_2 - T_1 + T_2 + T_4 \\ m_3 a_3 = -T_2 + T_3 \\ m_4 a_4 = -T_3 - T_4 \end{cases}$$

Čia $T_1 = k_1(u_2 - u_1); T_2 = k_2(u_3 - u_2); T_3 = k_3(u_4 - u_2); T_4 = k_4(u_4 - u_2)$;

Bendrojo pavidalo konstrukcijai kiekvienam i-am mazgui užrašomas dinamikos lygties išraiška:

$$m_i a_i = \sum_j T_j(u_1^j, u_2^j) + F_i, \quad i = 1, \dots, N;$$

Čia N-mazgų skaičius, F_i – i-ame mazge veikiančių jėgų atstojamoji, o pagal indeksą, j sumuojamos visų i-ame mazge sujungtų elementų sukurtos vidinės jėgos. Kiekvieno elemento sukurtos vidinės jėgos priklauso tik nuo jam priklausančių mazgų poslinkių.

Jeigu lygiagrečiai su tampriaisiais elementais konstrukcijoje naudojami slopintuvai(jėga proporcinga jų galų greičio skirtumui) apskaičiuojamos pagal formules:

$$T_1 = k_1(u_2 - u_1) + c_1(u_2 - u_1);$$

$$T_2 = k_2(u_3 - u_2) + c_2(u_3 - u_2);$$

$$T_3 = k_3(u_4 - u_2) + c_3(u_4 - u_3);$$

$$T_4 = k_4(u_4 - u_2) + c_4(u_4 - u_2);$$

Čia c_i - i-os spyruoklės klampiosios trinties koeficientas.

Matricinės sistemos išraiška užrašoma taip:

$$\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & m_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{Bmatrix} T_1 \\ -T_1 + T_2 + T_4 \\ -T_2 + T_3 \\ -T_3 - T_4 \end{Bmatrix} + \begin{Bmatrix} -F \\ F_2 \\ 0 \\ 0 \end{Bmatrix}$$

Bendruoju atveju išreiškiama pavidalu:

$$[M]\{\ddot{U}\} = \{T(U, \dot{U})\} + \{F(t)\},$$

arba

$$[M]\{\ddot{U}\} = \{f(U, \dot{U}, t)\}.$$

Tai yra paprastoji diferencialinė lygtis laiko atžvilgiu. Žinomomis laikome šios lygties pradinės sąlygas pavidalu:

$$\{U(0)\} = \{U\}_0; \{\dot{U}(0)\} = \{V\}_0.$$

Paprastoji diferencialinė lygtis su pradinėmis sąlygomis gali būti sprendžiama taikant įvairius skaitinio integravimo algoritmus.

Esant tiesiniam uždaviniui vieno elemento e būvio lygtis užrašoma:

$$[K_e]u_e = F_e.$$

Čia K_e – elemento standumo matrica, F_e – elemento apkrovos vektorius.

Pasinaudoję vieną iš žinomų metodų, elemento standumo matricią išreiškiame tokiu būdu:

$$[K_e] = \int_{V_e} [B_e]^T [D_e] [B_e] dV$$

$[B_e]$ – elemento geometrinė matrica, gaunama diferencijavus formos funkcijas.

$[D_e]$ – fizinių konstantų modelio matrica.

V_e – elemento tūris.

Iš elemento lygčių yra formuojama būvio lygtis:

$$[K]u = F$$

Esant tiesiniam uždaviniui, šis modelis yra tiesinė lygčių sistema. Išsprendus būvio lygtį ir suradus poslinkius u įtempimai elemente yra skaičiuojami:

$$\sigma_e = [\beta_e]u_e.$$

$$[\beta_e]\text{-sudaroma iš } [D_e] \text{ ir } [B_e] \text{ panaudojant } u_e(x) = [N_e(x)]u_e.$$

Įtempimų pasiskirstymą bet kuriame elemente galima nustatyti remiantis mazginių poslinkių ir dėsnų rezultatais.

Netiesiniai modeliai

Norint tiksliau aprašyti deformuojamo kūno elgseną, tiesiniai modeliai to neužtikrina. Siekiant didelio tikslumo naudojama modernizuota Lagranžo formuluotė. Sprendžiant netiesinius uždavinius keliamas tikslas nustatyti tikrąsias būvio kintamųjų reikšmės aktualioje konfigūracijoje atitinkančioje laiko momentą $t + \Delta t$. Formuluojant diskretinį modelį virtualiųjų darbų lygybės principu, reikia užrašyti šį principą kontinualiam kūnui aktualioje konfigūracijoje:

$$\int_{t+\Delta t V} {}^{t+\Delta t}\sigma_{ij} * \delta {}^{t+\Delta t}\varepsilon_{ij}^L {}^{t+\Delta t}dV = \delta {}^{t+\Delta t}W_e.$$

Čia ${}^{t+\Delta t}\sigma_{ij}$ yra Košy įtempimų tenzorius ir ${}^{t+\Delta t}\varepsilon_{ij}^L$ yra Almazni deformacijos, $\delta {}^{t+\Delta t}W_e$ yra išorinių jėgų virtualusis darbas.

Geometrinio objekto kūnas patiria dideles deformacijas ir poslinkius, todėl atsiranda nežinoma aktuali konfigūracija, kurios priklausomybė negali būti tiesiogiai išspręsta. Keliant prielaidas gali būti randamas apytikslinis sprendinys, kurio visi kintamieji remiasi žinoma ataskaitos konfigūracija laike t . Priklausomybė sprendimui naudojama modernizuota Lagranžo formuluotė. Tai praktinis formuluotės realizavimo būdas, kai virtualiųjų poslinkių principas išreiškiamas per antruosius įtempimus ${}^{t+\Delta t}{}_tS_{ij}$ ir deformacijas ${}^{t+\Delta t}e_{ij}$ užrašytas atskaitos konfigūracijos t atžvilgiu:

$$\int_{tV} {}^{t+\Delta t}{}_tS_{ij} * \delta {}^{t+\Delta t}e_{ij} {}^t dV = \delta {}^{t+\Delta t}W_e$$

Po lygties linearizavimo gaunama priklausomybė:

$$\int_{tV} {}^tD_{ijrs}^{ep} * \varepsilon_{rs}^L * \delta {}^t\varepsilon_{ij}^L {}^t dV + \int_{tV} {}^t\sigma_{ij} * \delta {}^t\varepsilon_{ij}^{NL} {}^t dV = \delta {}^{t+\Delta t}W_e - \int_{tV} {}^t\sigma_{ij} * \delta {}^t\varepsilon_{ij}^L {}^t dV \quad (2.21)$$

čia ${}^t\sigma_{ij}$ žinomas įtempimas laike t , ${}^t\varepsilon_{ij}^L$ ir ${}^t\varepsilon_{ij}^{NL}$ yra tiesinis ir netiesinis deformacijos komponentas ${}^te_{ij} = {}^t\varepsilon_{ij}^L + {}^t\varepsilon_{ij}^{NL}$ atskaitos konfigūracijos t atžvilgiu; ${}^tD_{ijrs}^{ep}$ yra tangentinis medžiagos savybių tenzorius t konfigūracijoje.

Lygties diskretizacijai naudojamas BEM. Baigtinių elementų (toliau BE) pusiausvyros lygti galima užrašyti:

$$([{}^tK_L] + [{}^tK_{NL}])\Delta U = \Delta F$$

Čia tK_L ir ${}^tK_{NL}$ yra globalios tiesinė ir geometrinė standumo matricos. ΔF - apkrovų vektorius atspindintis tūrinių jėgų, paviršinių jėgų ir pradinių įtempimų poveikį. Panaudojant formos funkcijas poslinkiams ir lygtį (2.21) yra išvedamos standumo matricos kiekvienam elementui. Tuomet jos yra sudedamos į globalias standumo matricas $[{}^tK_L]$ ir $[{}^tK_{NL}]$.

Tiesinė standumo matrica elementui $[{}^tK_L^e]$ yra gaunama iš pirmojo lygties (2.21) nario. Iš geometrinių tiesių ši matrica atspindi fizinį netiesiškumą ir užrašomas kaip:

$$[{}^tK_L^e] = \int_{tV} [{}^tB_L^e]^T [{}^tD^{ep}] [{}^tB_L^e] dV$$

Čia $[{}^tD^{ep}]$ – fizinių konstantų tamprumo matrica ir $[{}^tB_L^e]$ yra tiesinė deformacijų aproksimacinė matrica. Geometrinė standumo matrica $[{}^tK_{NL}^e]$ yra gaunama iš antrojo lygties (2.21) nario:

$$[{}^tK_{NL}^e] = \int_{tV} [{}^tB_{NL}^e]^T [{}^t\sigma] [{}^tB_{NL}^e] dV,$$

Čia $[{}^tB_{NL}^e]$ yra netiesinė deformacijų aproksimacinė matrica, $[{}^t\sigma]$ – įtempimo matrica.

Sudarant baigtinių elementų tinklo struktūrą yra svarbu baigtinių elementų modelio sprendimas. BE sudarymas yra iteracinis procesas. Pirmos iteracijos metu iš pirminio elementų tinklo sudaromas pirmasis sprendinys atspindintis pirmos iteracijos rezultatus. Vėliau iteracija yra tęsiama tol, kol yra gaunamas ieškomas tikslumas. Siekiant sėkmingai panaudoti BE tinklą, svarbu turėti pakankamai kokybišką pradinį geometrinio objekto BE tinklą, kuriame įvertinta konstrukcijos geometrija, apkrovos bei medžiagos pasiskirstymo duomenys.

Kiekvienas vidinis mazgas, esantis struktūriniame baigtinių elementų tinkle, yra apsuptas vienodo elementų skaičiaus. Taip sukuriama taisyklingos formos elementai, tačiau jų taikymas geometriniam objektams yra gan sudėtingas. Šie struktūriniai tinklai generuojami algebriniais ir lygčių metodais. Algebrinių geometrinių struktūrų generavimo metoduose tinklo tankumą reguliuoja mazgų išdėstymas srities kraštus sudarančiose kreivėse ir tam tikra svorio koeficientų funkcija. Sutankinus baigtinių elementų tinklą tam tikroje vienoje srityje, neleistinai pablogėja elementų, esančių kitose apibrėžimo srities dalyse, geometriniai rodikliai.

Baigtinių elementų tinkle dažniausiai naudojami nestruktūriniai BE tinklai. Nestruktūrinis tinklas – tai tinklas, kuriame elementų skaičius supantis elemento mazgą kinta

nagrinėjamoje srityje viduje. Nestruktūriniai tinklų elementų dydis gali keistis skaičiuojamosios srityje ribose, priklausomai nuo sprendžiamos problemos ypatumų. Tinklo tankumas yra apibrėžiamas tinklo tankio funkcija. Jei tankio funkcija yra pakankamai glodi, garantuojamas geros elementų geometrinės charakteristikos, nepaisant didelio baigtinių elementų dydžio skirtumo. Nestruktūriniai tinklams generuoti vis populiarėja mišrūs nestruktūrinio baigtinių elemento tinklo generavimo metodai, apjungiantys Delaunay ir plintančio fronto metodų pranašumus ir išvengia dalies trūkumų. Nors šie metodai skiriasi vienas nuo kito, tačiau visiems metodams bendra tai, kad tinklas turi turėti galimybę būti pritaikomas tam tikrai problemai spręsti. Šie metodai inžinerijoje vis labiau naudojami, nes sprendžiant daugelį inžinerinių problemų reikia, kad tinklas būtų tankesnis tik tam tikrose būdingose vietose, o kitur pakaktų ir retesnio.

Įprastai baigtinių elementų tinklų generavimo strategijos pradedamos nuo palyginti reto ir tolygaus tankumo tinklo. Atliekant sudėtingų inžinerinių sistemų analizę, reikia atlikti nemažai iteracijų tol, kol sukūrimas optimalus baigtinių elementų tinklas, kuriame aproksimacijos paklaida yra mažesnė už leistiną ir yra tolygiai pasiskirsčiusi visame inžinerinės sistemos modelyje.

Baigtinių elementų tinklų generavimas yra vienas svarbiausių analizės etapų. Paklaidų įvertys arba indikatoriai nurodo uždavinio apibrėžimo srityje dalis, kuriose baigtinio elementų tinklas turi būti tankesnis. Todėl, reikia sugeneruoti baigtinių elementų tinklą, kurio tankumas atitiktų paklaidos įverčio reikšmių pasiskirstymą uždavinio apibrėžimo srityje.

Pagrindinis dėmesys matematikoje skiriamas paklaidos įverčiams, išvedamiems prieš atliekant skaičiavimus ir suteikiantiems informaciją tik apie asimptotinį sprendinio konvergavimo greitį. Šios informacijos neįmanoma paversti konkrečiais skaičiais ir įvertinti skaitinio sprendinio paklaidos. Klasikinėje skaičiuojamosios mechanikos praktikoje sprendinio paklaida įvertinama remiantis euristiniais argumentais, kurie priklauso nuo mokslininko patirties ir kvalifikacijos. Skaitinio sprendinio tyrimas, kelis kartus tankinant baigtinių elementų tinklą visoje uždavinio apibrėžimo srityje, reikalauja didelių kompiuterinių resursų ir gali būti taikomas tik nedideliems uždaviniams. Analizė suteikia tikslesnę informaciją apie skaitinio sprendinio paklaidą ir naudoja ribotą resursų kiekį. Paklaidų įverčių reikšmės apskaičiuojamos po tam tikro skaičiavimų ciklo. Jie disponuoja konkrečiomis skaitinio sprendinio reikšmėmis, todėl gali tiksliau įvertinti jo paklaidą. Nustatomos apibrėžimo srityje dalys, kuriose paklaida yra didžiausia, ir tankinamas baigtinių elementų tinklas.

4. PROJEKTINĖ DALIS

Šiame skyriuje pateikiami esminiai kuriamos programinės įrangos techninės-projektinės dokumentacijos aspektai.

4.1. Projektavimo metodika

Architektūros projektavimas bus atliekamas naudojantis MagicDraw programine įranga, kurios pagalba bus sukurta projektinė dokumentacija naudojant Unified Modeling Language (UML) notaciją. Ja remiantis atsiras galimybė pakartotinai panaudoti projekte surastus problemų sprendimus kituose projektuose.

Sistema bus projektuojama remiantis evoliucinio metodo principais. Taikant šį metodą su užsakovais dirbama nuo pradinės specifikacijos iki galutinės programų sistemos.

Kuriama sistema pasižymės modulinio karkasu. Tai leis sistemą praplėsti naujomis funkcijomis ar lengvai atnaujinti. Tai taip palengvinti naujų produktų kūrimą, panaudojant jau sukurtus modulius ir taip sutaupant laiką. Kuriama sistema leis panaudoti objektų deformaciją kitų žanrų žaidimuose.

4.2. Architektūriniai reikalavimai

Architektūrinius sprendimus įtakojuojantys reikalavimai:

- sistema turi būti suprojektuota taip, kad ją galima būtų lengva išplėsti ar prijungti naujus modulius;
- kuriama sistema bus pateikta kaip atviro kodo, nekomercinė programinė įranga;
- sistema neturi leisti neautorizuotiems vartotojams prie jos prisijungti.

Sudarant sistemos architektūrą, turi būti atsižvelgta į būtinas programos vykdymo charakteristikas, apibrėžtas reikalavimų specifikacijoje

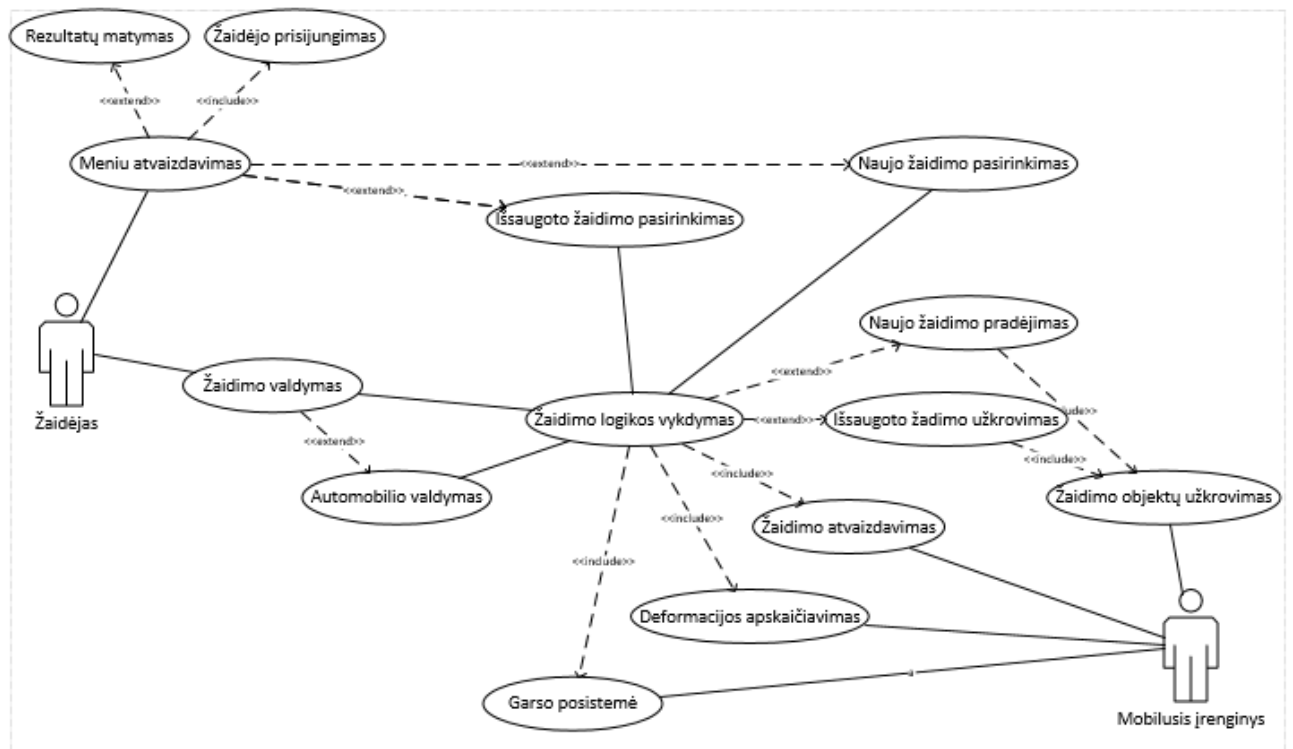
4.3. Architektūros pateikimas

Dokumente sistemos architektūra pateikiama keliais vaizdais: panaudojimo atvejų (PA), statiniu, dinaminiu ir išdėstymo. Šie vaizdai yra pateikiami kaip Rational Rose modeliai, naudojant unifikuotą modeliavimo kalbą (UML). Sistemos architektūra pateikta remiantis RUP (Rational Unified Process) rekomendacijomis. Sistemos specifikacija pateikta vaizdais, kuriems įgyvendinti reikia UML diagramų:

1. panaudojimo atvejų vaizdas (panaudojimo atvejų diagrama);
2. sistemos statinis vaizdas (paketai ir klasių diagramos);

3. sistemos dinaminis vaizdas (būsenų, veiklos, sekų, bendradarbiavimo diagramos);
4. išdėstymo vaizdas (išdėstymo diagrama).

4.4. Panaudos atvejų vaizdas



Pav. 4.1 Automobilio sudaužymo žaidimo panaudos atveju diagrama

Panaudojimo atvejis 1: „Žaidėjo prisijungimas“

Aktorius: žaidėjas.

Tikslas: perduoti programai vartotojo prisijungimo duomenis.

Ryšiai su kitais PA: iššaukiamas suaktyvavus kitus PA, norint nustatyti, ar vartotojas turi teisę į pasirinktą PA.

Prieš-sąlygos: patikrinama, ar duomenų bazė veikia ir yra pasiekiami programos.

Sužadinimo sąlyga: sistemos vartotojas paleidžia PĮ vykdomąją rinkmeną.

Po-sąlyga: vartotojas autorizuotas ir įgavo teises naudotis programos paslaugomis.

Pagrindinis scenarijus:

1. sistema paprašo įvesti vartotojo prisijungimo duomenis;
2. patikrinama, ar toks vartotojas egzistuoja duomenų bazėje ir jam suteikiamos atitinkamos teisės.

Alternatyvūs scenarijai:

- vartotojas įvedė neteisingus prisijungimo duomenis;
- vartotojas nusprendė baigti darbą dar neprisijungus prie programos.

Panaudojimo atvejis 2: „Rezultatų matymas“

Aktorius: žaidėjas.

Tikslas: parodyti vartotojui daugiausiai surinkusių taškų vartotojus.

Ryšiai su kitais PA: vartotojas turi būti sėkmingai identifikuotas žaidėjo prisijungimo PA.

Prieš-sąlygos: duomenų bazė veikia ir yra pasiekama programos.

Sužadinimo sąlyga: sistemos vartotojas meniu punkte pasirenka punktą, leidžiantį rodyti daugiausiai taškų surinkusius žaidėjus.

Po-sąlyga: vartotojas mato daugiausiai taškų surinkusių vartotojų sąrašą su atitinkama informacija.

Pagrindinis scenarijus:

1. identifikuotas vartotojas meniu sąrašė paspaudžia nuorodą, kad būtų rodomi rezultatai;
2. vartotojui pateikiamas daugiausiai surinkusių taškų vartotojų sąrašas su jų surinktais taškais.

Alternatyvūs scenarijai:

1. vartotojui rodomas tuščias rezultatų sąrašas;
2. vartotojui rodomas programos apribotas rezultatų kiekis.

Panaudojimo atvejis 3: „Meniu atvaizdavimas“

Aktorius: žaidėjas.

Tikslas: identifikuotas žaidėjas inicializuoja kitus PA, pasirinkdamas atitinkamą meniu punktą.

Ryšiai su kitais PA: vartotojas turi būti sėkmingai identifikuotas kaip žaidėjas, prisijungęs PA. Vartotojas gali inicializuoti „Rezultatų matymo“, „Naujo žaidimo pasirinkimas“, „Išsaugoto žaidimo pasirinkimo“ PA.

Prieš-sąlygos: vartotojas sėkmingai identifikuotas „Žaidėjo prisijungimo“ PA.

Sužadinimo sąlyga: žaidėjo PA sėkmingo prisijungimo metu sužadinamas „Meniu“ atvaizdavimas.

Po-sąlyga: vartotojas mato meniu, kuris inicializuoja atitinkamus PA.

Pagrindinis scenarijus:

1. vartotojas sėkmingai identifikuoja save programoje;
2. žaidėjo prisijungimo PA inicializuoja „Meniu“ atvaizdavimą.

Alternatyvūs scenarijai:

1. sukuriamas naujas žaidėjas programoje;
2. naujas žaidėjas identifikuojamas programoje.

Panaudojimo atvejis 4: „Naujo žaidimo pasirinkimas“

Aktorius: žaidėjas.

Tikslas: inicializuoti duomenis, numatytus pagal nutylėjimą, reikalingus kitiems PA.

Ryšiai su kitais PA: naujo žaidimo pasirinkimo PA praplečia „Meniu atvaizdavimas“; PA „Naujo žaidimo pasirinkimas“; PA inicializuoja „Žaidimo logikos vykdymą“.

Prieš-sąlygos: žaidėjas „Meniu atvaizdavime“ gali pasirinkti meniu punktą, inicializuojantis šį PA.

Sužadinimo sąlyga: žaidėjas PA „Meniu atvaizdavimas“ pasirenka meniu punktą, inicializuojantį „Naujo žaidimo pasirinkimas“ PA.

Po-sąlyga: šis PA perduoda pradinis duomenis reikalingus inicializuoti „Žaidimo logikos vykdymas“ PA.

Pagrindinis scenarijus:

1. žaidėjas meniu elementų sąrašė pasirenka punktą, nurodantį naujo žaidimo pasirinkimą.

Alternatyvūs scenarijai:

1. žaidėjas nurodo reikalingus specifinius parametrus ir tik tuomet inicializuoja naują žaidimą.

Panaudojimo atvejis 5: „Išsaugoto žaidimo pasirinkimas“

Aktorius: žaidėjas.

Tikslas: inicializuoti duomenis, išsaugotus žaidimo metu, reikalingus kitiems PA.

Ryšiai su kitais PA: išsaugoto žaidimo pasirinkimo PA praplečia „Meniu atvaizdavimas“ PA „Išsaugoto žaidimo pasirinkimas“ PA inicializuoja „Žaidimo logikos vykdymo“ PA.

Prieš-sąlygos: žaidėjas „Menu atvaizdavimas“ PA gali pasirinkti meniu punktą, inicializuojantis šį PA.

Sužadinimo sąlyga: žaidėjas „Menu atvaizdavimas“ PA pasirenka meniu punktą inicializuojantis „Išsaugoto žaidimo pasirinkimas“ PA.

Po-sąlyga: šis PA perduoda pradinis duomenis, reikalingus inicializuoti „Žaidimo logikos vykdymas“ PA.

Pagrindinis scenarijus:

1. žaidėjas meniu elementų sąrašė pasirenka punktą, nurodantis „Išsaugoto žaidimo pasirinkimą“;
2. vartotojas pasirenka išsaugoto žaidimo objektą.

Alternatyvūs scenarijai:

1. Žaidėjas.

Panaudojimo atvejis 6: „Žaidimo logikos vykdymas“

Aktorius: mobilusis įrenginys.

Tikslas: kontroliuoti viso žaidimo logiką bei sąryšį su kitais PA.

Ryšiai su kitais PA: „Žaidimo valdymas“ PA gali stabdyti šį PA, o vartotojas gali nurodyti kokius veiksmus reikia atlikti programai. „Automobilio valdymas“ PA atsakingas už žaidėjo įvesties duomenų pateikimą šiam PA. „Išsaugoto žaidimo pasirinkimas“ arba „Naujo žaidimo pasirinkimas“ PA perduoda duomenis, kurie reikalingi startuoti šiam PA. „Naujo žaidimo pradėjimas“ arba „Išsaugoto žaidimo užkrovimas“ PA perduoda duomenis reikalingus objektų atvaizdavimui. „Žaidimo atvaizdavimas“ PA atsakingas už objektų atvaizdavimą atliekant „Žaidimo logikos“ vykdymo PA. Deformacijos apskaičiavimo PA atsakingas už „Žaidimo logikos“ vykdymo metu įvykusių objektų deformacijas. Garso posistemės PA atsakinga už garsų skleidimą iš programos.

Prieš-sąlygos: inicializuoti duomenys reikalingi žaidimo logikos vykdymui.

Sužadinimo sąlyga: Iškviečiamas metodas, nurodantis pradėti žaidimo logikos vykdymo ciklą.

Po-sąlyga: sėkmingai komunikacija su kitais įtakojančiais PA.

Pagrindinis scenarijus:

1. iškviečiamas metodas nurodantis žaidimo logikos ciklo inicializavimą;
2. surenkami duomenys iš kitų susietų PA.

Alternatyvūs scenarijai:

1. nenumatytų duomenų paketo rinkiniui inicializavimo atveju žaidimo logikos vykdymas yra stabdomas.

Panaudojimo atvejis 7: „Naujo žaidimo pradėjimas“

Aktorius: mobilusis įrenginys.

Tikslas: inicializuoti duomenis, numatytus pagal nutylėjimą, reikalingus žaidimo logikos vykdymui bei žaidimo objektų užkrovimui.

Ryšiai su kitais PA: „Žaidimo logikos vykdymo“ PA suteikia pradinis duomenis reikalingus sėkmingam žaidimo logikos vykdymui. „Žaidimo objektų užkrovimas“ atsakingas už objektų duomenų saugojimą ir pasiekiamumą.

Prieš-sąlygos: duomenų bazėje pateikta objektų pradiniai duomenys ir kita reikalinga informacija.

Sužadinimo sąlyga: iškviečiamas metodas nurodantis gražinti objektų duomenų paketus.

Po-sąlyga: gražinami numatytieji pagal nutylėjimą, žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. iškviečiamas metodas objektų duomenų gražinimui;
2. nuskaitomi numatytieji objektų duomenys iš duomenų bazės;
3. gražinami objektų duomenys.

Alternatyvūs scenarijai:

1. iškviečiamas metodas objektų duomenų gražinimui;
2. duomenų bazėje nerandama duomenų aprašančių objektą;
3. rodomas klaidos rezultatas,

Panaudojimo atvejis 8: „Išsaugoto žaidimo užkrovimas“

Aktorius: mobilusis įrenginys.

Tikslas: inicializuoti duomenis, išsaugotus žaidimo saugojimo metu, reikalingus žaidimo logikos vykdymui bei žaidimo objektų užkrovimui.

Ryšiai su kitais PA: „Žaidimo logikos vykdymo“ PA suteikti pradiniai duomenys reikalingi sėkmingam žaidimo logikos vykdymui. „Žaidimo objektų užkrovimas“ atsakingas už objektų duomenų saugojimą ir pasiekiamumą.

Prieš-sąlygos: duomenų bazėje patiekta išsaugotų objektų duomenys ir kita reikalinga informacija.

Sužadinimo sąlyga: Iškviečiamas metodas nurodantis gražinti objektų duomenų paketus.

Po-sąlyga: gražinami žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. iškviečiamas metodas objektų duomenų gražinimui;
2. nuskaitomi numatytieji objektų duomenys iš duomenų bazės;
3. gražinami objektų duomenys.

Alternatyvūs scenarijai:

1. iškviečiamas metodas objektų duomenų gražinimui;
2. duomenų bazėje nerandama duomenų aprašančių objektą;
3. rodomas klaidos rezultatas.

Panaudojimo atvejis 9: „Žaidimo objektų užkrovimas“

Aktorius: mobilusis įrenginys.

Tikslas: saugoti žaidimo objektų duomenis.

Ryšiai su kitais PA: „Naujo žaidimo pradėjimas“ ir „Išsaugoto žaidimo užkrovimas“ PA nurodo specifinius parametrus reikalingus duomenų mainams su kitais PA.

Prieš-sąlygos: nurodomi parametrai, nusakantys su kokiais objektais reikia duomenų mainų.

Sužadinimo sąlyga: iškviečiamas metodas nurodantis gražinti objektų duomenų paketus.

Po-sąlyga: gražinami žaidimo objektų duomenys.

Pagrindinis scenarijus:

1. iškviečiamas metodas objektų duomenų gražinimui;
2. nuskaitomi numatytieji objektų duomenys iš duomenų bazės.
3. gražinami objektų duomenys.

Alternatyvūs scenarijai:

4. iškviečiamas metodas objektų duomenų gražinimui;
5. duomenų bazėje nerandama duomenų aprašančių objektą;
6. gražinamas klaidos rezultatas.

Panaudojimo atvejis 10: „Žaidimo atvaizdavimas“

Aktorius: mobilusis įrenginys.

Tikslas: žaidimo objektų atvaizdavimas 3D grafika mobiliajame įrenginyje.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo, kuriuos objektus atvaizduoti mobiliajame įrenginyje.

Prieš-sąlygos: priskiriamos objektų reikšmės, reikalingos duomenų atvaizdavimui.

Sužadinimo sąlyga: Iškviečiamas metodas inicializuojantis objektų atvaizdavimą.

Po-sąlyga: žaidimo objektai atvaizduojami 3D grafika mobiliajame įrenginyje.

Pagrindinis scenarijus:

1. priskiriamos žaidimo objektų duomenys;
2. atvaizduojami objektai 3D grafikoje;
3. tęsiamas ciklas vėl nuo pirmo punkto.

Alternatyvūs scenarijai:

1. klaidingai inicializuoti žaidimo objektų duomenys;
2. atvaizduojamas objektas 3D grafikoje;
3. tęsiamas ciklas, iškviečiantis kitų objektų užkrovimą.

Panaudojimo atvejis 11: „Deformacijos apskaičiavimas“

Aktorius: mobilusis įrenginys.

Tikslas: apskaičiuoti žaidimo objektų geometrinės korekcijas.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo, kuriuos ir kaip objektus reikia deformuoti.

Prieš-sąlygos: priskiriamos objekto reikšmės reikalingos geometrinėms korekcijoms apskaičiuoti.

Sužadinimo sąlyga: Iškviečiamas metodas inicializuojantis objektų deformacijos skaičiavimą.

Po-sąlyga: išsaugomi objektai su geometrinėmis deformacijomis.

Pagrindinis scenarijus:

1. priskiriami žaidimo objekto duomenys;
2. vykdomos geometrinio objekto deformacijos;
3. išsaugomos geometrinio objekto deformacijos.

Alternatyvūs scenarijai:

1. priskiriami žaidimo objekto duomenys;

2. geometrinio objekto deformacijos apskaičiavimo metu įvyksta klaida;
3. rodoma deformacijos apskaičiavimo klaida.

Panaudojimo atvejis 12: „Garso posistemė“

Aktorius: mobilusis įrenginys.

Tikslas: skleisti žaidimo garsus.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ PA nurodo žaidimo objektų statusus parametrus.

Prieš-sąlygos: įtraukti visi garso takeliai, reikalingi žaidimui.

Sužadinimo sąlyga: iškviečiamas metodas inicializuojantis garsų posistemę.

Po-sąlyga: mobilusis įrenginys per garsiakalbius skleidžia garsus, įtakotus žaidimo logikos.

Pagrindinis scenarijus:

1. inicializuojama „Garso posistemės“ PA;
2. „Žaidimo logikos vykdymo“ PA nurodo parametrus įtakojančius „Garso posistemės PA“.

Alternatyvūs scenarijai:

1. inicializuojama „Garso posistemės“ PA;
2. žaidimo logika nurodo nenumatytus parametrų rinkinius, įtakojančius „Garso posistemės“ PA;
3. „Garso posistemė“ laukia naujų „Žaidimo logikos vykdymo“ PA parametrų.

Panaudojimo atvejis 13: „Žaidimo valdymas“

Aktorius: žaidėjas.

Tikslas: nurodyti mobiliajam įrenginiui vartotojo pasirinktis.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ priima vartotojo nurodymus.

Prieš-sąlygos: programa reaguoja į vartotojo paspaudimus.

Sužadinimo sąlyga: žaidėjas paspaudžia ant numatytų komponentų.

Po-sąlyga: komponento paspaudimas inicializuoja priskirto metodo iškvietimą.

Pagrindinis scenarijus:

1. žaidėjas paspaudžia ant komponento;
2. komponentas iškviečia jam priskirtą metodą.

Alternatyvūs scenarijai:

1. žaidėjas paspaudžia ant komponento;
2. komponentui nėra priskirtas metodas;
3. Komponento paspaudimas įvykdomas be metodo iškvietimo.

Panaudojimo atvejis 14: „Automobilio valdymas“

Aktorius: mobilusis įrenginys.

Tikslas: nuskaityti giroskopo ar akselerometro duomenis.

Ryšiai su kitais PA: „Žaidimo logikos vykdymas“ priima judėjimo kryptį iš šio PA.

Prieš-sąlygos: aptinkama ir nuskaityta mobiliojo įrenginio akselerometras ir giroskopas.

Sužadinimo sąlyga: programiškai inicializuojamas kartu su žaidimo logikos vykdymo ciklo pradžia.

Po-sąlyga: automobilio valdymo PA perduoda įrenginio padėties parametrus žaidimo logikos vykdymo ciklui.

Pagrindinis scenarijus:

1. inicializuojamas „Automobilio valdymas“ PA kartu su „Žaidimo logikos vykdymo“ PA;
2. gaunami mobiliojo įrenginio padėties parametrai iš akselerometro ir giroskopo;
3. perduodami komponentų duomenys „Žaidimo logikos vykdymo“ PA.

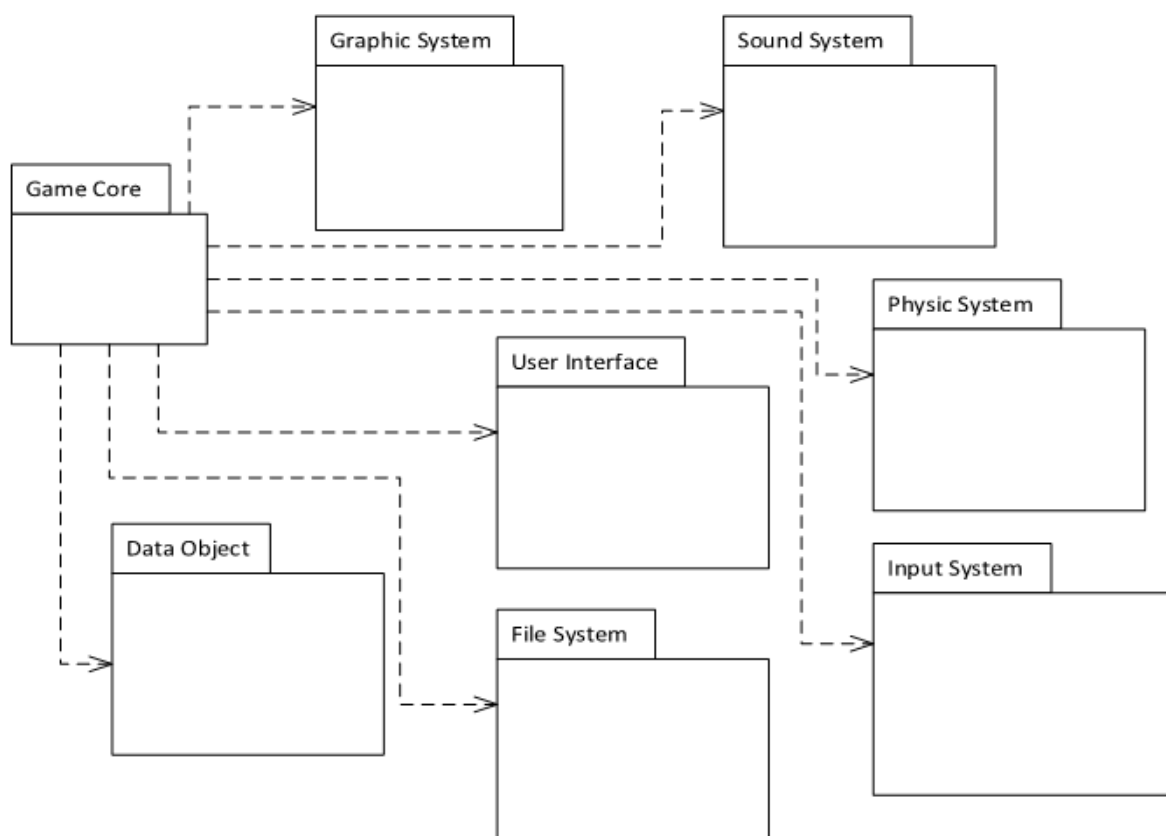
Alternatyvūs scenarijai:

1. inicializuojamas „Automobilio valdymas“ PA kartu su „Žaidimo logikos vykdymas“ PA;

2. nesėkmingas inicializavimas rodo klaidos kodą;
3. „Žaidimo logikos vykdymas“ stabdomas.

4.5. Sistemos statinis vaizdas

Sistema sudaryta iš aštuonių pagrindinių paketų, kurie yra išskaidyti pagal juose vykdomas duomenų apdorojimo sritis. Kiekvienas paketas yra atsakingas už jam pavestas užduotis. Toks paketinis išskaidymas leidžia lengviau rasti nagrinėjamą sritį, išanalizuoti ir rasti klaidas ar praplėsti norimą sritį. Statinis sistemos vaizdas pateiktas žemiau esančiame paveikslėlyje.



Pav. 4.2 Sistemos paketų diagrama

5. TYRIMO DALIS

Šioje dalyje tiriama sukurtos programinės įrangos kokybės analizė.

5.1. Tikslas

Išanalizuoti sukurtos programinės įrangos kokybę su pagrindinėmis programų išeities teksto charakteristikomis (metrikomis).

5.2. Tyrimui naudojamų programų metrikų aprašymas

Tyrimui naudotų metrikų sąrašas su paaiškinimais pateikta žemiau esančioje lentelėje.

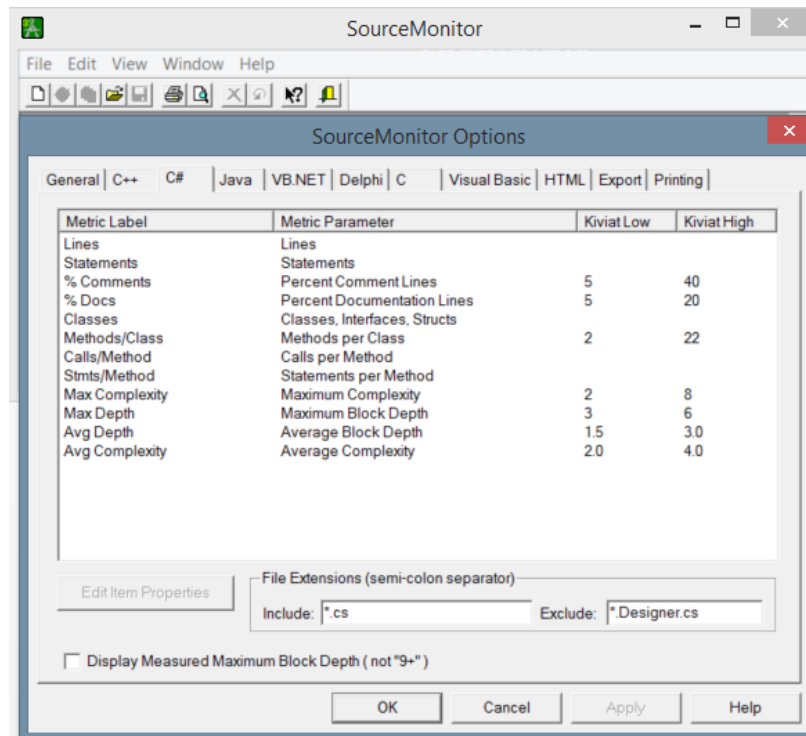
Lentelė 5.1 Naudojamų metrikų aprašymas

Metrika	Aprašymas
Lines	Kodo eilučių kiekis.
Statements	Sakinių skaičius.
Percent Comment Lines	Procentinis eilučių skaičius su komentarais.
Percent Document Lines	Procentinis eilučių skaičius su dokumentacija
Classes, Interfaces, Structs	Klasių, sąsajų, struktūrų skaičius visame projekte.
Method per Class	Metodų skaičius klasėje
Calls per Method	Metodo iškvietimų skaičius.
Statements per Method	Sakinių skaičius metode.
Maximum Complexity	Didžiausias sudėtingumas.
Maximum Block Depth	Didžiausias bloko gylis.
Average Block Depth	Vidutinis bloko gylis.
Average Complexity	Vidutinis sudėtingumas.

5.3. Tyrimui naudojama programinė įrangos aprašymas

Tyrimui pasirinkta nemokama programa SourceMonitor. Šią programinę įrangą galima atsisiųsti iš kūrėjų svetainės: <http://www.campwoodsw.com/sourcemonitor.html>

Tyrimui naudotą programinės įrangos SouceMonitor v3.5 versiją galima atsisiųsti per šią nuorodą: <http://www.campwoodsw.com/SMSetupV350.exe>.



Pav. 5.1 SourceMonitor v3.5 programinės įrangos išvaizda

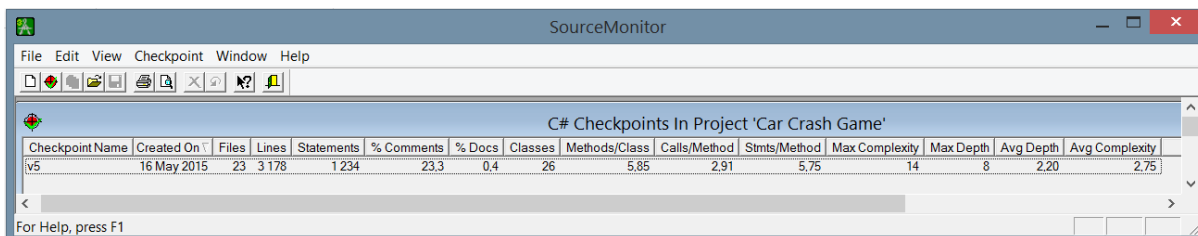
SourceMonitor yra nemokama programa, kurios dėka galima pažvelgi į savo programos kodo vidų ir išanalizuoti įvairias programų metrikas. Ši programa yra parašyta C++ programavimo kalba, todėl dirba labai greitai: išanalizuoja 10000 eilučių kodo per vieną sekundę.

Pagrindinės SourceMonitor galimybės:

1. greitas kodo analizavimas ir metrikų pateikimas;
2. gali matuoti metrikas išeities kodui parašytam C++, C, C#, VB.NET, Java, Delphi, Visual Basic (VB6) ir HTML programavimo kalbomis;
3. pateikia metodų ir funkcijų lygio metrikas išeities kodui parašytam C++, C, C#, VB.NET, Java, ir Delphi programavimo kalbomis;
4. atvaizduoja metrikas lentelėse ir diagramose, įskaitant ir Kiviat diagramas; gali išeksportuoti surinktą informaciją į XML ir CSV failus, tolesnei analizei.

5.4. Rezultatai

Apdorojus kuriamos sistemos projektą su SourceMonitor programa, buvo gauti bendri viso projekto metrikų rezultatai.

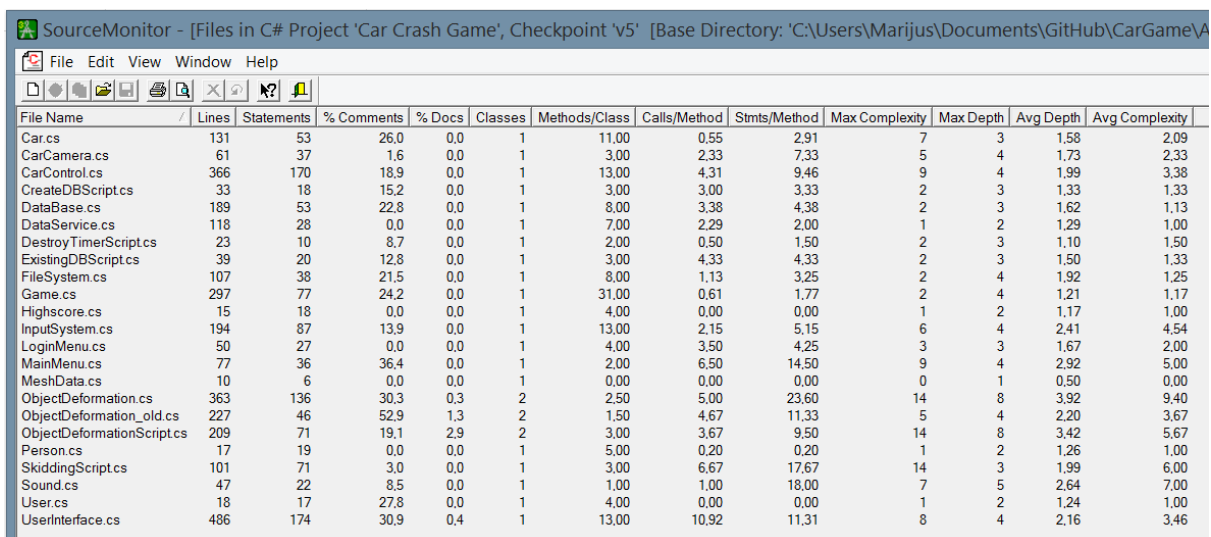


The screenshot shows the SourceMonitor application window. The title bar reads 'SourceMonitor'. The menu bar includes 'File', 'Edit', 'View', 'Checkpoint', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons. The main area displays a table titled 'C# Checkpoints In Project 'Car Crash Game''. The table has the following columns: Checkpoint Name, Created On, Files, Lines, Statements, % Comments, % Docs, Classes, Methods/Class, Calls/Method, Stmts/Method, Max Complexity, Max Depth, Avg Depth, and Avg Complexity. The data row for checkpoint 'v5' shows: 16 May 2015, 23, 3 178, 1 234, 23.3, 0.4, 26, 5.85, 2.91, 5.75, 14, 8, 2.20, 2.75.

Checkpoint Name	Created On	Files	Lines	Statements	% Comments	% Docs	Classes	Methods/Class	Calls/Method	Stmts/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity
v5	16 May 2015	23	3 178	1 234	23.3	0.4	26	5.85	2.91	5.75	14	8	2.20	2.75

Pav. 5.2 Bendra viso projekto metrikų suvestinė

Source Monitor taip pat turi galimybę peržiūrėti ir atskirų projekto failų metrikas. Rezultatai pateikiami žemiau esančiame paveikslėlyje.

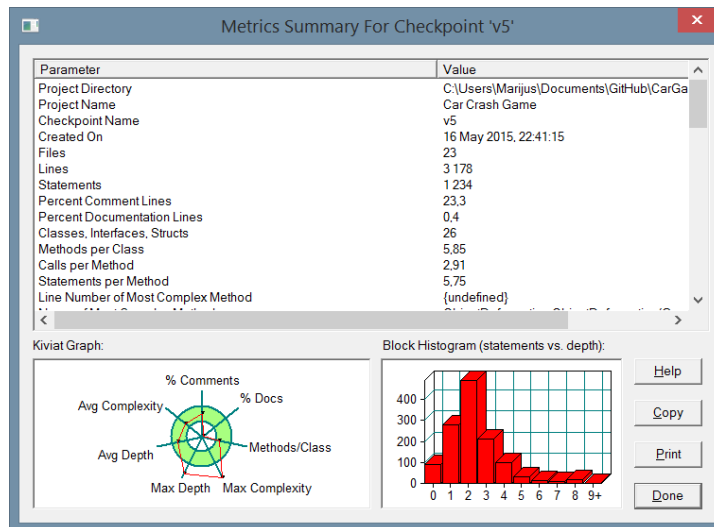


The screenshot shows the SourceMonitor application window with a detailed table of file metrics. The title bar reads 'SourceMonitor - [Files in C# Project 'Car Crash Game', Checkpoint 'v5'] [Base Directory: 'C:\Users\Marijus\Documents\GitHub\CarGame\A...'. The menu bar includes 'File', 'Edit', 'View', 'Window', and 'Help'. Below the menu bar is a toolbar. The main area displays a table with the following columns: File Name, Lines, Statements, % Comments, % Docs, Classes, Methods/Class, Calls/Method, Stmts/Method, Max Complexity, Max Depth, Avg Depth, and Avg Complexity. The data rows list various files and their corresponding metrics.

File Name	Lines	Statements	% Comments	% Docs	Classes	Methods/Class	Calls/Method	Stmts/Method	Max Complexity	Max Depth	Avg Depth	Avg Complexity
Car.cs	131	53	26.0	0.0	1	11.00	0.55	2.91	7	3	1.58	2.09
CarCamera.cs	61	37	1.6	0.0	1	3.00	2.33	7.33	5	4	1.73	2.33
CarControl.cs	366	170	18.9	0.0	1	13.00	4.31	9.46	9	4	1.99	3.38
CreateDBScript.cs	33	18	15.2	0.0	1	3.00	3.00	3.33	2	3	1.33	1.33
DataBase.cs	189	53	22.8	0.0	1	8.00	3.38	4.38	2	3	1.62	1.13
DataService.cs	118	28	0.0	0.0	1	7.00	2.29	2.00	1	2	1.29	1.00
DestroyTimerScript.cs	23	10	8.7	0.0	1	2.00	0.50	1.50	2	3	1.10	1.50
ExistingDBScript.cs	39	20	12.8	0.0	1	3.00	4.33	4.33	2	3	1.50	1.33
FileSystem.cs	107	38	21.5	0.0	1	8.00	1.13	3.25	2	4	1.92	1.25
Game.cs	297	77	24.2	0.0	1	31.00	0.61	1.77	2	4	1.21	1.17
Highscore.cs	15	18	0.0	0.0	1	4.00	0.00	0.00	1	2	1.17	1.00
InputSystem.cs	194	87	13.9	0.0	1	13.00	2.15	5.15	6	4	2.41	4.54
LoginMenu.cs	50	27	0.0	0.0	1	4.00	3.50	4.25	3	3	1.67	2.00
MainMenu.cs	77	36	36.4	0.0	1	2.00	6.50	14.50	9	4	2.92	5.00
MeshData.cs	10	6	0.0	0.0	1	0.00	0.00	0.00	0	1	0.50	0.00
ObjectDeformation.cs	363	136	30.3	0.3	2	2.50	5.00	23.60	14	8	3.92	9.40
ObjectDeformation_old.cs	227	46	52.9	1.3	2	1.50	4.67	11.33	5	4	2.20	3.67
ObjectDeformationScript.cs	209	71	19.1	2.9	2	3.00	3.67	9.50	14	8	3.42	5.67
Person.cs	17	19	0.0	0.0	1	5.00	0.20	0.20	1	2	1.26	1.00
SkiddingScript.cs	101	71	3.0	0.0	1	3.00	6.67	17.67	14	3	1.99	6.00
Sound.cs	47	22	8.5	0.0	1	1.00	1.00	18.00	7	5	2.64	7.00
User.cs	18	17	27.8	0.0	1	4.00	0.00	0.00	1	2	1.24	1.00
UserInterface.cs	486	174	30.9	0.4	1	13.00	10.92	11.31	8	4	2.16	3.46

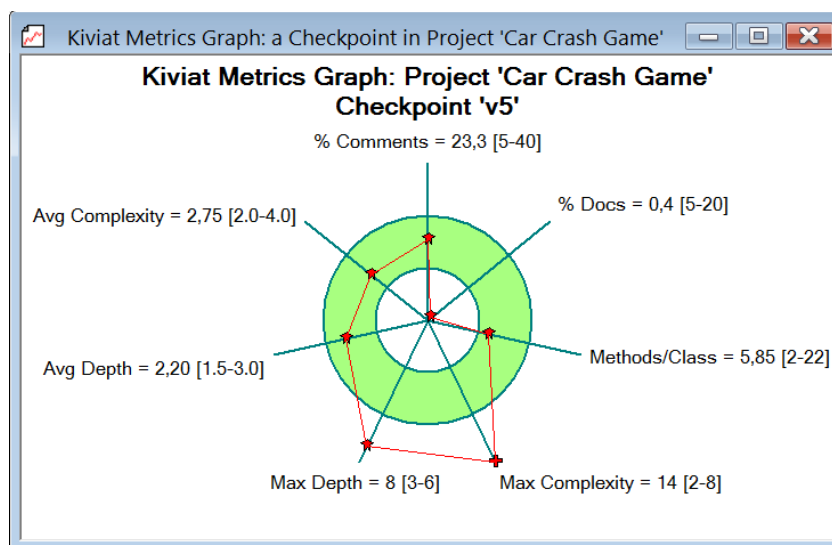
Pav. 5.3 Atskirų projekto failų metrikos

Bendri projekto rezultatų suvestinės rezultatai pateikti žemiau esančiame paveikslėlyje.



Pav. 5.4 Bendra projekto rezultatų suvestinė

Siekiant įvertinti visos sistemos prižiūrimumo lygį, nuspręsta naudoti Radar(Kiviat) diagramą. Su šia diagrama atvaizduojami visos sistemos prižiūrimumo lygis grafiškai pagal išmatuotas metrikas.



Pav. 5.5 Kiviat metrikų grafikas

Remiantis gautais Kiviat metrikų grafiko rezultatais, matoma kuriamoje sistemoje nėra dokumentacijos. Didžiausias sudėtingumo rodiklis gerokai viršija rekomenduojamas ribas. Remiantis atskirų failų metrikų rezultatais (Pav. 4.4.2) didžiausias sudėtingumas randamas klasėse: ObjectDeformation.cs, Object DeformationScripts.cs SkiddingScripts.cs, CarControl.cs, MainMenu.cs. Šioms klasėms reikalinga pertvarka.

Remiantis metrikų rezultatais (Pav. 4.4.2) didžiausias gylis randamas klasėse: ObjectDeformation.cs ir ObjectDeformationScripts.cs. Šiuose klasės metoduose bloko gylis viršija rekomenduojama didžiausią 6 vertę. Reikalinga pertvarka mažinant bloką gylius.

Remiantis Kiviat metrikų diagrama (Pav. 4.4.4) matuojamos metrikos: metodų skaičiaus klasėje metrika, vidutinio gylio metrikos, vidutinio sudėtingumo metrika bei komentarų metrikos yra rekomenduojamose ribose.

5.5. Rekomendacijos:

1. kuriamoje sistemoje reikalingas dokumentacijos rašymas;
2. ObjectDeformation.cs, Object DeformationScripts.csSkiddingScripts.cs, CarControl.cs, MainMenu.cs klasėms reikalinga skaidymas į mažesnes klases siekiant sumažinti sudėtingumą;
3. ObjectDeformation.cs, Object DeformationScripts.cs metodams reikalingas bloko gylio mažinimas.

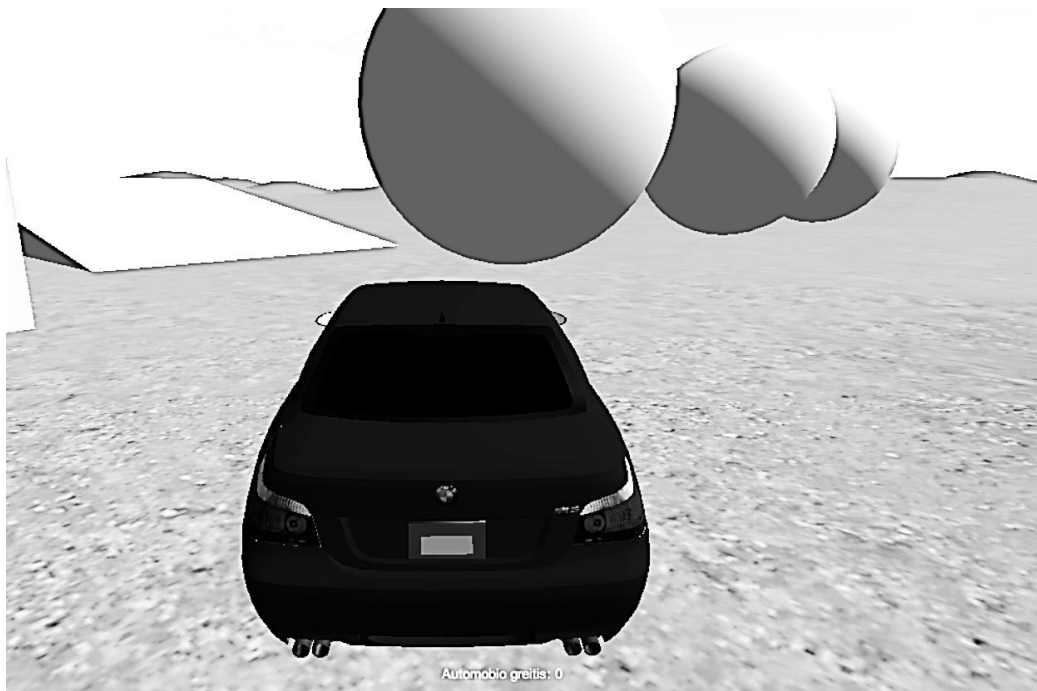
Atlikus paminėtus pakitimus, programa taptų lengviau skaitoma, ją būtų lengviau prižiūrėti, bei rasti ir ištaisyti atsiradusias klaidas, ar įvedant naujas funkcijas.

6. EKSPERIMENTINĖ DALIS

Šiame skyriuje pateikiama sukurta programinė įranga, kurioje realizuotas geometrinių objektų deformacijos algoritmas, eksperimentinis įvertinimas. Siekiama rasti optimaliausius sprendimo parametrus, leisiančius deformacijos algoritmui greitai rasti kokybiškiausią sprendimą mechaniniuose deformacijos algoritmo uždaviniuose. Taip pat svarbu išanalizuoti, ar deformacijos algoritmo ir grafikos apdorojimo sistemos optimizacija pritaikyta ribotiems skaičiavimo resursams, ar jos pakaks mobiliesiems įrenginiams, turintiems vidutinės spartos skaičiavimo resursus.

6.1. Tyrimo Aplinka

Tiriama programinė įranga yra žaidybinio pobūdžio, todėl joje nėra tik tiriamasis objektas (automobilis) su jį veikiančiomis išorinėmis jėgomis. Tyrimo simuliuojamiems duomenims turi įtakos ir pati aplinka. Tyrimo aplinka, kaip ir pats žaidimas, sudarytas iš tam tikros reljefinės aplinkos bei joje esančių tam tikrų geometrinių objektų, kurių pagrindinė paskirtis yra kliūtis buvimas žaidimo aplinkoje. Žaidimo aplinkoje valdomas tam tikros konstrukcijos geometrinis objektas, kontaktuojantis su geometrine kliūtimi ir leidžia simuliuoti deformacijos algoritmo duomenis, kurių vizualiniai rezultatai pateikiami vartotojams 3D grafiniu modeliu.



Pav. 6.1 Eksperimentinė aplinka

Siekiant simuliuoti realaus gyvenimo aplinkoje esančius objektus ir jų įtaką kontaktuojantiems dviem geometriniais objektams, jų deformacijos proceso sugeneruotiems rezultatams atvaizduoti, buvo nuspręsta naudoti tam tikrus supaprastintus realaus gyvenimo geometrinius objektus. Žaidimo aplinką (Pav.5.1.1) sudaro trampilynai, sferos, stačiakampio formos geometriniai objektai. Su kiekviena geometrine figūra galima simuliuoti skirtingas išorinių jėgų įtakas nagrinėjam objektui, kurios sukelia deformacijas tiriamajam objektui.

Simuliuojant išorinių jėgų poveikį nagrinėjam objektui svarbus jo sukeltos jėgos krypties vektorius, nukreiptas į tiriamąjį objektą, jėgos paveiktą geometrinio objekto konstrukcijos elemento vietą bei atsiradusios išorinės jėgos dydis, įtakojantis nagrinėjamo objekto konstrukcijos deformacijos procesą.

Žaidimo aplinkoje esančios sferos pavidalo geometriniai objektai yra veikiami pakankamai mažos gravitacinės jėgos, todėl jie yra puikūs objektai, leisiantys simuliuoti išorines jėgas, kurių vektorių kryptys yra nukreiptos iš viršaus į nagrinėjamą objektą. Nagrinėjamas objektas, atsitrenkęs į sferą pakels geometrinį objektą į viršų, o krisdamas žemyn ir pataikydamas ant tiriamojo geometrinio objekto, sukels išorinę jėgą, kurios veikimo kryptis bus nukreipta į tiriamąjį objektą.

Aplinkoje esantys trampilynai ir reljefiniai nelygumai suteikia analizuojam objektui galimybę atitrūkti trumpam laikui nuo važiojamosios dalies. Jam pakilus, dėl veikiamos gravitacinės jėgos, vektoriaus, kuris yra nukreiptas priešingai objekto kilimo trajektorijai, grąžina nagrinėjamą objektą atgal į reljefinę dalį. Objekto ir reljefo kontakto metu išorinės jėgos vektorius nukreiptas iš apačios į viršų, todėl galima stebėti deformacijos rezultatus, kai išorinės jėgos vektorius paveikia iš apačios nagrinėjamą objektą.

Kubų formos geometrinių objektų pagrindinė paskirtis yra analizuojam objektui suteikti išorinės jėgos vektorių, nukreiptą tiesiai į nagrinėjamą objektą. Šios jėgos vektorius geriausiai atsispindi dažniausiai kontaktuojančiose objekto sąveikose tarpusavyje. Kontaktuojančių objektų jėgų vektoriai nukreipti Y ar X ašyse. Dėl šių jėgų vektoriaus krypties, sąlygiškai dažno poveikio realaus gyvenime, yra vienas esminių nagrinėjamų deformacijos poveikių duomenų analizės srityje.

Šios programinės įrangos pagrindinė paskirtis yra deformacijų vizualizavimas, todėl nagrinėjamas geometrinis objektas šiuo atveju yra tam tikros formos geometrinė mašina. Šis geometrinis objektas yra pagrindinis tiriamojo darbo eksperimentinis modelis. Šiuo modeliu simuliuojamos deformacijos algoritmo tyrimo rezultatais remsimės, norėdami vizualiai įvertinti deformacijos algoritmo sugeneruotus rezultatus jo kokybei įvertinti. Mašinos modelis pateiktas žemiau esančioje nuotraukoje.



Pav. 6.2 Eksperimentiniams duomenimis simuliuoti naudojamas geometrinis objektas

Eksperimentiniam tyrimui pasirinktas modelis yra aukštos kokybės. Mašinos modelis yra sudarytas iš atskirų dalių. Modelyje yra išskirtos atskiros dalys: langai, durys, ratai, priekiniai ir galiniai bamperiai, bagažinė, priekinis pakeliamasis dangtis ir t.t. Šio modelio dalys yra pakankamai aukštos kokybės, todėl leis geriau įvertinti deformacijos poveikį, nagrinėjamos dalims, jas paveikus išorinėmis jėgomis.

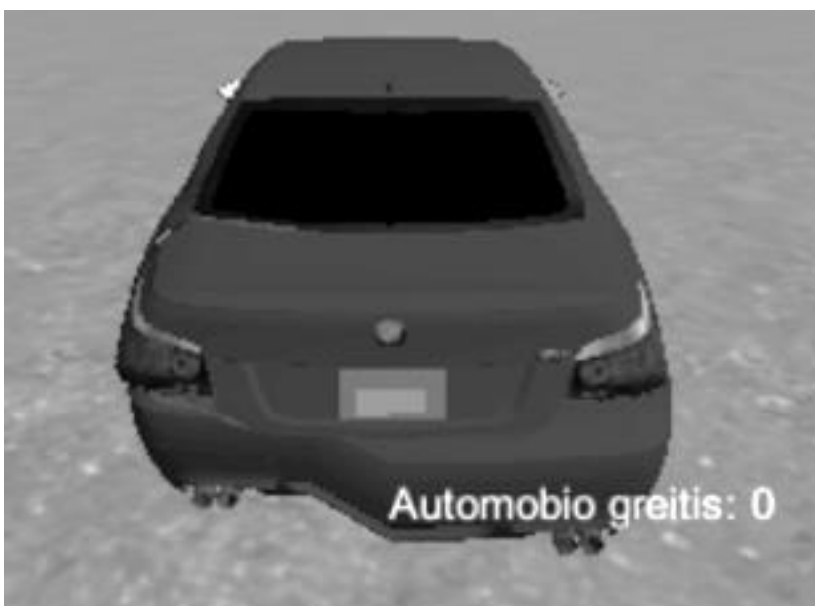
6.2. Grafinis deformacijos kokybės įvertinimas

Eksperimento metu, tiriant mašinos modelio ir kontaktuojančio geometrinio objekto sąveiką tarpusavyje, buvo pastebėta, jog kontaktuojančios geometrinių objektų konstrukcijos struktūra pakito netinkamai. Po deformacijos iš algoritmo duomenų apdorojimo buvo tikimasi, jog nagrinėjamos konstrukcijos elemento vietas paveikusios išorinės jėgos sritys tolygiai padarys konstrukcijos elementų poslinkį. Tačiau kontaktuojančios mašinos dalys, tokios kaip galinis bamperis, dešinysis žibintas, po deformacijos algoritmo duomenų apdorojimo visai pranyko. Nagrinėjamo geometrinio automobilio konstrukcijos pateiktos žemiau esančiame paveikslėlyje.



Pav. 6.3 Geometrinio objekto konstrukcijos elementų dingimas po kontakto

Nagrinėjant deformacijos algoritmą, buvo nuspręsta padidinti standumo koeficientą ir iš naujo pakartoti susimuliuoto automobilio ir kito geometrinio objekto susidūrimo kontaktą. Po kontakto, nagrinėjama geometrinio konstrukcijos sritis, šiuo atveju, galinis dešinysis bamperis, deformacijos proceso algoritmas įvykdė pakankamai tikslų konstrukcijos elementų poslinkį. Nagrinėjamos sritys vizualiai įlinko, o grafiniai objektų rezultatai pakankamai tikroviški, jei juos lygintume su realiame gyvenime vykšančiomis mechaninėmis deformacijomis. Grafiniai konstrukcijos elementų postūmio rezultatai pateikiami žemiau esančiame paveikslėlyje.



Pav. 6.4 Galinio bamperio konstrukcijos elementų poslinkis po kontakto su kitu geometrinio objektu.

Siekiant įvertinti nagrinėjamo objekto sąveiką su labiau nei įprasta padidėjusiomis išorinėmis jėgomis, kurių jėgos vektorius yra nukreiptas iš viršaus į nagrinėjamą konstrukciją, buvo susimuluota situacija, kai geometrinės sferos objektas buvo numetamas ant tiriamojo objekto. Jėgos paveiktoje srityje baigtinių elementų tinklelis plyšo, o gauti rezultatų duomenys neatitiko realiame gyvenime esančių mechaninių deformacijų. Gauti konstrukcijų elementų poslinkių vizualiniai rezultatai pateikti žemiau esančiame paveikslėlyje.



Pav. 6.5 Baigtinių elementų tinklelio trūkimas po deformacinės jėgos poveikio nagrinėjamai sričiai.

Po konstrukcijos struktūros elementų poslinkių funkcionalumo pertvarkos buvo vėl bandoma tiriamojo geometrinio objekto konstrukcijos elementus paveikti išorinėmis jėgomis, kurių jėgos vektorius nukreiptas iš viršaus į nagrinėjamą objektą. Po deformacijos ir konstrukcijos elementų poslinkių, nagrinėjamo geometrinio objekto stogo baigtinių elementų tinklelis įgavo formą, panašią į sferos įlinkio pasekmes. Gauti vizualiniai rezultatai pateikti žemiau esančiame paveikslėlyje.



Pav. 6.6 Geometrinio objekto konstrukcijos rezultatai po geometrinės sferos objekto nukritimo ant nagrinėjamo objekto paviršiaus.

Siekiant tikroviškumo, buvo išmėginta tiriamojo geometrinio objekto sąveika su reljefu po geometrinio objekto atotrūkio nuo reljefo. Geometrinio objekto nusileidimo metu, įvykus kontaktui su reljefu, didžioji dalis deformacinės jėgos yra nukreipiama į objekto ratus. Kadangi realiame gyvenime esant panašiai žaidimo simuliuojamai situacijai visa jėga tenkanti ratams yra nukreipiama į amortizatorius ir taip sumažinamas deformacijos poveikis, nuspręsta jog nusileidimo metu, kontaktuojant geometriniam objektui su reljefu, neturės didesnės įtakos. Mat nusileidimo metu kontaktavimas turės didesnę reikšmę tik tuo atveju, kai nusileidimo metu kontaktuoja ne tik ratai, tačiau ir tam tikros geometrinio objekto sritys. Gauti rezultatai pateikti žemiau esančiame paveikslėlyje.



Pav. 6.7 Geometrinio objekto konstrukcijos rezultatai po sąveikos su reljefu.

Atlikus eksperimentus su įvairiomis išorinėmis jėgomis, kurios deformacijos jėgos vektoriai įvairiomis kryptimis paveikia tiriamąjį geometrinį objektą, buvo nustatyta tam tikrų neatitikimų. Simuliuoti aukštos kokybės vizualinius rezultatus buvo sudėtinga dėl resursų stokos mobiliuosiuose įrenginiuose, atliekant skaičiavimo procesus. Gauti rezultatai tik iš dalies atitinka realiame gyvenime vykstančias mechanines deformacijas, kuomet dėl kontakto su kitais objektais įvyksta nagrinėjamų objektų deformacijos. Tačiau ši programa yra tik žaidybinio tipo, todėl gaunami rezultatai po deformacijos algoritmo duomenų apdorojimo ir nagrinėjamų objekto sričių poslinkių perstumdymo, supažindins bent šiek tiek su realiame gyvenime vykstančiomis deformacijomis.

6.3. Deformacijos algoritmo greitimeikos tyrimas

Sistema modeliuojanti 3D geometrinių objektų deformacijos procesą turi atlikti deformacijos skaičiavimus su vidutinės klasės mobiliaisiais įrenginiais. Siekiant iširti tinkamiausią sistemos veikimą, būtina iširti deformacijos algoritmo įtaka skaičiavimo resursų poreikiui, reikalingam gauti rezultatus po geometrinio objekto deformacijos. Kadangi deformacijos algoritmas yra iteracinis procesas suformuojantis galutinius rezultatus, buvo nuspręsta iširti jo įtaką laiko atžvilgiu. Kadangi programa yra skirta mobiliesiems įrenginiams, buvo tikslinga atlikti tyrimą mobiliajame įrenginyje. Įrenginyje buvo simuliuojamos didėjančios iteracijos ir laiko poreikių priklausomybė šių iteracijos metu apdorojamų duomenų kiekiui apdoroti. Algoritmo iteracijoms simuliuoti parenkamas geometrinės formos

automobilis, kuris važiuodamas tiesiai, atsitreikia į priekyje esantį geometrinį objektą, taip sužadindamas deformacijos procesą.



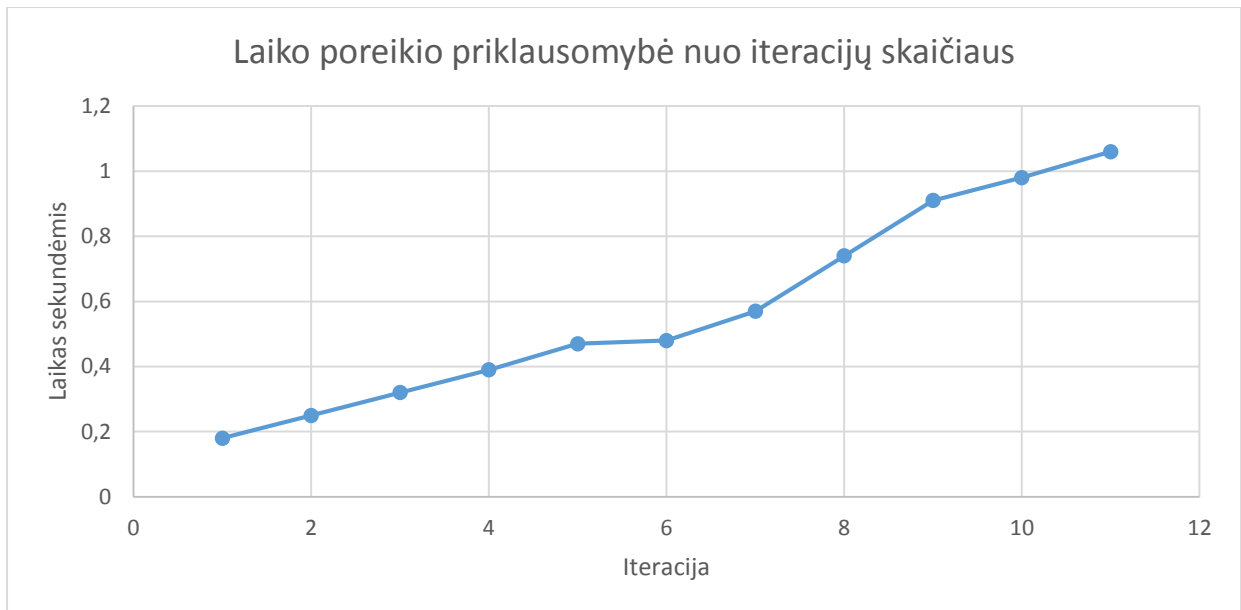
Pav. 6.8 Sistemos spartos tyrimas simuliuojant rezultatus Windows Phone emuliatoriumi.

Scenarijaus simuliuojama algoritmo iteracijų įtaka skaičiavimo poreikiui, šiuo atveju, laikui, pateikiama žemiau esančioje lentelėje.

Lentelė 6.1 Iteracijų kiekio įtaka duomenų apdorojimo laikui

Iteracijos numeris	Reikalingas laikas duomenims apdoroti (s)
1	0,18
2	0,25
3	0,32
4	0,39
5	0,47
6	0,48
7	0,57
8	0,74
9	0,91
10	0,98
11	1,06

Atliekant tyrimą buvo pastebėta, jog nagrinėti iteracijų kiekio priklausomybę netenka prasmės jau po 11 iteracinio skaičiavimo, nes atliekant skaičiavimus su 11-ka iteracijų užtrunka daugiau nei 1s. Toks skaičiavimo poreikis stabdo žaidimą ir gali nebetenkinti vartotojo keliamų lūkesčių. Todėl nagrinėjimas daugiau nei 11-os iteracijų nebetenka prasmės. Grafinis duomenų atvaizdavimas pateiktas žemiau esančiame pavyzdyje.



Pav. 6.9 Grafiniai laiko ir iteracijų kiekio priklausomybės rezultatai

Nagrinėjant iteracijų kiekio ribas matome, jog priklausomybė kinta sąlygiškai tiesine funkcija. Tačiau galima daryti prielaidą, jog didėjant iteracijų kiekiui, funkcinė priklausomybė turėtų kisti hiperboline funkcija. Nes kiekvienos iteracijos metu turėtų didėti viršūnių skaičius ir jas perstumdyti reikalingas didesnis skaičiavimo resursų poreikis. Tačiau laikas reikalingas įgyvendinti iteraciją reikalauja pakankamai daug skaičiavimo resursų. Sumažinus iteracijos skaičiavimo poreikius, rezultatai labiau atspindėtų tikruosius rezultatus.

Kuriama sistema yra tik žaidybinio tipo ir skirta mobiliesiems įrenginiams, neturintiems didelių skaičiavimo resursų, lyginant su šiuolaikiniais kompiuteriais. Tad nuspręsta kuo labiau taupyti skaičiavimo poreikius. Todėl buvo pasirinkta tik su vienu algoritmu vykdoma iteracija. Gautas pakankamai aukštos kokybės rezultatus atspindintis geometrinio objekto deformacijos algoritmas.

7. IŠVADOS

Projekto metu buvo sukurtas žaidybinio tipo žaidimas, kuriame yra realizuotas baigtinių elementų deformacijos algoritmas, leidžiantis pakankamai greitai rasti kokybiškiausią sprendimą mechaniniuose deformacijos uždaviniuose. Ši sistema yra skirta mobiliųjų įrenginių platformai, todėl kūrimo ir analizės metu buvo atsižvelgta į skaičiavimo resursų stoką, atliekant deformacijos algoritmų skaičiavimo procesus. Todėl ši programa buvo kuriama atsižvelgiant į turimus mobiliųjų įrenginių skaičiavimo resursus.

Tiriamąo projekto metu taip pat buvo:

1. išanalizuotas realiame gyvenime veikiamų išorinių jėgų poveikis nagrinėjamiems objektams.
2. Apžvelgtas baigtinių elementų metodų algoritmas ir jo principai.
3. Ištirtos geometrinių objektų deformacijos elgsenos ir jų procesiškumas..
4. Įvertinti skaičiavimo poreikiai, reikalingi deformacijos apskaičiavimui, naudojant baigtinių elementų algoritmą.
5. Įvertintas tikslumas, reikalingas geometrinių objektų deformacijos vizualizavimui.
6. Mechaninėms deformacijoms rezultatams sugeneruoti programoje buvo pritaikytas baigtinių elementų metodo principas, naudojant baigtinių elementų tinklelio deformacijos poslinkių koordinačių plokštumoje analizę.
7. Išanalizuota sukurtos programinės įrangos kokybė su pagrindinėmis programų išeities teksto charakteristikomis (metrikomis).
8. Pastebėta, kad turint ribotus skaičiavimo resursus, šiuo atveju, vidutinės klasės telefono turimus skaičiavimo resursus, sunku simuliuoti aukštos kokybės geometrinių objektų deformacijos procesus.
9. Nustatyta, kad gauti rezultatai po deformacijos proceso tik iš principo atitinka realaus gyvenime vykstančias mechanines deformacijas.
10. Siekiant nustatyti aukštesnės kokybės deformacijos poslinkiams, reikalingas didesnis dėmesys baigtinių elementų tinkleliams, juos prisitaikant prie nagrinėjamo paviršiaus.
11. Tyrimo metu, siekiant taupyti skaičiavimo resursus, buvo nuspręsta vykdyti tik vieną algoritmo iteracijos procesą.
12. Siekiant kokybiško rezultato mechanikos deformacijos uždaviniuose, baigtinių elementų tinklo generavimas yra viena iš aktualiausių nagrinėjamų sričių.

13.LITERATŪRA

- [1] BAIGTINIŲ ELEMENTŲ METODAS MECHANIKOS INŽINERIOJE [Žiūrėta 2013 11 30], prieiga per internetą: < http://stud.ppf.ktu.lt/vaiaciulis/paskaitos/bemi/bemi_01.pdf> .
- [2] BAIGTINIŲ ELEMENTŲ METODAS [Žiūrėta 2013 11 30], prieiga per internetą: < http://moodle.vgtu.lt/pluginfile.php/82459/mod_resource/content/1/MM1_html_/Vadovelis/d_ata/html/16-07s.htm> .
- [3] The Finite Element Method for Engineers redagavo Kenneth H. Huebner [Žiūrėta 2013 11 30], prieiga per internetą: < http://www.google.lt/books?hl=lt&lr=&id=f3MZE1BYq3AC&oi=fnd&pg=PR17&ots=P52U2Nh--F&sig=kocT8DuMV1xrx6P3pMTYnS8rk6k&redir_esc=y#v=onepage&q&f=false> .
- [4] A First Course in the Finite Element Method Autorus (-iai): Daryl L. Logan [Žiūrėta 2014312 01], prieiga per internetą: < http://books.google.lt/books?id=KptPymzHa_gC&printsec=frontcover&dq=The+Finite+Element+Method&hl=lt&sa=X&ei=oUqbUp6XL8qp7AbQ8IHQCQ&ved=0CDgQ6AEwAQ#v=onepage&q=The%20Finite%20Element%20Method&f=false> .
- [5] The Finite Element Method and Its Reliability Autorius (-iai): Ivo Babuška, Theofanis Strouboulis [Žiūrėta 2013 12 01], prieiga per internetą: < http://books.google.lt/books?id=_Gv9KMOra7sC&printsec=frontcover&dq=The+Finite+Element+Method&hl=lt&sa=X&ei=oUqbUp6XL8qp7AbQ8IHQCQ&ved=0CEkQ6AEwAw#v=onepage&q=The%20Finite%20Element%20Method&f=false> .
- [6] Understanding and Implementing the Finite Element Method Autorius (-iai): Mark S. Gockenbach [Žiūrėta 2013 12 01], prieiga per internetą: < <http://books.google.lt/books?id=sP-Acc5d3Q4C&printsec=frontcover&dq=The+Finite+Element+Method&hl=lt&sa=X&ei=oUqbUp6XL8qp7AbQ8IHQCQ&ved=0CFEQ6AEwBA#v=onepage&q=The%20Finite%20Element%20Method&f=false>> .
- [7] Introduction to Finite Element Method [Žiūrėta 2013 12 01], prieiga per internetą: < http://mms2.ensmp.fr/tribo_paris/lectures/Introduction_FE.pdf> .

- [8] Lectures on The Finite Element Method By Ph. Ciarle [Žiūrėta 2013 12 01], prieiga per internetą: < <http://www.math.tifr.res.in/~publ/ln/tifr49.pdf>>.
- [9] R. Barauskas, R. Belevičius, R. Kačianauskas. *Baigtinių elementų metodo pagrindai*, Vilnius, Technika, 2004.
- [10] A gentle introduction to the Finite Element Method By Francisco Javier Sayas [Žiūrėta 2013 12 01], prieiga per internetą: < <http://www.math.udel.edu/~fjsayas/anIntro2FEM.pdf>>.
- [11] Short term course on finite element method for engineering applications [Žiūrėta 2013 12 01], prieiga per internetą: < <http://www.iitmandi.ac.in/fema2013/>>.
- [12] Mixed finite element methods with applications to viscoelasticity and gels By Marie E. Rognes [Žiūrėta 2013 12 01], prieiga per internetą<<http://www.ima.umn.edu/~arnold/papers/RognesThesis.pdf>> .
- [13] INTRODUCTION TO THE FINITE ELEMENT METHOD [Žiūrėta 2013 12 01], prieiga per internetą<http://catarina.udlap.mx/u_dl_a/tales/documentos/lim/albores_b_mi/capitulo7.pdf/.
- [14] Finite Element Analysis in Roll Forming Applications [Žiūrėta 2013 12 01], prieiga per internetą<<http://www.rfcorp.com/literature-a-tech-info/white-papers/finite-element-analysis>>.
- [15] Large Deformation Finite Element Analysis for Offshore Applications [Žiūrėta 2013 12 01], prieiga per internetą: <<http://www.civil.iitb.ac.in/~dns/IACMAG08/pdfs/M29.pdf>>.
- [16] Programų priežiūros ir tobulinimo (T120M013) modulio paskaitų medžiaga.
- [17] Kaner, C., & Bond, W. P. (2004). Software engineering metrics: What do they measure and how do we know? Proceedings of 10th International Software Metrics Symposium, prieiga per internetą: <<http://testingeducation.org/a/metrics2004.pdf>>.
- [18] D.M. Coleman, D. Ash, B. Lowther, and P.W. Oman, Using Metrics to Evaluate Software System Maintainability, IEEE Computer, 1994, 44-49, prieiga per internetą: <<http://www.idi.ntnu.no/emner/it3605/artikler/r8044.pdf>>.
- [19] Greg Pope. Measuring “Good” in Good Enough Software. University of California, Lawrence Livermore National Laboratory, TR No. W-7405-Eng-48, prieiga per internetą: <http://home.comcast.net/~gregorypope/published_papers/Measuring_Good2.doc>.

[20] S.R. Chidamber, C.F. Kemerer. A Metrics suite for Object Oriented design. M.I.T. Sloan School of Management E53-315. 1993, prieiga per internetą:

http://faculty.salisbury.edu/~stlauterburg/COSC425/MetricForOOD_ChidamberKemerer94.pdf.

[21] Victor Basili, Lionel Briand and Walcelio Melo. A Validation of Object-Oriented Design Metrics as Quality Indicators. IEEE Transactions on Software Engineering. Vol. 22, No. 10, October 1996, prieiga per internetą:

<<http://www.cs.umd.edu/users/basili/publications/journals/J60.pdf>>.

[22] R Barauskas, *Baigtinių elementų metodas su MATLAB ir COMSOL Multiphysics*, Kaunas, 20xx.

[23] R Barauskas, R Barauskienė. *Fizikine elgsena grįstos animacijos: standžiųjų kūnų sistemos*, Mokomoji knyga, ISBN 978-9955-25-936-7, Kaunas, 2011, 120p.

[24] P. Žiliukas, R. Barauskas. *Mechaniniai virpesiai*, VPU leidykla, 2008, 410p.

[25] R. Barauskas. *Skaitiniai inžinerijos metodai*, UAB Vitae Litera, 2007, 190p.

[26] R. Barauskas, R. Belevičius, R. Kačianauskas. *Baigtinių elementų metodo pagrindai*, Vilnius, Technika, 2004, 610p.

[27] R. Barauskas, *Baigtinių elementų metodo pagrindai*, KTU, „Technologija“, 1998, 375p.

[28] Kazimieras Petkevičius, *Skaitinė deformuojamų kūnų mechanika*, KTU leidykla „Technologija“, 2012, 166p.

14.TERMINŲ IR SANTRUMPŲ ŽODYNAS

Windows Phone – tai uždaro kodo, Microsoft kurta operacinė sistema, skirta mobiliems įrenginiams.

BEM – (Baigtinių elementų metodas) skaitinis metodas, leidžiantis rasti apytikslius diferencialinių lygčių dalinėmis išvestinėmis ar integralinių lygčių sprendinius,

BE – (Baigtinis elementas) sudėtingos geometrinės formos objektas, atvaizduojamas nesudėtingos formos figūromis.

3D grafika – grafika atvaizduojama 3 dimensijose.

DirectX- grafikos biblioteka Windows Phone operacinėms sistemoms.

Empirinis – paremtas patyrimu.