

VILNIAUS GEDIMINO TECHNIKOS
UNIVERSITETAS

Lina Vasiliauskiene

PRISITAIKANČIOSIOS BAIGTINIŲ ELEMENTŲ STRATEGIJOS
PLOKŠTUMINIAMS TAMPRUMO TEORIJOS UŽDAVINIAMS

Daktaro disertacija

Technologijos mokslai, mechanikos inžinerija (09T)

Vilnius 2006

Disertacija rengta 2000–2006 metais Vilniaus Gedimino technikos universitete.
Disertacija ginama eksternu.

Mokslinis konsultantas:

prof. habil. dr. Romualdas Baušys (Vilniaus Gedimino technikos universitetas,
technologijos mokslai, mechanikos inžinerija – 09T).

Turinys

PAVEIKSLŲ SĄRAŠAS.....	2
LENTELIŲ SĄRAŠAS	4
BENDRA DAKTARO DISERTACIJOS CHARAKTERISTIKA.....	5
1 . PAGRINDINIAI KOKYBIŠKAI ĮVERTINTOS BAIGTINIŲ ELEMENTŲ ANALIZĖS TAIKYMO ASPEKTAI DVIMATĖMS INŽINERINĖMS PROBLEMOMS SPREŠTI.....	9
1.1 PRISITAIKANČIOSIOS BAIGTINIŲ ELEMENTŲ STRATEGIJOS: MOKSLINIŲ DARBŲ APŽVALGA	11
1.1.1 Tinklo generavimo metodai.....	11
1.1.2 Paklaidų vertinimas.....	13
1.1.3 Tinklo optimizavimo strategijos	14
1.1.4 Singuliarių taškų įtakos sprendinio konvergavimo greičiui įvertinimas.....	16
1.2 PRISITAIKANČIOSIOS BAIGTINIŲ ELEMENTŲ STRATEGIJOS: PAGRINDINĖS LYGTYS IR ALGORITMAI	17
1.2.1 Pagrindinės tamprumo teorijos lygtys.....	19
1.2.2 Baigtinių elementų metodas.....	21
1.2.3 Pagrindinės diskretizacijos paklaidų vertinimo priklausomybės.....	22
1.2.4 h-prisitaikančiosios strategijos algoritmas.....	24
1.2.5 Plintančio fronto technikos algoritmas.....	27
1.3 IŠVADOS	30
2 . PRADINIO BEVEIK OPTIMALAUS TINKLO GENERAVIMAS SUDĖTINGIEMS DVIMAČIAMS MECHANINIAMS OBJEKTAMS.....	32
2.1 PRADINIO BEVEIK OPTIMALAUS TINKLO GENERAVIMAS	33
2.2 PRADINIO BEVEIK OPTIMALAUS TINKLO GENERAVIMO ALGORITMAS	35
2.3 MECHANINIŲ KONSTRUKCIJŲ MODELIAVIMAS TAIKANT PASIŪLYTĄ ALGORITMĄ.....	37
2.4 IŠVADOS	43
3 . GLOBALIŲJŲ BEI LOKALIŲJŲ OPTIMALUMO KRITERIJŲ TAIKYMAS PRISITAIKANČIOSE BAIGTINIŲ ELEMENTŲ TINKLO GENERAVIMO PROCEDŪROSE	44
3.1 SPRENDINIO TINKAMUMO ĮVERTINIMAS	45
3.2 GLOBALIEJI IR LOKALIEJI OPTIMALUMO KRITERIJAI.....	46
3.3 OPTIMALUMO KRITERIJŲ ĮRAUKIMAS Į BAIGTINIŲ ELEMENTŲ TINKLO GENERAVIMO PROCEDŪRĄ	48
3.4 PRAKTINIS GLOBALIŲJŲ IR LOKALIŲJŲ OPTIMALUMO KRITERIJŲ TAIKYMAS INŽINERINIŲ UŽDAVINIŲ SPRENDIMUI	50
3.4.1 Srities, sudarytos iš dviejų skirtingų medžiagų, uždavinys.....	50
3.4.2 Tempiamos stačiakampės plokštės su skylių uždavinys.....	53
3.4.3 L-formos plokštės uždavinys	55
3.5 IŠVADOS	59
4 . SINGULIARIŲ TAŠKŲ ĮTAKOS VERTINIMAS H-PRISITAIKANČIŲJŲ BAIGTINIŲ ELEMENTŲ ANALIZĖJE SPRENDŽIANT TAMPRUMO TEORIJOS UŽDAVINIUS	61
4.1 BENDRA SPRENDINIO IŠRAIŠKA SINGULIARIOSE SRITIES ZONOSE	61
4.2 POPROCESORINIS PAKLAIDOS VERTINIMAS ĮTRAUKIANT SINGULIARIOJO TAŠKO ĮTAKĄ	64
4.2.1 Superkonvergencinio elementų fragmento atkūrimo metodo panaudojimas singuliaraus sprendinio radimui.....	64
4.2.2 Praktinio algoritmo konstravimas sprendinio suradimui singuliaraus taško aplinkoje.....	66
4.3 TRADICINIO POPROCESORINIO SPRENDINIO IR PAGERINTO POPROCESORINIO SINGULIARAUS SPRENDINIO PALYGINIMAS NAGRINĖJANT L-FORMOS PLOKŠTĖS UŽDAVINĮ	69
4.4 PAKTINIS PRISITAIKANČIŲJŲ BAIGTINIŲ ELEMENTŲ STRATEGIJOS, ĮVERTINANČIOS SINGULIARIOS ZONOS ĮTAKĄ, TAIKYMAS SPRENDŽIANT PLOKŠTUMINIUS TAMPRUMO TEORIJOS UŽDAVINIUS	83
4.5 IŠVADOS	90
5 . DARBO REZULTATŲ APIBENDRINIMAS	91
6 . CITUOTOS LITERATŪROS SĄRAŠAS	93

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Tradicinis prisitaikančiųjų baigtinių elementų strategijų algoritmas	18
2.1 pav. Tradicinis kontrolinės erdvės apibrėžimas.....	33
2.2 pav. Praplėstos kontrolinės erdvės apibrėžimas	33
2.3 pav. Algoritmo schema	36
2.4 pav. Konstrukcijų A ir B, susidedančių iš dviejų skirtingų medžiagų, schemas	37
2.5 pav. Pradinio beveik optimalaus tinklo atvejis konstrukcijai A: BE tinklai ir įtempimų pasiskirstymo diagramos.....	39
2.6 pav. Tolygaus pradinio tinklo atvejis konstrukcijai A: BE tinklai ir įtempimų pasiskirstymo diagramos.....	40
2.7 pav. Pradinio beveik optimalaus tinklo atvejis konstrukcijai B: BE tinklai ir įtempimų pasiskirstymo diagramos.....	41
2.8 pav. Tolygaus pradinio tinklo atvejis konstrukcijai B: BE tinklai ir įtempimų pasiskirstymo diagramos.....	42
3.1 pav. Praplėstasis optimalaus tinklo generavimo algoritmas.....	49
3.2 pav. Dviejų medžiagų uždavinys	50
3.3 pav. Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus tinklas.....	51
3.4 pav. Galutiniai tinklai, gauti pagal tolygų pradinį tinklą: (a) – taikant optimalumo kriterijų A; (b) – taikant optimalumo kriterijų B; (c) – taikant optimalumo kriterijų C; (d) – taikant optimalumo kriterijų D.....	52
3.5 pav. Galutiniai tinklai, gauti pagal pradinį beveik optimalų tinklą: (a) – taikant optimalumo kriterijų A; (b) – taikant optimalumo kriterijų B; (c) – taikant optimalumo kriterijų C; (d) – taikant optimalumo kriterijų D.....	52
3.6 pav. Stačiakampės plokštės su skylė uždavinys	53
3.7 Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus pradinis tinklas	54
3.8 pav. Galutiniai tinklai, gauti pagal kriterijų A: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	54
3.9 pav. Galutiniai tinklai, gauti pagal kriterijų B: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	55
3.10 pav. Galutiniai tinklai, gauti pagal kriterijų C: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	55
3.11 pav. Galutiniai tinklai, gauti pagal kriterijų D: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	55
3.12 pav. L-formos plokštės uždavinys	56
3.13 pav. Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus pradinis tinklas	56
3.14 pav. Galutiniai tinklai, gauti pagal kriterijų A: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	58
3.15 pav. Galutiniai tinklai, gauti pagal kriterijų B: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	58
3.16 pav. Galutiniai tinklai, gauti pagal kriterijų C: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	58
3.17 pav. Galutiniai tinklai, gauti pagal kriterijų D: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui	59
4.1 pav. Prizminė trimatė sritis su singuliarisia briauna.....	62
4.2 pav. Plokštuminis daugiakampis su singuliariniu tašku C.....	63
4.3 pav. Singuliarinio elemento fragmento formavimas, naujo singuliarinio taško parinkimas.....	67
4.4 pav. Reguliarūs tolygūs tinklai su vienoda singuliarisia sritimi: poslinkių U_x konvergavimo grafikas.....	71
4.5 pav. Reguliarūs tolygūs tinklai su vienoda singuliarisia sritimi: poslinkių U_y konvergavimo grafikas.....	71

4.6 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: visos nagrinėjamos srities W paklaidų konvergavimo palyginimas	71
4.7 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: 150 elementų tinklas.....	72
4.8 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: 600 elementų tinklas.....	72
4.9 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: 2400 elementų tinklas.....	72
4.10 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: 9600 elementų tinklas.....	73
4.11 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_x konvergavimo grafikas.....	74
4.12 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_y konvergavimo grafikas.....	75
4.13 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: visos nagrinėjamos srities W paklaidų konvergavimo palyginimas	75
4.14 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 200 elementų tinklas	75
4.15 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 400 elementų tinklas	76
4.16 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 800 elementų tinklas	76
4.17 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 1600 elementų tinklas	76
4.18 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: poslinkių U_x konvergavimo grafikas	79
4.19 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: poslinkių U_y konvergavimo grafikas	79
4.20 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: visos nagrinėjamos srities W paklaidų konvergavimo palyginimas	79
4.21 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: paklaidų konvergavimo palyginimas srityje, apribotoje spinduliu $R=0,1$, išvestu iš singuliaraus srities taško.....	80
4.22 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 216 elementų tinklas.....	80
4.23 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 418 elementų tinklas.....	80
4.24 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 576 elementų tinklas	81
4.25 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 814 elementų tinklas.....	81
4.26 pav. Elementų skaičiaus pasiskirstymas galutiniuose optimaliuose tinkluose skaičiuojant pagal skirtingas strategijas....	84
4.27 pav. Galutiniai tinklai, gauti pagal kriterijų A naudojant singuliarų daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui.....	86
4.28 pav. Galutiniai tinklai, gauti pagal kriterijų B naudojant singuliarų daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui.....	87
4.29 pav. Galutiniai tinklai, gauti pagal kriterijų C naudojant singuliarų daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui.....	87
4.30 pav. Galutiniai tinklai, gauti pagal kriterijų D naudojant singuliarų daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui.....	87
4.31 pav. Konvergavimo greičio palyginimas.....	89

LENTELIŲ SĄRAŠAS

2.1 lentelė. Mechaninių konstrukcijų A ir B skaičiavimo rezultatai, leidžiama paklaida $h = 10\%$	39
3.1 lentelė. Dviejų medžiagų uždavinio skaičiavimo rezultatai, leidžiama paklaida $h = 10\%$	53
3.2 lentelė. Stačiakampės plokštės su skylė skaičiavimo rezultatai, leidžiama paklaida $h = 4\%$	54
3.3 lentelė. L-formos plokštės skaičiavimo rezultatai, leidžiama paklaida $h = 5\%$	57
4.1 lentelė. Reguliarių tolygių tinklų su vienoda singuliaria sritimi rezultatai.....	70
4.2 lentelė. Nereguliarių tolygių tinklų su vienoda singuliaria sritimi skaičiavimo rezultatai	74
4.3 lentelė. Nereguliarių tolygių tinklų su kintama singuliaria sritimi poslinkių reikšmės.....	82
4.4 lentelė. Nereguliarių tolygių tinklų su kintama singuliaria sritimi paklaidų reikšmės	82
4.5 lentelė. L-formos plokštės skaičiavimo rezultatai, $h = 5\%$	85

BENDRA DAKTARO DISERTACIJOS CHARAKTERISTIKA

Aktualumas. Ištobulėjusi kompiuterinė technika bei skaitmeninius metodus naudojančios kompiuterinės programos atveria plačias galimybes kurti patikimus inžinerinių konstrukcijų projektus. Tyrinėjant realius projektus reikalingos didelės kompiuterio laiko sąnaudos, todėl ypač svarbu greitai rasti kokybiškai įvertintus projektų sprendinius. Tačiau dažniausiai inžinerinio projektavimo srityje susiduriama su tokiais uždaviniais, kurių tikslus sprendinys nėra žinomas ir tokiems uždaviniams taikomi apytiksliai skaičiavimo metodai, randantys ne tikslų, o artutinį sprendinį.

Jau keletą dešimtmečių baigtinių elementų metodas yra vyraujanti inžinerinių konstrukcijų skaičiavimo technologija. Vienas iš šio metodo sunkumų yra neišreikštinė skaičiavimo rezultatų priklausomybė nuo pasirinkto baigtinių elementų tinklo. Nepaisant didelio kiekio šios srities darbų, tinklo generavimo klausimas dar nėra išsemtas.

Projektuotojui labai svarbu gauti patikimus skaičiavimo rezultatus ir įvertinti skaičiavimo paklaidas kiekvienu konkrečiu skaičiavimo atveju, todėl automatizuotas baigtinių elementų metodo diskretizacijos paklaidų nustatymas yra ne mažiau svarbus šiuolaikinės analizės uždavinys. Kokybiškai įvertinti baigtinių elementų sprendiniai gaunami taikant prisitaikančiąsias baigtinių elementų strategijas. Šie metodai yra visiškai automatizuoti, iteracinis procesas valdomas vartotojo nustatyta sprendinio paklaida bei remiasi įvairiomis paklaidų vertinimo procedūromis. Taikant šias strategijas, rasti kokybiškai įvertinti sprendiniai jau pakankamai patikimai aprašo nagrinėjamus projektus esant įvairiems išoriniams poveikiams.

Prisitaikančiųjų baigtinių elementų strategijų taikymas ir tobulinimas nagrinėjant mechaninių konstrukcijų būvį yra populiarus skaičiuojamosios mechanikos bei mechanikos inžinerijos uždavinys, praplečiantis skaitinių eksperimentų galimybes.

Tikslas ir uždaviniai. Darbo tikslas – sukurti sudėtingos geometrijos mechaninių konstrukcijų diskretizavimo baigtiniais elementais metodiką taikant prisitaikančiąsias strategijas. Tikslui pasiekti disertacijoje nagrinėjami šie uždaviniai:

1. Sukurti racionalią procedūrą pradiniam beveik optimaliam baigtinių elementų (BE) tinklui generuoti.
2. Atlikti prisitaikančiųjų BE strategijų, paremtų lokaliaisiais ir globaliaisiais optimalumo kriterijais, skaitinę realizaciją bei analizę:
 - a) Išanalizuoti globaliųjų bei lokaliųjų paklaidos vertinimo kriterijų įtaką sprendinio konvergavimo greičiui, elementų pasiskirstymui galutiniame optimaliame BE tinkle bei įvertinti elementų skaičiaus priklausomybę galutiniame tinkle nuo pasirinkto optimalumo kriterijaus.
 - b) Įvertinti globaliųjų bei lokaliųjų paklaidų vertinimo kriterijų veikimą skirtingomis pasirinktos konstrukcijos darbo sąlygomis (skirtingų medžiagų sąlytis, sudėtingos geometrijos atvejais).

3. Sukurti sprendinio pagerinimo singuliaraus taško aplinkoje metodą sudėtingos geometrijos uždavinių sprendimui.

Tyrimo objektas ir metodai. Darbo objektas – mechaninių struktūrų modeliavimo metodai. Pagrindiniu tyrimo metodu pasirinktas baigtinių elementų metodas. Baigtinių elementų tinklui generuoti taikomas plintančio fronto technikos (PFT) algoritmas (angl. *Advancing Front Technique*).

Naujumas ir originalumas. Šioje disertacijoje siekiama greitai ir be didelių skaičiavimo sąnaudų rasti pakankamai optimalų BE tinklą, kuriame būtų tenkinami tiek globaliosios, tiek lokalsios paklaidos reikalavimai, ypač daug dėmesio skiriant tokiems paklaidų vertinimo kriterijams formuluoti ir tirti. Todėl remiantis baigtinių elementų metodu bei kompiuterinėmis technologijomis, sukurtos naujos strategijos, susidedančios iš šių komponentų:

1. Sukurta nauja strategija pradiniam beveik optimaliam BE tinklui generuoti. Tikslui pasiekti pasiūlyta taikyti naujus *patobulintos kontrolinės erdvės* bei *kontrolinės sferos* apibrėžimus, leidžiančius geriau apibūdinti nagrinėjamos konstrukcijos topologinius bei fizinius aspektus.
2. Derinant pradinio beveik optimalaus BE tinklo generavimo procedūrą su įvairiais optimalumo kriterijais, gauta nauja metodologija, leidžianti sumažinti iteracijų, reikalingų optimaliam tinklui gauti, skaičių, bei leidžianti vartotojui savo nuožiūra reguliuoti BE skaičių bei tankumą galutiniame tinkle.
3. Išnagrinėjus esamus sprendinio skaičiavimo singuliaraus taško aplinkoje metodus, pasiūlytas naujas sprendinio pagerinimo singuliariose zonose metodas *h*-prisitaikančiosioms baigtinių elementų strategijoms.

Teorinė ir praktinė reikšmė. Taikant pasiūlytą pradinio BE tinklo generavimo metodą, galima iš anksto, dar neapskaičiavus sprendinio ir neatlikus paklaidų analizės, keisti elementų skaičių ir tankumą tam tikrose mus dominančiose konstrukcijos vietose. Toks patobulintas pradinis beveik optimalus tinklas prisitaikančiųjų strategijų procese leidžia sumažinti iteracijų skaičių, reikalingą galutiniam optimaliam tinklui gauti. Atlikus įvairių optimalumo kriterijų analizę, galima konstatuoti, kad elementų skaičius bei tankumas galutiniame tinkle tiesiogiai priklauso nuo pasirinkto optimalumo kriterijaus. Todėl derinant pradinio beveik optimalaus BE tinklo generavimo procedūrą su įvairiais optimalumo kriterijais, įtaka prisitaikančiųjų strategijų procesui daroma dviem būdais – iteracijų skaičius sumažinamas naudojant pradinį beveik optimalų tinklą, o elementų skaičius minimizuojamas parenkant atitinkamą optimalumo kriterijų. Išnagrinėjus sprendinio skaičiavimo singuliaraus taško aplinkoje metodus, pasiūlytas naujas metodas, leidžiantis kiekvienoje prisitaikančiųjų baigtinių elementų strategijos iteracijoje gauti tikslesnį sprendinį ir dar labiau sumažinti skaičiavimo sąnaudas, reikalingas sudėtingos geometrijos uždaviniui išspręsti.

Nagrinėjant prisitaikančiųjų baigtinių strategijų skaitinius pavyzdžius, buvo naudoti ir integruoti tarpusavyje trys programiniai paketai. Pradinio beveik optimalaus tinklo generavimo algoritmas

L. Vasiliauskienė. Pristaikančiosios baigtinių elementų strategijos plokštuminiams tamprumo teorijos uždaviniams

realizuotas atskiru programiniu vienetu „Microsoft Visual Fortan 5.0“ programavimo aplinkoje, nagrinėtų pavyzdžių baigtinių elementų analizės atlikimui parinktas „ANSYS 5.6“ programinis paketas, o diskretizacijos paklaidų vertinimas bei optimalumo kriterijų analizė atlikta bei realizuota atskiru modulių naudojant „Microsoft Visual C++ 6.0“ programinę aplinką.

Daktaro disertacijos darbe teikiami ginti šie teiginiai:

- Pradinio beveik optimalaus tinklo generavimo metodika.
- Adaptyvios baigtinių elementų strategijos su lokaliaisiais ir globaliaisiais tinklo optimalumo kriterijais.
- Poprocesorinio sprendinio radimo metodika įvertinant geometrinį singuliarumą.

Darbo rezultatų aprobavimas. Disertacijos tyrimų medžiaga pristatyta šiose Lietuvos tarptautinėse konferencijose:

1. „*Modern Building Materials, Structures and Techniques*“, Vilnius, 2001 m. gegužės 16–18d. p. 301, pranešimo tema „Intelligent initial finite element mesh generation for solution of engineering problems“.
2. „*Mathematical Modelling and Analysis*“, Vilnius, 2001 m. gegužės 31–birželio 2 d., p. 87, pranešimo tema „Sub-optimal finite element mesh generation for 2D problems“.
3. „*Mechanika 2003*“, Kaunas, 2003 m. balandžio 3–4 d., pranešimo tekstas išspausdintas atskiru leidiniu, pranešimo tema „Adaptive finite element analysis based on different mesh optimality criteria“.
4. „*Mathematical Modelling and Analysis*“, Trakai, 2003 m. gegužės 28–31d., p. 80, pranešimo tema „Adaptive mesh refinement procedure based on different mesh optimality criteria“.
5. „*Mechatronic Systems and Materials 2005*“, Vilnius, 2005 m., spalio 20–23 d., p. 62, pranešimo tema „Mesh optimization strategies for solutions of engineering problems“.

Disertacijos tyrimų medžiaga pristatyta Lietuvos konferencijoje:

6. „*XII Lietuvos skaičiuojamosios mechanikos asociacijos*“ konferencija, Vilnius, 2004 m., balandžio 9d., pranešimo tema „Prisitaikančiųjų baigtinių elementų strategijos pagerinimas plokštuminių tamprumo teorijos uždavinių sprendimui“.

Darbo rezultatų publikavimas. Disertacijos tyrimų medžiaga išspausdinta Lietuvos mokslo žurnaluose ir testiniuose leidiniuose su svorio koeficientu (ISI):

1. Vasiliauskienė L. ir Baušys R. (2002) Intelligent Initial Finite Element Mesh Generation for Solutions of 2D problems // *Informatica*, 13(2), p. 239–250.
2. Vasiliauskienė L. ir Baušys R. (2006) Mesh Optimization Strategies for Solution of Engineering Problems // *Solid State Phenomena*, 113, p. 157–162.

Referuotoje tarptautinių mokslo organizacijų organizuotų konferencijų, vykusių Lietuvoje, pranešimų medžiagoje:

3. Vasiliauskienė L., Baušys R. (2003). Adaptive Finite Element Analysis Based on Different Mesh Optimality Criteria // Konferencijos „Mechanika–2003“ pranešimų leidinys, p. 133–138.

Lietuvos mokslo žurnaluose, referuojamuose pripažintose tarptautinėse duomenų bazėse:

4. Vasiliauskienė L., Baušys R. (2003) Intelligent Adaptive Mesh refinement procedure based on global and local mesh optimality criteria // *Mechanika*, 3(41), p. 11–17.

Disertacijos sandara. Įvadiniame skyriuje trumpai apibrėžta tyrimų sritis, suformuluotas darbo aktualumas, tikslas, apibūdintas tyrimų objektas, aptartas darbo naujumas ir originalumas.

Pirmajame skyriuje apžvelgiami tokie pagrindiniai mechaninių konstrukcijų modeliavimo taikant prisitaikančiųjų baigtinių elementų analizę, kaip baigtinių elementų tinklo generavimo metodai, paklaidų vertinimo būdai, tinklo optimizavimo strategijos bei sprendinio tikslinimo būdai singuliaruose taškuose. Trumpai pateikiama prisitaikančiųjų baigtinių elementų strategijų vystimosi istorija, apžvelgiami kitų autorių darbai. Taip pat pateikiamos pagrindinės tamprumo teorijos lygtys, smulkiau aprašomas klasikinis prisitaikančiųjų baigtinių elementų strategijos algoritmas, pateikiamos paklaidų vertinimo priklausomybės, h -prisitaikančiųjų baigtinių elementų strategijos algoritmas bei plintančio fronto technikos algoritmas, skirtas baigtinių elementų tinklui generuoti.

Antrajame skyriuje pateikiama nauja strategija, skirta sudėtingų mechaninių konstrukcijų pradiniam beveik optimaliam tinklui generuoti. Taip pat pateikiami skaitiniai pavyzdžiai, leidžiantys iliustruoti pasiūlyto metodo pranašumą lyginant su tradiciniu pradinio baigtinių elementų tinklo generavimo būdu.

Trečiajame skyriuje išnagrinėti dažniausiai mechanikos uždavinių sprendimui taikomi lokalieji ir globalieji tinklo optimalumo kriterijai. Taip pat išanalizuota skirtingų optimalumo kriterijų įtaka prisitaikančiųjų baigtinių elementų analizės procesui. Pateikti trys skirtingo sudėtingumo inžineriniai uždaviniai, leidžiantys geriau suprasti vieno ar kito optimalumo kriterijaus parinkimo įtaką galutinio optimalaus tinklo gavimo procesui.

Ketvirtajame skyriuje pateiktas naujas metodas, leidžiantis tiksliau aproksimuoti sprendinį nagrinėjant sudėtingas mechanines konstrukcijas su singuliariomis zonomis. Patikslinto sprendinio singuliarame taške pranašumas iliustruojamas lyginant jį su baigtinių elementų sprendiniu, taip pat įkomponuojant patikslinto singuliarus sprendinio skaičiavimo metodą į prisitaikančiųjų baigtinių elementų strategiją.

Darbas baigiamas pateikiant gautas išvadas bei išvardijant cituotos literatūros sąrašą. Disertacijos tekstas pateiktas 99 kompiuteriu sumaketuotuose 1.5 eilėtarpiu išspausdintuose puslapiuose. Tekste yra 57 paveikslai, 9 lentelių ir 129 cituotos literatūros šaltinių. Disertacija parašyta lietuvių kalba.

1. PAGRINDINIAI KOKYBIŠKAI ĮVERTINTOS BAIGTINIŲ ELEMENTŲ ANALIZĖS TAIKYMO ASPEKTAI DVIMATĖMS INŽINERINĖMS PROBLEMOMS SPRĘSTI

Baigtinių elementų metodas yra labiausiai paplitęs matematinių modelių skaitmeninio skaičiavimo įrankis, plačiai taikomas visose srityse, kur uždaviniai aprašomi diferencialinėmis lygtimis, o ypač stipriai įsitvirtino struktūrinės inžinerijos bei kitose panašiose disciplinose. Taikant šį metodą sprendžiami mechaniniai, šiluminiai, hidrauliniai, elektromagnetiniai ir kitokių fizinių sistemų uždaviniai, taip pat labai sėkmingai modeliuojami dinaminiai procesai. Baigtinių elementų metodo algoritmas yra universalus sprendžiant įvairių fizinių sistemų uždavinius, todėl nebereikia gilintis į daugelį tradicinių analitinių lygčių sprendimo būdų, kurių didžiausias trūkumas yra tas, kad jie dažnai yra specifiniai kiekvienai atskirai uždavinių klasei.

Nagrinėjant bet kokią vientisą kūną, laisvės laipsnių skaičius yra begalinis, be to, tikslus sprendinys dažniausiai nėra žinomas, galima rasti tik apytikrį sprendinį, kuris skaičiuojamas padarius prielaidą, kad vientiso kūno būvis gali būti aprašomas baigtiniu nežinomųjų skaičiumi. Baigtinių elementų metodo esmė – nagrinėjamos konstrukcijos suskaidymas į tokias smulkesnes dalis, kurioms galime pritaikyti standartinius skaičiavimus. Taip nuo begalinio laisvės laipsnių skaičiaus pereinama prie baigtinio. Nagrinėjamos srities skaidymas į žinomo dydžio ir formos dalis, vadinamas srities *diskretizacija*, o gautoji dalelė – *baigtiniu elementu*.

Baigtinių elementų metodo raidos istorija prasideda šeštajame dešimtmetyje. Vieną iš pirmų straipsnių 1956 metais su bendraautorais parašė Turner (Turner ir kt., 1956), o besibaigiant septintajam dešimtmečiui pasirodė pirmosios knygos, skirtos baigtinių elementų metodui. Didelę įtaką baigtinių elementų metodo raidai turėjo Zienkiewicz (Zienkiewicz, 1967) darbai. Tačiau reikėtų neužmiršti, kad dėl silpnų to meto kompiuterinių galimybių sprendžiamų uždavinių apimtis buvo labai ribota, todėl tinklo generavimo technikoms bei algoritmams buvo skiriama palyginti mažai dėmesio. Sprendžiant tuo metu aktualius uždavinius tiesiog buvo stengiamasi kaip galima labiau sumažinti tinklo elementų skaičių, kad baigtinių elementų uždavinį galėtų išspręsti tuo metu prieinami ribotos atminties bei greičio kompiuteriai. Aštuntajame dešimtmetyje situacija pasikeitė – dėl per didelės sprendžiamo uždavinio dydžio apimties pagrindiniu tikslu tapo automatinio tinklo generavimo algoritmo kūrimas bei tobulinimas. Būtent dėl šios priežasties to meto straipsniuose bei knygose daugiausia dėmesio buvo skiriama ir tinklo generavimo algoritmams (Irons, Ahmad, 1980; Hinton, Owen, 1979; Bathe, 1978; Desai, Abel, 1972). Devintajame dešimtmetyje skaičiavimo stočių atmintis buvo jau pakankamai didelė, kad galėtų išspręsti sudėtingus uždavinius, tačiau asmeninių kompiuterių pajėgumai dar buvo riboti, todėl įvairūs uždavinių sprendimo pagreitinimo algoritmai išliko vis dar aktualūs. Taip pat reikėtų paminėti, kad kaip tik tuo laikotarpiu buvo automatizuotos paklaidų vertinimo procedūros taikant poprocesorinius paklaidų vertinimo metodus. Dešimtajame dešimtmetyje tinklo generavimas tapo vienu iš svarbesnių baigtinių elementų analizės priešprocesorinių žingsnių. Baigtinių elementų analizės automatizavimas

naudojant diskretizacijos paklaidos vertinimą bei prisitaikančiuosius tinklus kėlė vis didesnį susidomėjimą tiek gamyboje, tiek moksle (Akin, 1994), todėl ypač daug mokslininkų dirbo prisitaikančiųjų strategijų kūrimo bei tobulinimo srityje (Szabo, Babuška, 1991; Lee, Hoobs, 1999; Stupak, Baušys, 2000; Dobrowolski, Villegas, 2005; Belhachmi ir kt., 2005).

Lietuvos mokslininkai taip pat dirba skaitinio modeliavimo ir optimizavimo srityje siekdami išanalizuoti įvairias mechanikos problemas. Skaičiuojamosios mechanikos, tampriųjų-plastinių konstrukcijų prisitaikomumo teoriją bei konstrukcijų optimizacijos bei matematinio programavimo metodus statybinėje mechanikoje nagrinėjo J. Atkočiūnas su bendraautorais (Atkočiūnas, Nagevičius, 2004; Atkočiūnas, Merkevičiūtė, 2004; Atkočiūnas, Jarmolajeva ir Merkevičiūtė, 2004). Baigtinių elementų metodą plačiai nagrinėjo ir savo moksliniuose tyrimuose taikė R. Barauskas su bendraautorais (Barauskas, 1998; Barauskas, Kuprys, Leonavičiūtė, 2003; Barauskas, Kačianauskas, Belevičius, 2004). Baigtinių elementų metodo taikymą konstrukcinių elementų irimui tyrė R. Kačianauskas (Stupak, Kačianauskas, 2000; Kačianauskas, Markauskas, 2004), o prisitaikančiąsias baigtinių elementų strategijas bei jų taikymą įvairių mechanikos uždavinių sprendimui analizavo R. Baušys (Baušys, Wiberg, 1999). Reiktų paminėti, kad labai sėkmingai prisitaikančiąsias BE strategijas dvimačių struktūrų irimo modeliavimui pritaikė E. Stupak (2004), o prisitaikančiąsias BE strategijas erdvinio atveju plėtojo ir moksliniuose tyrimuose taikė V. Žarnovskij (2005).

Prisitaikančiųjų strategijų tikslas yra kaip galima labiau sumažinti skaičiavimo sąnaudas, reikalingas apytiksliam sprendiniui rasti, kai žinoma iš anksto nustatyta leistina paklaida. Prisitaikančiųjų baigtinių elementų strategijų metodai susideda iš tokių pagrindinių problemų:

1. Baigtinių elementų tinklo generavimo algoritmo parinkimas prisitaikančiųjų baigtinių elementų strategijų algoritmui.
2. Diskretizacijos paklaidų nustatymas ir įvertinimas, kurio esmę sudaro naujo aukštesnės tikslumo klasės sprendinio suradimas, nes tikslus uždavinio sprendinys dažniausiai nėra žinomas visai.
3. Naujo prisitaikančiojo tinklo generavimas arba tinklo optimizacija, leidžianti išsaugoti optimalų diskretizacijos paklaidų vertinimo parametrų pasiskirstymą.
4. Tinklo elementų kokybės vertinimo bei gerinimo procedūros, leidžiančios generuoti baigtinių elementų tinklą esant minimaliam prastos formos elementų skaičiui.

Šiuolaikinės kompiuterinės technologijos labai supaprastina sudėtingų inžinerinių konstrukcijų projektavimo galimybes. Šios technologijos sudaro sąlygas visapusiškai modeliuoti konstrukcijų būvį be prototipų bandymo, kuriems gaminti bei testuoti reikia didelių materialinių bei laiko sąnaudų. Baigtinių elementų metodo pagrindu yra sukurta daug universalių bei specializuotų kompiuterinių programų. Programos „ANSYS“ (Swanson Analysis Systems Inc., Houston), „SC/Probe“ (MacNeal-Schwendler Corp., Los Angeles), „I-DEAS“ (Structural Dynamic Research Corp., Milford) ir pan. leidžia greitai ir patikimai ištirti projektuojamas sistemas ir dar esant projektavimo stadijai patikrinti ir prognozuoti jų

L. Vasiliauskienė. Pristatantiosios baigtinių elementų strategijos plokštuminiam tamprumo teorijos uždaviniams

savybes. Taip pat beveik kiekvienoje techninės pakraipos mokymo įstaigoje yra sukurta mažesnių, specializuotų baigtinių elementų programų, naudojamų atliekant mokslinius tyrimus bei sprendžiant taikomuosius inžinerinius uždavinius. Drauge su tokiais automatizuoto geometrinio projektavimo programomis, kaip AutoCAD, CATIA ir pan., tokiais aukšto lygio programavimo kalbomis kaip FORTRAN, Pascal, C bei su tokiais matematinei analizei orientuotomis programavimo terpėmis kaip MATLAB, MATCAD, MAPLE, baigtinių elementų programos sudaro galingą įrankį, skirtą inžinerinėms sistemoms geometriškai bei fiziškai modeliuoti.

1.1 Pristaikančiosios baigtinių elementų strategijos: mokslinių darbų apžvalga

1.1.1 Tinklo generavimo metodai

Sprendžiant vieną ar kitą uždavinį, sunku iš anksto pasakyti, kurį baigtinių elementų tinklo generavimo metodą taikyti. Paprastai pati uždavinio sąlyga pasako, kuris metodas labiau tiktų vienam ar kitam atvejui. Yra išskiriamos dvi technikos – struktūrinių ir nestruktūrinių tinklų generavimas. Sakoma, kad tinklas yra struktūrinis, jei kiekvieną jo mazgą supa tas pats elementų skaičius. Kai jo mazgą supančių elementų skaičius kinta, tai tinklas yra nestruktūrinis.

Struktūriniai tinklai pradėti naudoti dar šeštajame dešimtmetyje, kai tuo tarpu nestruktūriniai tinklai pasirodė tik aštuntajame dešimtmetyje. Išsamią struktūrinių tinklų apžvalgą padarė Frey ir George (2000). Visus struktūrinius tinklus sieja bendra idėja – parenkama paprastos geometrijos sritis (parametrinė), ji sudalijama, o paskui padarius atitinkamas transformacijas ar atvaizdžius į tikrą nagrinėjamą sritį, gaunamas tos srities tinklas. Tokių transformacijų pavyzdys gali būti Bezier kreivės, B-splainai, Lagrandžo daugianariai ir t.t. Tokiu būdu gautų tinklų elementai dažniausiai turi gerą formą, kuri sąlygoja mažesnes paklaidas atliekant tolimesnius skaičiavimus. Tačiau toks metodas tinka tik aiškią geometriją turinčioms sritims, o jei sritis yra kiek sudėtingesnė ir joje esama skylių, įpjovų ar kitokių geometrinių nesuderinamumų, struktūrinių tinklų generavimas tampa beveik neįmanomas, nes iš vartotojo reikalauja daug rankinio darbo dalijant šią sritį į standartines geometrines sritis.

Kadangi realiame gyvenime nagrinėjamos sritys turi gan sudėtingą topologiją bei kraštines sąlygas, tai neišvengiamai baigtinių elementų tinklo generavimas turi būti kaip galima labiau automatizuotas, o vartotojo įsikišimas apribotas tik srities geometrijos bei kraštinių sąlygų nustatymu. Rekursinės erdvės dekompozicijos, Delaunay trinaguliacijos bei plintančio fronto technika – yra trys labiausiai paplitę nestruktūriniai tinklo generavimo metodai. Visi šie trys metodai susideda iš keturių pagrindinių žingsnių – srities geometrijos apibrėžimo, elemento dydžio funkcijos nustatymo, tinklo generavimo bei tinklo kokybės gerinimo. Išsamų rekursinės erdvės dekompozicijos, Delaunay trinaguliacijos bei plintančio fronto technikos algoritmų taikymą matematiniam programavimui galima rasti Topping su bendraautorais (Topping ir kt., 2004) parašytoje knygoje.

Rekursinis erdvės dekompozicijos metodas buvo pristatytas kaip vienas iš būdų automatiškai aproksimuoti įvairius geometrinius objektus (Frykestig, 1994; Samet 1984; Yerry, 1983). Pasirodė, kad šis metodas yra pakankamai geras ir patikimas, kad susidorotų su jam keliamais automatinio tinklo reikalavimais (Kela, 1986). Taikant šį metodą, nagrinėjama sritis padengiama įvairaus dydžio celėmis, reiškiančioms tos srities sudalijimą. Šios celės gaunamos rekursiškai dalijant šakninę celę tol, kol sritis visiškai ir tiksliai padengiama smulkesnėmis celėmis. Paskui kiekviena gauta celė padengiama baigtinių elementų tinklu, o tų tinklų sąjunga duoda galutinį srities tinklą. Šio metodo pagrindinė idėja yra ta, kad kuo smulkiau sudalijama sritis, tuo paprastesnė gaunama kiekvienos celės geometrija, tuo lengviau sugeneruojamas baigtinių elementų tinklas. Šiuo metodu sugeneruotos celės kartu nusako ir generuojamų baigtinių elementų tinklo elementų dydį. Paprastai tinklo, gauto taikant rekursinės erdvės sudalijimo metodą, kokybė yra pakankamai aukšta, tačiau nors elementai ir atitinką jiems numatytą dydį, jie gana dažnai dar turi netaisyklingą formą srities kraštuose. Dėl šios priežasties įvairios tinklo tobulinimo procedūros dar taikomos galutiniam tinklui gauti.

Delaunay trianguliacija bei su ja susijusios sritys vis dar yra viena iš svarbesnių skaičiuojamosios geometrijos dalykų. Patį pirmąjį straipsnį dar 1934 metais parašė pats Delaunay (Delaunay, 1934). Nuo to laiko susidomėjimas šiuo metodu tik didėja, nes jis leidžia lanksčiai ir efektyviai sugeneruoti baigtinių elementų tinklą (Preparata, 1985; Joe, 1991; El-Hamalawi, 2004; Linardakis, Chrisochoides, 2006). Delaunay trianguliacijos sėkmės priežastis yra gana lengva metodo realizacija, be to, šis metodas yra labai efektyvus lyginant su kitais metodais. Tačiau Delaunay trianguliacija neišsaugo tikslių srities kraštų, todėl gautiniame tinkle gali atsirasti blogos formos elementų, turinčių neigiamos įtakos tolimesniems skaičiavimams.

Trečiasis nestruktūrinių tinklų generavimo metodas – plintančio fronto technika – kaip ir prieš tai minėtieji metodai, yra labai efektyvus ir generuoja aukštos kokybės elementus visoje nagrinėjamoje srityje, tačiau priešingai, nei kiti paminėti metodai, PFT dar papildomai išsaugo srities kraštų vientisumą. Dėl šių savybių PFT metodas pasirinktas disertacijoje nagrinėjamų uždavinių tinklui generuoti.

Plintančio fronto techniką 2D atveju pirmasis pristatė George 1971 metais, o klasikinį PFT algoritmą jo dabartine forma pirmasis aprašė Lo (1985) ir Peraire (1987). Skaitmeniniu būdu patobulintą PFT metodą pristatė Löhner ir Parikh (1988), kiek vėliau – Frykestig (1994). Klasikinis plintančio fronto technikos algoritmas susideda iš tokių pagrindinių žingsnių – visų pirma inicializuojamas bei pagal tam tikras taisykles surūšiuojamas srities frontas ir nustatoma elemento dydžio funkcija. Paskui pagal iš anksto numatytą kriterijų atrenkamas srities fronto elementas bei atitinkamas optimalus taškas, reikalingas naujam baigtiniam elementui formuoti. Jei tinklo elementas buvo surastas teisingai, papildoma tinklo duomenų struktūra bei fronto elementai. Procesas tęsiamas tol, kol fronte liks bent vienas elementas. Jei būtina, baigus generuoti tinklą, galima gautą tinklą papildomai optimizuoti.

Didėjant prisitaikančiųjų strategijų populiarumui, daug dėmesio buvo skiriama automatinio, efektyvaus ir patikimo tinklo generatoriaus paieškoms (Frykestyg, 1994; George, Seveno, 1994; Lohner, 1996), kurių metu įrodyta, kad plintančio fronto technikos algoritmas atitinka visus geram tinklo generatoriui keliamus reikalavimus ir prisitaikančiųjų strategijų algoritmui tinka geriausiai.

1.1.2 Paklaidų vertinimas

Kadangi baigtinių elementų metodas yra aproksimacinis skaičiavimo metodas, tai neišvengiamai susiduriame su įvairios rūšies paklaidomis, atsirandančiomis dėl matematinio modeliavimo, baigtinių elementų formulavimo ar srities diskretizacijos netikslumų. Srities diskretizacijos paklaidos – tai turimos srities sudalijimo į smulkesnes dalis pagal tam tikras taisykles rezultatas. Diskretizacijos paklaidos apibrėžiamos kaip tikslaus ir baigtinių elementų sprendinio skirtumas, tačiau tik labai retais atvejais yra žinomas tikslus uždavinio sprendinys, todėl diskretizacijos paklaidų vertinimas ir suradimas yra viena iš svarbiausių baigtinių elementų analizės proceso dalių.

Diskretizacijos paklaidų vertinimas skirstomas į priešprocesorinius ir poprocesorinius paklaidų vertinimus. Priešprocesoriniai paklaidų įverčiai paprastai pateikia tik kokybinį sprendinio ir baigtinių elementų tinklo įvertinimą, pvz., tokį kaip tinklo reguliarumas, asimptotinis konvergavimo greitis ar sprendinio tolydumas. Poprocesoriniai paklaidų įverčiai naudojami specifinei informacijai apie įvairių matavimų paklaidas gauti. Poprocesorinio paklaidų vertinimo tikslas yra nustatyti, kaip tiksliai baigtinių elementų sprendinys aproksimuoja tikslų pateiktojo uždavinio sprendinį, naudojant jau suskaičiuotus duomenis (Bathe, 1996). Diskretizacijos paklaidos gali būti įvertintos naudojant įvairias paklaidų vertinimo procedūras, pavyzdžiui, interpoliacinius paklaidų vertinimo metodus, ekstrapoliacinius paklaidų vertinimo metodus, rezidiumų metodus bei poprocesorinius paklaidų vertinimo metodus.

Šiame darbe bus taikomi tik *poprocesoriniai* paklaidų vertinimo metodai, kurių esmę sudaro tikslaus sprendinio pakeitimas poprocesoriniu sprendiniu. Tikslesnis poprocesorinis sprendinių laukas gaunamas iš originalaus baigtinių elementų sprendinio naudojant poprocesorines technologijas, dar vadinamas atkūrimo procedūromis. Šio tipo paklaidų vertinimo metodų patikimumui turi įtakos atkurto sprendinio kokybė, o tikslumas priklauso nuo asimptotinių poprocesorinio sprendinio savybių. Norint gauti kaip galima tikslesnius paklaidos įverčius, patikslintas sprendinys turi būti aukštesnio konvergavimo laipsnio, lyginant su baigtinių elementų sprendiniu. Bendruoju atveju gauti baigtinio elemento įtempiai yra tik iš dalies nenutrūkstantys, kai tikslus sprendinys yra glodusis visoje nagrinėjamoje srityje, todėl norint iš diskrečiojo gauti logišką ir glodųjį sprendinį, naudojamos poprocesorinės procedūros. Vadinasi, poprocesorinės procedūros gali būti naudojamos dviem atvejais: pirmiausia jos padeda sujungti trūkiuosius baigtinių elementų sprendinius į bendrą vientisą sprendinį, paskui prireikus gautas poprocesorinis sprendinys gali būti naudojamas įvertinant diskretizacijos paklaidą poprocesoriu paklaidos vertinimo metodu.

Klasikinė ir seniausiai naudojama poprocesorinė procedūra yra paprasčiausias vidurkinimas. Ji atsirado kartu su pačiu baigtinių elementų metodu. Kadangi baigtinių elementų įtempių sprendinys yra trūkus elemento kraštuose, todėl vientisam įtempių laukui gauti ir yra naudojama paprasto vidurkinimo procedūra. Nemažai tyrimų buvo atlikta siekiant tobulinti poprocesorines procedūras (Braack, Ern, 2003; Chatzipantelidis, Lazarov, 2005). Vidurkinimo techniką nagrinėjo Bramble ir Shatz (1977), globaliąją L_2 projekciją pasiūlė Oden ir Brauchli (1971). Didelis lūžis plėtojant poprocesorines procedūras įvyko Zienkiewicz ir Zhu (1992) pristatius superkonvergencinį elementų fragmento atkūrimo metodą (SPR). Metodo esmę sudaro lokaliųjų elementų fragmentų bei superkonvergencinių taškų, kuriuose baigtinių elementų sprendinys įgyja tiksliausią reikšmę, nagrinėjimas. Taikant mažiausiųjų kvadratų metodą surandamas naujas aukštesnės tikslumo klasės baigtinių elementų kintamųjų laukas. Šį metodą sėkmingai patobulino Wiberg su bendraautorais (1993, 1994), Aalto (1997), Kvamsdal ir Okstad (1998), Baušys su Wiberg (2000) ir Selna ir Hakam (2004), Gu su bendraautorais (2004), Cai ir Starke (2004), Bochev ir Grunzburger (2006).

1.1.3 Tinklo optimizavimo strategijos

Kad ir koks efektyvus būtų baigtinių elementų tinklo generavimo algoritmas, atsirastų tokių uždavinių, kai tam tikras elementų skaičius netenkins formai ir dydžiui keliamų reikalavimų. Tokiais atvejais elementų kokybei pagerinti reikalingas papildomas žingsnis – vadinamoji tinklo optimizacija. Norint diskretizuoti sudėtingos geometrijos sritį ir išvengti labai didelio skaičiaus elementų, reikia naudoti skirtingo dydžio elementus skirtingoms srities vietoms aproksimuoti. Nemažai autorių nagrinėjo elementų dydžio kontrolės problemas (Owen, 1992; Owen, Saigal, 2000). Paprastai elemento dydžio kontrolės funkcijos pristatomos kartu su tam tikru tinklo generavimo algoritmu. Vienas iš populiariesnių elementų dydžio nusakymo būdų yra grubaus pradinio tinklo konstravimas iš nagrinėjamos srities viršūnių. Pavyzdžiui, Delaunay trianguliacijos algoritme elemento dydį kiekvienoje srities viršūnėje gali nurodyti pats vartotojas, taip pat elemento dydis gali būti apskaičiuojamas remiantis tam tikromis prielaidomis, padarytomis iš srities geometrijos ypatumų ar naudojantis paklaidos normomis iš prieš tai buvusio tinklo. Vieni iš paskutiniųjų Borouchaki ir bendra autorių (Borouchaki ir kt., 1998) darbų pristato modifikuotą Delaunay algoritmą, kuriame naudojamas interpoliuotas tolygus elementų dydžio tenzorių laukas tiek plokštumoje, tiek trimačiu atveju. Taip pat nagrinėjamos galimybės naudoti paklaidų normas iš prieš tai buvusio sprendinio naujam elementų dydžiui perskaičiuoti. Pristatydami prisitaikančiųjų baigtinių elementų strategiją Peraire su bendraautorais (Peraire ir kt., 1987, 1988) apibrėžė kartu ir elemento dydžio nusakymo funkciją plintančio fronto technikos algoritmui. Nemažą įtaką plintančio fronto technikos algoritmo elemento dydžiui nustatyti parodė R.Löhner ir P.Parikh (Löhner, 1988; Löhner, Parikh, 1988), o kiek vėliau – J.Frykestig (Frykestig, 1994).

Prisitaikančiosios baigtinių elementų strategijos gali būti realizuotos įvairiais būdais, tačiau apibendrinant galima išskirti keturias pagrindines diskretizacijos paklaidų mažinimo strategijas:

1. *H-versija* – tai adaptyvusis procesas, kurio metu yra keičiamas elemento dydis, o elementą aprašančio daugianario eilė nekinta. Baigtinio elemento dydis mažėja, jei paklaida netenkina leistinos vartotojo paklaidos, ir didėja, jei paklaida pasiekia reikiamą dydį. Taikant šį metodą, gaunami netolygūs tinklai su kintamo dydžio elementais (Demkowicz ir kt., 1985; Hansbo, 1989; Zeng, 1991; Zhu, Zienkiewicz, 1988; Okstad, Mathisen, 1994). Ši schema prisitaikančiųjų baigtinių elementų strategijose naudojama dažniausiai. Smulkiau šis metodas bus aprašytas 1.2.4 skyriuje.

2. *P-versija*. Priešingai nei *h-versija*, esant *p-versijai* elemento dydis nekinta, o ji aprašančio daugianario eilė kinta priklausomai nuo gautos paklaidos. Jei paklaida didelė, daugianario eilė didėja, o jei paklaida yra mažesnė už tam tikrą nustatytą paklaidą, tai daugianario eilė mažėja (Babuška ir kt., 1981; Gupta ir kt., 1991; Moller, 1994). Žinoma, kad *p-versija* konverguoja greičiau nei *h-versija*, tačiau *h-versija* yra populiareesnė dėl savo paprastumo.

3. *H-p-versija*. Sujungus *h* ir *p* versijas gaunama labai efektyvi *h-p-versija* (Szabo, 1990; Zeng, 1991; Zienkiewicz ir kt., 1989; Zeng, Wiberg, 1992; Melenk, 2005). Šiuo atveju konvergavimo greitis pasiekia net eksponentę (Babuška ir kt. 1986; Babuška, Suri, 1990).

4. *R-versija*. Yra dar vienas – taip vadinamasis *r*-metodas, kuris fiksuoja laisvės laipsnių skaičių ir perstumia mazgus. Tačiau šis metodas labai sprendinio nepagerina neduoda, todėl prisitaikančiųjų baigtinių elementų strategijose taikomas retai (Babuška ir kt., 1986).

Prisitaikančiosios baigtinių elementų strategijos ne tik ieško sprendinio, atitinkančio leistiną vartotojo paklaidą, bet ir tokio sprendinio, kuris būtų susietas su vadinamuoju optimaliu tinklu. Dauguma prisitaikančiųjų baigtinių elementų strategijų remiasi energetine paklaidos norma ir optimaliu laiko toki tinkle, kuriame paklaidos pasiskirstymas visų elementų atžvilgiu yra tolygus (Zienkiewicz, Zhu, 1987). Yra kelios priežastys, dėl kurių paklaidai įvertinti parenkama energetinė paklaidos norma. Visų pirma, energetinė paklaidos norma turi aiškią fizikinę prasmę, susietą su baigtinių elementų diskretizacijos procesu. Taip pat yra sukurta aiški matematinė teorija, leidžianti paaiškinti skirtingas energetinės paklaidos normos savybes. Be to, yra nemažai paklaidų vertinimo strategijų, suskurtų energetinės paklaidos normos pagrindu. Ir galiausiai, taikant tolygaus paklaidos pasiskirstymo ploto vienetui optimalumo kriterijų gaunamas pigiausias tinklas, turintis mažiausią laisvės laipsnių skaičių (Li, Bettles, 1995), kai tuo tarpu taikant kitus optimalumo kriterijus gaunami žymiai didesni elementų skaičių turintys tinklai.

Šiuolaikinių kompiuterinių technologijų galimybės labai išaugo, todėl vis daugiau dėmesio skiriama alternatyvių optimalumo kriterijų paieškai ir taikymui (Bugada, Onate, 1992; Picasso, 2003; Coorevits, Bellenger, 2004; Stevenson, 2005), nes tokie kriterijai leistų vartotojui tinklą tankinti tose vietose, kur tuo metu sukoncestruotas jo dėmesys (pvz. didžiausių įtempimų vietose). Pavyzdžiui, siekiant

L.Vasiliauskienė. Pristatikančiosios baigtinių elementų strategijos plokštuminiams tamprumo teorijos uždaviniais

sukurti prie įtempimų lauko pasiskirstymo priderintą BE tinklą, prisitaikančiųjų BE strategijų algoritme naudojamas paprastas įtempimų tolydumo indikatorius (Stupak, Kačianauskas, 2000), lygus iš mazginių įtempimų reikšmių intervalo išrinktai maksimaliai reikšmei, padalintai iš įtempimų globalių ašių atžvilgiu, Mizeso įtempimų ar kitų dominančių dydžių. Taip apibrėžtas tolydumo indikatorius gali būti laikomas optimalumo kriterijumi ir taikomas suradimui informacijos, reikalingos naujo tinklo generavimui.

Parenkant alternatyvius optimalumo kriterijus išsiplečia prisitaikančiųjų baigtinių elementų strategijų uždavinys, pagal kurį galutinis optimalus tinklas turi ne tik tenkinti iš anksto nustatytą vartotojo paklaidą, bet ir užtikrinti, kad gauti poslinkiai, deformacijos ir įtempimai yra pakankamai saugūs prie esamų apkrovų. Optimalumo kriterijai yra skirstomi į globaliuosius ir lokaliuosius optimalumo kriterijus, kurie plačiau aptarti bus 3.2 skyriuje.

1.1.4 Singuliarių taškų įtakos sprendinio konvergavimo greičiui įvertinimas

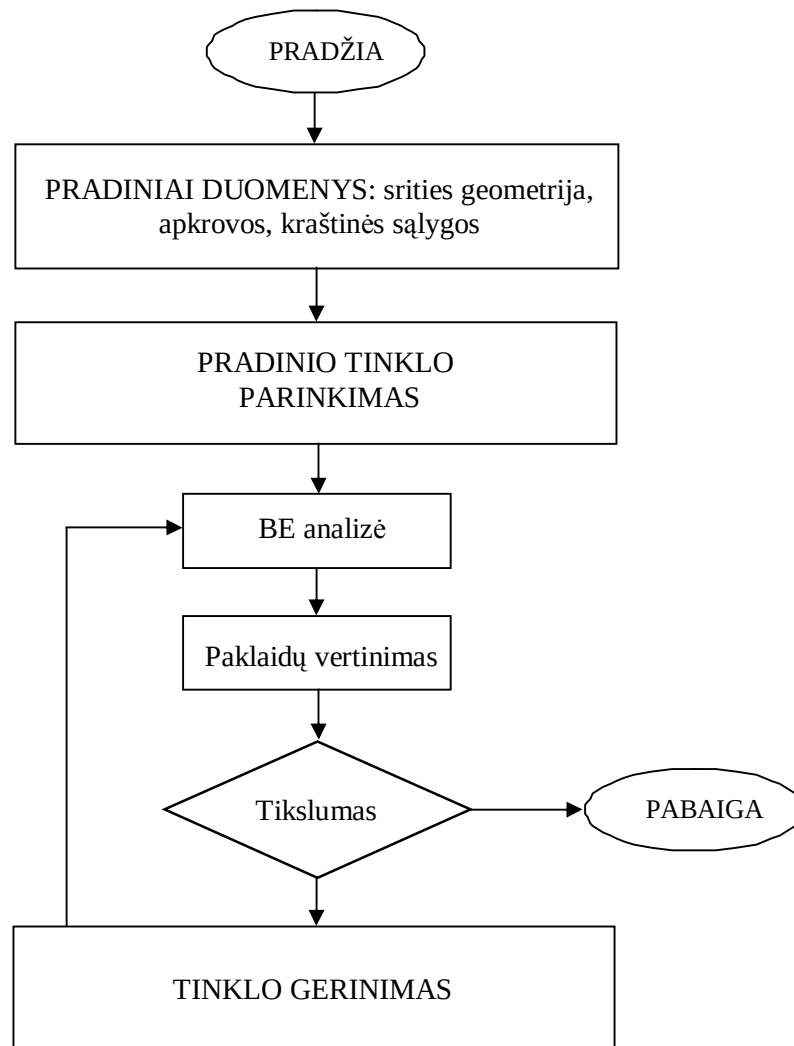
Prisitaikančiųjų BE tinklų taikymas įvairius inžinerinių uždavinių modeliavimui smarkiai praplečia skaitinių eksperimentų galimybes, tačiau net ir labai efektyvus algoritmas gali neduoti laukiamo rezultato, jei nagrinėjamoji sritis turi singuliarių taškų, su kuriais labai dažnai susiduriame irimo mechanikos, šilumos sklaidimo uždaviniuose, tiriant kompozitines medžiagas ar nagrinėjant sudėtingos geometrijos uždavinį.

Nemažai autorių nagrinėjo sprendinio tikslinimo metodus singuliarioje zonoje. Vienas iš senesnių būdų pagerinti baigtinių elementų analizės procesą yra parinkti specialius singuliarus elementus, jungiančius standartinį BE tinklą su singuliaru elementu (Barsoum, 1976). Tokie singuliarūs elementai gaunami parenkant specialias formos funkcijas BE analizei. Šis metodas yra gana populiarus, tačiau neleidžia pagerinti sprendinio tikslumo tankinant BE tinklą. Tokiu atveju singuliarioje zonoje vietoj standartinių baigtinių elementų siūloma naudoti singuliarių elementų juostas, vadinamas S-juostomis (Aalto, 1998; Aalto, 1999). S-juostos geometrija apibrėžiama pagal singuliarų srities tašką P ir ribinę kreivę Γ , kuri kaip tik ir atskiria S-juostą nuo standartinio izoparametrinio BE tinklo. Elemento S-juostoje geometrija yra apibrėžiama taip, kad S-juostos koordinatės būtų tiesiškai susijusios su elemento koordinatėmis. Autorius parodo, kad sujungus S-juostas su superkonvergenciniu elementų fragmento atkūrimo metodu, gaunama efektyvi technologija uždaviniams, turintiems singuliarių zonų, skaičiuoti. Nagrinėdamas sritis su singuliariais kampais, Apel pasiūlė lokalių tinklo gradavimą singuliariaus taško aplinkoje (1996, 1998). Lygties, aprašančios šilumos sklaidimo ar elastinės deformacijos uždavinius, tikslus sprendinys turi aibę singuliarių taškų. Kiekvienas singuliarus taškas yra nusakomas tikrinių reikšmių seka (Kondratiev, 1967; Kondratiev, 1970; Grisvard, 1985; Knyazev, Osborn, 2006). Struktūrinės mechanikos uždaviniuose šios tikrinės reikšmės padeda nusakyti srities įtrūkimo ar visiško suirimo tikimybę šalia singuliariojo taško. Tai ypač aktualu elektronikoje bei naudojant kompozitines

medžiagas. Tokių singuliariųjų sričių, turinčių smailių kampų, susidedančių iš keleto medžiagų, ar esant sudėtingoms kraštinėms sąlygoms, analizei yra sukurta keletas metodų. Vieni iš pavyzdžių galėtų būti Steklovo bei modifikuotasis Steklovo metodai (Yosibash, Szabo, 1995). Turint apskaičiuotas tikrines reikšmes, galima skaičiuoti įtempimų intensyvumo koeficientus, kurie dvimatėje tiesinėje elastiškumo teorijoje yra tiesiogiai susieti su įtempimų būvio energija. Sujungiant Steklovo metodus su p -baigtinių elementų versija, įtempimų intensyvumo daugikliai apskaičiuojami ypač tiksliai tiek izotropinėms, tiek anizotropinėms sritims (Yosibash, Szabo, 1995, 1996). Bendruoju eliptinės lygties atveju tikslios tikrinės reikšmės nėra žinomos. Tokiu atveju yra taikomas modifikuotasis Steklovo metodas. Šis metodas, išspręstas baigtinių elementų metodu, užtikrina tikrinių reikšmių konvergavimą didėjant laisvės laipsnių skaičiui. Tikslūs skaičiavimo metodai, leidžiantys surasti tikslias tikrines reikšmes ir tikrines funkcijas, apsiriboja izotropinėmis medžiagomis iš ne daugiau trijų regionų (Dempsey, Sinclair, 1981; Ying, 1986). Tikrinės reikšmės gali būti surastos taikant iteracinį matricų metodą, kai tikrinės reikšmės randamos pagal skaitmeninį Rungės, Kuto ir Fehlbergo metodą (Papadakis, 1988). Smulkiau netiesinių lygčių tikrinių reikšmių skaičiavimo procesą aprašo Leguillon su bendraautoriais (1987). Sudėtingos geometrijos uždavinių su singuliariaisiais taškais sprendimą nagrinėjo Qian (2001), Song ir Wolf (2002).

1.2 Pristaikančiosios baigtinių elementų strategijos: pagrindinės lygtys ir algoritmai

Taikant baigtinių elementų metodą, labai svarbu turėti tinkamą baigtinių elementų tinklą. Kadangi baigtinių elementų sprendinio paklaida tiesiogiai priklauso nuo baigtinių elementų tinklo kokybės, labai dažnai prireikia papildomų veiksmų tai kokybei pagerinti. Ilgą laiką baigtinių elementų tinklų generavimo kokybės vertinimas rėmėsi tik panašių uždavinių sprendimo patirtimi. Vartotojas turėdavo pats nuspręsti, ar gauto sprendinio kokybė jau pakankamai gera, ar dar ne. Jei paklaida dar buvo pakankamai didelė, jis turėdavo konstruoti naują tinklą ir pakartoti iš naujo paklaidų vertinimo procesą. Taip procesas būdavo kartojamas tol, kol vartotojas gaudavo reikiamo tikslumo sprendinį. Tačiau rankinis darbas yra pernelyg neapibrėžtas ir per daug brangus laiko atžvilgiu, todėl elementų dydžio bei tankumo pasiskirstymas gali būti prognozuojamas taikant automatinius tinklo generavimo metodus. Elemento dydis naujame tinkle perskaičiuojamas remiantis prieš tai buvusios analizės rezultatais. Tokiu būdu automatiškai vienas po kito gaunami vis geresnės ir geresnės kokybės tinklai. Po keleto bandymų gaunamas tinklas, jau tenkinantis leistiną vartotojo paklaidą. Taigi tik minimaliai įsikišus vartotojui, gauname automatinį algoritmą galutiniam aukštos kokybės tinklui generuoti.



1.1 pav. Tradicinis prisitaikančiųjų baigtinių elementų strategijų algoritmas

Pirmieji adaptyvūs tinklai baigtinių elementų analizei buvo pasiūlyti Melosh ir Marshal (Melosh, Marshal, 1977), paskui – Peano ir kt. (Peano, Pasini, Riccioni, Sardella, 1979) darbai. Nuo to laiko daug pastangų įdėta plėtojant prisitaikančiąsias baigtinių elementų strategijas bei optimizuojant tinklo generavimo procesą (Baušys, 1999; Lee, Lo, 1993; Löhner, 1996; Zienkiewicz, Taylor, 2000; Morin ir kt. 2002).

Pagrindinė prisitaikančiųjų baigtinių elementų strategijos idėja remiasi diskretizacijos paklaidų vertinimu. Šios paklaidos skaičiuojamos visoje srityje remiantis gautu baigtinių elementų sprendiniu, tada pagal gautas paklaidas, generuojamas naujas srities tinklas ir skaičiuojamas naujas baigtinių elementų sprendinys tam gautam tinklui. Kaip jau buvo paminėta, šiuo metu yra žinomos keturios pagrindinės prisitaikančiojo proceso versijos, tai *h*-versija, *p*-versija, *h-p*-versija bei *r*-versija.

Sakoma, kad baigtinių elementų analizės algoritmas yra prisitaikantis, jei skaičiavimo procese jis taip naudoja tarpinius rezultatus, kad gautasis galutinis sprendinys yra laikomas optimaliu. Toks algoritmas yra visiškai automatinis, o iteracinis procesas visiškai valdomas leistina vartotojo paklaida. Tradicinis prisitaikantis tinklo tobulinimo algoritmas susideda iš tokių žingsnių (1.1 pav.):

L. Vasiliausienė. Pristaikančiosios baigtinių elementų strategijos plokštuminiams tamprumo teorijos uždaviniams

1. Sukonstruojamas pradinis tinklas, parenkama norima sprendinio paklaida.
2. Duotajam tinklui apskaičiuojamas baigtinių elementų sprendinys.
3. Įvertinama globalioji bei lokali diskretizacijos paklaidos.
4. Tikrinama globalioji paklaida. Jei ji patenka į leistinas paklaidos ribas, skaičiavimai stabdomi ir algoritmas baigiamas. Kitu atveju – gautoji paklaida kartu su leidžiama paklaida naudojamos tinklo tobulinimo informacijai ruošti.
5. Patobulinamas senas tinklas arba sugeneruojamas naujas ir grįžtama į antrą žingsnį.

Dažniausiai tinklo generavimo procesas pradedamas nuo vartotojo gana grubiai sukonstruoto pradinio tinklo, turinčio vos kelis elementus, o elemento dydžio funkcija priskiriama rankiniu būdu kiekvienam tinklo elementui. Generuojant baigtinių elementų tinklą, naujo taško elemento dydžio funkcija nustatoma taikant tiesinę interpoliaciją, tada generuojamas naujas tinklas ir skaičiuojamas naujas baigtinių elementų sprendinys bei vertinama diskretizacijos paklaida, siekiant perskaičiuoti elemento dydžio funkciją jau iš „posteriori“ paklaidos vertinimo. Yra dvi pagrindinės priežastys, kodėl reikėtų stengtis gauti optimalų tinklą su tolygiai paskirstytomis sprendinio diskretizacijos paklaidomis. Pirmoji priežastis yra ta, kad optimaliame tinkle eliminuojama sprendinio singuliarumų įtaka sprendinio konvergavimo greičiui. Tai leidžia pasiekti norimą tikslumą didžiausiu konvergavimo greičiu, kartu naudojant mažiausią laisvės laipsnių skaičių. Antroji priežastis ta, kad diskretizacijos paklaidos nustatymo efektyvumas didėja baigtinių elementų tinklui artėjant prie optimalaus. Be to, praktiniu požiūriu, nereikia visiškai tiksliai optimizuoti kiekvieno tinklo elemento tikrinant lokalias paklaidos normas, užtenka surasti optimalų tinklą, kuriame būtų tenkinami tik globaliosios diskretizacijos paklaidos reikalavimai. Norint sugeneruoti patį geriausią tinklą nagrinėjamam uždaviniui, reikia rasti kompromisą tarp ieškomo sprendinio tikslumo ir skaičiavimo efektyvumo. Sprendinio tikslumas gali būti lyginamas su tinklo elementų skaičiumi – kuo elementų skaičius didesnis, tuo jų dydis mažesnis ir tuo tiksliau aproksimuojama nagrinėjamoji sritis. Tačiau reikia nepamiršti, kad smarkiai padidėjęs elementų skaičius lemia didesnes skaičiavimo sąnaudas. Paprastai optimaliam tinklui rasti prireikia 3–5 iteracijų. Prisitaikančiąsias baigtinių elementų strategijas sudaro dvi pagrindinės dalys: diskretizacijos paklaidų nustatymas bei naujo prisitaikančiojo tinklo generavimas. Šios dvi dalys plačiau aprašytos tolesniuose skyriuose.

1.2.1 Pagrindinės tamprumo teorijos lygtys

Tamprumo teorijos metodais nustatomos projektuojamų objektų racionalūs matmenys, pavoingosios ir leistinos apkrovos. Ši teorija nagrinėja įtempimų pasiskirstymą kietuosiuose tampruosiuose kūnuose, tų kūnų deformacijas ir jų taškų poslinkius, kuriuos sukelia įvairūs išoriniai poveikiai. Realios konstrukcijos mechaninį modelį apibūdina geometrinė forma, įtvirtinimo paviršius,

apkrovimas ir medžiagos mechaninių rodiklių pasiskirstymas. Pagrindinės tiesiškai tampraus kūno statinės pusiausvyros lygtys aprašomos tokių diferencialinių lygčių sistema:

$$\operatorname{div} s + f = 0, \quad (\text{pusiausvyros lygtys}) \quad (1.1)$$

$$e = Lu, \quad (\text{vientisumo lygtys}) \quad (1.2)$$

$$s = De = DLu, \quad (\text{fizinės lygtys}) \quad (1.3)$$

kraštinės sąlygos

$$u|_{G_1} = u_0, \quad (\text{poslinkių kraštinės sąlygos}) \quad (1.4)$$

$$n^T s|_{G_2} = t_0, \quad (\text{natūraliosios kraštinės sąlygos}) \quad (1.5)$$

čia u , e ir s – atitinkamai poslinkių, deformacijų ir įtempimų laukai. f apibrėžia kūną veikiančias jėgas, L – deformacijų diferencialinis operatorius, o D – tampraus būvio standumo matrica. $G = G_1 \cup G_2$ – nagrinėjamos srities Ω kraštas. n apibrėžia išorinį normalės vektorių ant G_2 . u_0 , o t_0 – iš anksto nusakyti poslinkiai ir įtempimai atitinkamai ant G_1 ir G_2 .

Poslinkių laukas apibrėžiamas kiekvienam elementui, kuris dažniausiai aprašomas tokia forma:

$$u_h = Nd, \quad (1.6)$$

čia N – formos funkcijos, o d – apibendrinti poslinkiai to elemento mazguose.

Deformacijų ir įtempimų laukai apskaičiuojami pagal formules (1.2) ir (1.3):

$$e_h = LNd, \quad (1.7)$$

$$s_h = DLNd, \quad (1.8)$$

Čia p – formos funkcijų daugianarių laipsnis, o m – aukščiausias diferencialinio operatoriaus L laipsnis. Tada elemento poslinkių, deformacijų ir įtempimų tikslumas išreiškiamas formulėmis:

$$\|u - u_h\| \leq C_u h^{p+1}, \quad (1.9)$$

$$\|e - e_h\| \leq C_e h^{p-m+1}, \quad (1.10)$$

$$\|s - s_h\| \leq C_s h^{p-m+1}, \quad (1.11)$$

Čia u , ε ir σ – atitinkamai tikslūs poslinkiai, tikslios deformacijos ir tikslūs įtempimai, C_u , C_e ir C_s yra konstantos, o h – elemento dydžio charakteristika.

1.2.2 Baigtinių elementų metodas

Baigtinių elementų metodas leidžia pakeisti diferencialinių lyčių sprendimą algebrinių lygčių sistemos sprendimu. Tam konstrukcija sudalinama į baigtinį skaičių figūrų, kurios vadinamos baigtiniais elementais. Skaičiuojant baigtinių elementų metodu reikalaujama, kad ieškoma funkcija būtų tolydi tik baigtinio elemento viduje, o tarp elementų gali būti ieškomos funkcijos trūkiai. Todėl baigtinių elementų metodą patogu taikyti sudėtingų konstrukcijų (netaisyklingos formos, su skylėmis ar išpjovomis, esant skirtingoms kraštinėms sąlygoms) skaičiavimo uždaviniams spręsti. Baigtinių elementų metodą galima skaidyti į tokius etapus: informacijos apie atskirus elementus suformavimas, atskirų elementų pagrindinių priklausomybių suformavimas, atskirų elementų surinkimas į vieną modelį ir šio modelio sprendimas nustatant pagrindinius nežinomuosius bei kitų nežinomų dydžių apskaičiavimas pavieniuose elementuose.

Pagrindinės tiesiškai tamprus kūno statinės pusiausvyros lygtys gali būti gaunamos minimizuojant bendrą potencinę energiją:

$$p = \frac{1}{2} \int_V [\sigma]^T \varepsilon dV - \int_V [u]^T p e dV - \int_S [u]^T q dS, \quad (1.12)$$

čia s ir ε yra įtempimų ir deformacijų vektoriai, u – poslinkiai kiekviename kūno taške, p – kūną veikiančios jėgos ploto vienetui, q – . Integruojama kūno tūryje V , S – apkrauto kūno plotas.

Poslinkių laukas nusakomas formule:

$$u = Nu^e, \quad (1.13)$$

čia N – formos funkcijos, o u^e – apibendrinti poslinkiai to elemento mazguose.

Elementų deformacijos išreiškiamos tokia lygybe:

$$\varepsilon = Bu^e, \quad (1.14)$$

čia B – elementų geometrinių rodiklių matrica, gaunama diferencijuojant formos funkcijas.

Galiausiai randama elementų įtempimų matrica:

$$\sigma = D\varepsilon, \quad (1.15)$$

čia D – modulio fizinių konstantų matrica.

Viso kūno potencinė energija yra lygi visų elementų potencinių energijų sumai:

$$p = \sum_e p_e, \quad (1.16)$$

$$p_e = \frac{1}{2} \int_{V_e} [\mathbf{u}^e]^T [\mathbf{B}]^T \mathbf{D} \mathbf{B} \mathbf{u}^e dV - \int_{V_e} [\mathbf{u}^e]^T [\mathbf{N}]^T p dV - \int_{S_e} [\mathbf{u}^e]^T [\mathbf{N}]^T \mathbf{q} dS, \quad (1.17)$$

čia V_e – elemento tūris, S_e – apkrautos elemento dalies plotas.

Minimizuojame elemento potencinę energiją ir gauname tokias lygtis:

$$\frac{\partial p_e}{\partial \mathbf{u}^e} = \int_{V_e} ([\mathbf{B}]^T \mathbf{D} \mathbf{B}) \mathbf{u}^e dV - \int_{V_e} [\mathbf{N}]^T p dV - \int_{S_e} [\mathbf{N}]^T \mathbf{q} dS = \mathbf{K}^e \mathbf{u}^e - \mathbf{F}^e, \quad (1.18)$$

$$\mathbf{F}^e = \int_{V_e} [\mathbf{N}]^T p dV + \int_{S_e} [\mathbf{N}]^T \mathbf{q} dS, \quad (1.19)$$

$$\mathbf{K}^e = \int_{V_e} ([\mathbf{B}]^T \mathbf{D} \mathbf{B}) \mathbf{u}^e dV. \quad (1.20)$$

$\mathbf{F}^e$ – mazginių jėgų vektorius, $\mathbf{K}^e$ – elemento standumo matrica.

Baigtinių elementų metodo populiarumą lėmė automatinis duomenų surinkimas panaudojant vartotojui labai patogius geometrinius įvaizdžius, galimybė įvertinti įvairiausias kraštines sąlygas ir galimybė tame pačiame modelyje turėti įvairios formos elementus, kas labai padidina metodo efektyvumą.

1.2.3 Pagrindinės diskretizacijos paklaidų vertinimo priklausomybės

Tyrinėjant gauto baigtinių elementų sprendinio kokybę, būtina atsižvelgti į diskretizacijos paklaidas. Nagrinėjamame srities taške diskretizacijos paklaida taikant baigtinių elementų analizę apibrėžiama kaip tikslaus ir procesorinio baigtinių elementų sprendinio skirtumas:

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_h, \quad (1.21)$$

čia $\mathbf{u}$ – tikslus sprendinys, $\mathbf{u}_h$ – atitinkamas BE aproksimacijos sprendinys.

Šiuo konkrečiu atveju paklaida išreikšta poslinkiais, nors lygiai taip pat gali būti išreiškiama įtempimais, deformacijomis ar kitais mus dominančiais dydžiais.

Naudojantis prisitaikančiąsias baigtinių elementų strategijas dažnai tenka įvertinti tam tikro sprendinio tikslumą. Kadangi tikslus sprendinys dažniausiai nėra žinomas, baigtinių elementų sprendiniui įvertinti reikia taikyti papildomus aproksimacijos metodus. Populiariausias paklaidos įvertis remiasi energetine norma ir apibrėžiamas tokia formule:

$$\|\mathbf{e}\|_{E(W)} = \left(\int_W \mathbf{e}^T \mathbf{L} \mathbf{e} dW \right)^{\frac{1}{2}} \quad (1.22)$$

čia $\mathbf{L}$ – pagrindinių lygčių diferencialinis operatorius.

Diskretizacijos paklaidų nustatymo metodai pateikia ir globaliąją paklaidos charakteristiką, išreikštą viena ar kita norma, bei paklaidų lauką, apimantį visą nagrinėjamą diskretizacijos sritį. Diskretizacijos paklaidų laukas yra pagrindinė priemonė kuriant naują prisitaikančiųjų baigtinių elementų tinklą. Praktikoje ši norma gaunama susumavus kiekvieno srities Ω tinklo elemento normas:

$$\|e\|_{E(W)}^2 = \sum_{i=1}^{nel} \|e_i\|^2 = \sum_{i=1}^{nel} \int_{W_i} (s_h - s^*)^T D^{-1} (s_h - s^*) dW_i, \quad (1.23)$$

čia $\|e\|_{E(W)}$ – bendra diskretizacijos paklaida, $\|e_i\|$ – i -tojo elemento paklaida, Ω_i – i -tojo elemento sritis, o nel yra bendras elementų skaičius, s_h – baigtinių elementų įtempimų sprendinys, o s^* – patikslintas įtempimų sprendinys, gautas pritaikius įtempimų atkūrimo procedūrą (Borooman, Zienkiewicz, 1997; Zienkiewicz, Zhu 1992).

Visos srities globalioji bei atskiros elemento paklaidos išreiškiamos tokiomis formulėmis:

$$\begin{aligned} \|e\| &\leq Ch^{p-m+1}, \\ \|e_i\| &\leq C_e h^{p-m+2}. \end{aligned} \quad (1.24)$$

Tačiau tokios paklaidų išraiškos nėra patogios praktiniams skaičiavimams, nes turi energijos dimensijas, todėl dažniausiai naudojama bedimensinė paklaidų forma:

$$h = \left(\frac{\|e\|^2}{\|u_h\|^2 + \|e\|^2} \right)^{\frac{1}{2}}, \quad (1.25)$$

$$h_i = \left(\frac{\|e_i\|^2}{\|u_h\|^2 + \|e\|^2} \right)^{\frac{1}{2}}, \quad (1.26)$$

čia

$$\|u_h\|^2 = \int_W s_h D^{-1} s_h dW. \quad (1.27)$$

yra energijos norma, h ir h_i – atitinkamai bendroji leistina bei atskiros elemento paklaidos. Remiantis aukščiau minėtomis lygybėmis, santykinė globalioji paklaida išreiškiama tokia formule:

$$h = \left(\sum_{i=1}^{nel} (h_i)^2 \right)^{\frac{1}{2}}. \quad (1.28)$$

Diskretizacijos paklaidų įvertinimas gali būti atliktas prieš baigtinių elementų analizę (apriorinis paklaidų vertinimas) arba po baigtinių elementų analizės (posteriorinis arba poprocesorinis paklaidų vertinimas). Šiame darbe bus taikomi poprocesoriniai diskretizacijos paklaidų vertinimo metodai. Šio tipo paklaidų vertinimo metodų pagrindinė idėja pagrįsta naujo aukštesnės tikslumo klasės sprendinio radimu ir jo naudojimu vietoj tyrinėjamos inžinerinės konstrukcijos tikslaus sprendinio, kuris dažniausiai nėra žinomas. Taigi poprocesorinio paklaidų vertinimo metodų esmę sudaro tikslaus sprendinio pakeitimas poprocesoriniu aukštesnės tikslumo klasės sprendiniu

$$e_u \approx \bar{e}_u = u^* - u^h, \quad (1.29)$$

čia $\bar{e}_u$ – taškinė apskaičiuota diskretizacijos paklaida. Ši paklaida taip pat reiškia energetine norma

$$\|\bar{e}\|_{E(W)} = \left(\int_W \bar{e}_u^T L \bar{e}_u dW \right)^{\frac{1}{2}} \quad (1.30)$$

ir yra lygi atskirų nagrinėjamų srities elementų paklaidų normų sumai:

$$\|\bar{e}\|_{E(W)}^2 = \sum_{i=1}^{nel} \|\bar{e}_i\|^2 = \sum_{i=1}^{nel} \int_{W_i} \bar{e}_u^T D^{-1} \bar{e}_u dW_i. \quad (1.31)$$

Paklaidų vertinimo metodų kokybė matuojama efektyvumo indeksu

$$q = \frac{\|\bar{e}\|}{\|e\|} = \frac{\|u^* - u^h\|}{\|u - u^h\|}. \quad (1.32)$$

Jei $q > 1$, tai paklaida yra pervertinta, o kai $q < 1$, tai paklaida nepakankamai įvertinta. Šis efektyvumo indeksas gali būti skaičiuojamas tik tais atvejais, kai yra žinomas tikslus tyrinėjamo modelio sprendinys. Patikimas paklaidų vertinimas turi būti asimptotiškai tikslus – kai elementų dydis h artėja prie nulio, efektyvumo indeksas turi konverguoti prie vieneto. Efektyvumo indeksas gali būti skaičiuojamas arba visai sričiai q_w , arba atskiram elementui q_k .

1.2.4 *h*-pristaikančiosios strategijos algoritmas

Ši pristaikančioji strategija formuluojama taip: norint generuoti baigtinių elementų tinklą

$$T_h = \{W_e | e = 1, 2, \dots, nel\}, \quad (1.33)$$

turintį mažiausią laisvės laipsnių skaičių, reikia, kad globalioji diskretizacijos paklaida būtų:

$$\bar{h} \leq h_{TOL}, \quad (1.34)$$

čia $\bar{h}$ – sprendinio paklaida, h_{TOL} – maksimali leidžiama paklaida, kurią sudaro atskirų elementų paklaidų dedamosios:

$$\bar{h} = \sqrt{\sum_{k=1}^{nel} \bar{h}_k^2}, \quad (1.35)$$

$$\bar{h}_k = \frac{\|\bar{e}\|_k}{\sqrt{\|u^h\|^2 + \|\bar{e}\|^2}}, \quad (1.36)$$

čia $\|u^h\|$ pažymi baigtinių elementų sprendinio normą.

Šis uždavinys yra netiesinis minimizacijos uždavinys ir sprendžiamas iteraciniu būdu. Santykinė elemento paklaida h_i prisitaikančiosiose strategijose apibrėžia paklaidos pasiskirstymą, taip pat ji naudojama ir tinklo dydžio funkcijai $h = h(x)$ nustatyti.

Apskaičiavus diskretizacijos paklaidą ir patikrinus, ar sprendinys atitinka (1.34) sąlygą, sprendžiama, ar reikia generuoti naują baigtinių elementų tinklą, kurio generavimas pagrįstas tinklo dydžio funkcija h .

Atsižvelgiant į (1.34) sąlygą, apibrėžiamas norimos globaliosios paklaidos normos dydis

$$\|e\|_r = \frac{h_{TOL} \sqrt{\|u^h\|^2 + \|\bar{e}\|^2}}{K}. \quad (1.37)$$

Šiuo atveju atsargos koeficientas K (jo reikšmė paprastai būna lygi 1,0–1,5) leidžia išvengti bereikalingų iteracijų tais atvejais, kai diskretizacijos paklaidos dydis yra labai artimas leidžiamai paklaidai.

Norima atskiro elemento norma $\|e\|_{r_k}$ aprašoma optimalumo kriterijumi. Optimalumo kriterijų rūšys bei jų įtaka prisitaikančiosioms baigtinių elementų strategijoms bus aprašyta 3 skyriuje. Bendruoju atveju lokalusis tinklo tobulinimo koeficientas užrašomas tokiu santykiu:

$$x_k = \frac{\|e\|_k}{\|e\|_{r_k}}. \quad (1.38)$$

Koeficientas x_k pateikia naujo baigtinių elementų tinklo generavimo informaciją: jei $x_k > 1$, tai lokioji paklaida viršija leistiną paklaidos dydį, vadinasi, šio elemento dydis turi būti mažinamas, o jei $x_k < 1$, tai reiškia, kad šio elemento lokioji paklaida yra mažesnė už leistiną paklaidą, ir šio elemento dydis gali būti padidintas.

Tankinant baigtinių elementų tinklą, atskiro elemento dydis h artėja prie nulio, o diskretizacijos paklaida, išreikšta energetine norma, apibrėžiama tokia nelygybe:

$$\|e\|_{E(W)} \leq Ch^{\min(p,l)}, \quad (1.39)$$

čia C – teigiama konstanta, priklausanti nuo tikslaus sprendinio tolygumo, p – elemento bazinių funkcijų eilė, l – sprendinio singuliarumų įtaką aprašantis parametras. Kadangi naudojant optimalų tinklą singuliarumai neturi didelės įtakos, tai šią nelygybę galime perrašyti taip:

$$\|e\|_{E(W)} \leq Ch^p. \quad (1.40)$$

Atliekant skaičiavimus nelygybės pakeičiamos lygybėmis ir k -tajam senojo tinklo elementui gauname:

$$\|e\|_{E(k)} \approx C_1 h_k^p. \quad (1.41)$$

Naujo tinklo generavimo metu iš turimo k -tojo elemento bus sukurti nauji h'_k dydžio elementai. Tada naujai gautame tinkle diskretizacijos paklaida bus aprašoma tokia forma:

$$\|e\|_{E(k')} \approx C_2 (h'_k)^p. \quad (1.42)$$

Kadangi iteracinio proceso esmė – gauti optimalų baigtinių elementų tinklą, kurio visų elementų tinklo tobulinimo koeficientas x_k būtų lygus vienetui, tai laikydami, kad $C_1 \approx C_2$, gauname naują tinklo dydžio funkciją

$$h'_k = \frac{h_k}{x^{\frac{1}{p}}}. \quad (1.43)$$

Ši prielaida buvo padaryta tariant, kad uždavinio singuliarumai konvergavimo greičiui įtakos neturi, tačiau tais atvejais, kai singuliariosios sritys yra žinomos iš anksto (pvz., kampai, skirtingų medžiagų sąlyčio sritys ir t. t.), singuliariųjų taškų aplinkoje naujas elemento dydis gali būti skaičiuojamas parametru p keičiant į l :

$$h'_k = \frac{h_k}{x^{\frac{1}{l}}}. \quad (1.44)$$

Prisitaikančiųjų strategijų efektyvumas labiausiai priklauso nuo diskretizacijos paklaidų – kuo paklaidos tiksliau įvertintos, tuo mažiau sąnaudų reikia galutiniam optimaliam tinklui gauti.

1.2.5 Plintančio fronto technikos algoritmas

Labai svarbu prisitaikančiųjų baigtinių elementų strategijų algoritmui parinkti kokybišką tinklo generatorių, nes yra žinoma kad baigtinių elementų dydis bei pasiskirstymas turi tiesioginę įtaką prisitaikančiųjų strategijų algoritmo konvergavimo greičiui, be to, jei tinkle yra pernelyg daug elementų, neatitinkančių formai keliamų reikalavimų, proceso konvergavimas gerokai sulėtės. Taip pat tinklo generatorius turi būti greitas, paprastai taikoma tokia taisyklė – jei sugeneruoti baigtinių elementų tinklą užtrunka ne ilgiau, nei baigtinių elementų analizė, vadinasi, tinklo generatorius atitinka greičiui keliamus reikalavimus. Plintančio fronto technikos (PFT) algoritmas yra vienas iš populiariausių bei efektyviausių tinklo generavimo metodų, taikomų prisitaikančiosioms baigtinių elementų strategijoms. Šis metodas susideda iš trijų pagrindinių žingsnių:

1. Tinklo parametrizacija (kraštinių sąlygų aprašymas, elemento dydžio funkcijos nustatymas ir pan.).
2. Srities krašto diskretizacija.
3. Tinklo viršūnių bei elementų sukūrimas arba, kitaip tariant, paties PFT algoritmo realizacija.

Ši schema yra bendrojo pobūdžio ir prireikus taikoma ir kitiems metodams, pvz., Delaunay, tačiau egzistuoja esminis skirtumas, leidžiantis išskirti PFT algoritmą iš kitų metodų, t.y. klasikiniame PFT algoritme tinklo viršūnės sukuriamos tinklo elementų generavimo metu ir yra viena iš paties tinklo generavimo proceso dalių, kai tuo tarpu pagal Delaunay ir rekursinį erdvės dekompozicijos metodus viršūnės sukuriamos dar prieš pradedant tinklo elementų generavimą.

PFT algoritmas yra iteratyvioji procedūra, kuri prasideda nuo nagrinėjamos srities krašto diskretizacijos ir žingsnis po žingsnio užpildo sritį naujais elementais. Per kiekvieną žingsnį parenkamas krašto elementas ir jam surandama optimali viršūnė. Klasikinį plintančio fronto technikos algoritmą galima išskaidyti į tokius pagrindinius žingsnius:

1. Pradinių duomenų įvedimas:
 - a) srities briaunų aprašymas, t.y. srities briaunų tinklo T sudarymas;
 - b) elementų tinklo dydžio funkcijos nustatymas nagrinėjamai sričiai.
2. Naudojant srities briaunų tinklą T , generuojamas srities frontas F .
3. Fronto F analizė (jei frontas F netuščias):
 - a) pagal tam tikrus kriterijus pasirenkamas krašto elementas f ;
 - b) tam elementui f sukuriamas optimalus taškas P_{opt} ;
 - c) patikrinama, ar jau egzistuoja toks tinklo taškas V , kuris galėtų būti naudojamas vietoje taško P_{opt} . Jei taip, tai P_{opt} priskiriame tašką V ;
 - d) iš fronto elemento f ir P_{opt} sugeneruojamas naujas elementas K ;
 - e) patikriname, ar sugeneruotas elementas atitinka elemento dydžio reikalavimus;

4. Srities fronto F ir einamojo tinklo papildymas:

- a) iš fronto pašalinami visi elementai, naudoti naujam elementui K kurti;
- b) pagal elemento K primityvų informaciją papildomas frontas F ;
- c) nauju elementu K papildomas einamasis tinklas;

5. Jei frontas F nėra tuščias, grįžtame į trečią žingsnį.

6. Optimizuojamas gautas tinklas.

Ši schema rodo pagrindines metodo dalis, tačiau visiškai neatspindi tam tikrų niuansų, galinčių iškilti tinklo generavimo metu. Sunkiausios metodo vietos yra susijusios su fronto primityvo parinkimu, fronto analize bei optimalaus taško parinkimo galimybėmis, naujai sukonstruoto elemento atitikimu iškeltiems dydžio ir formos reikalavimams bei tinkamų duomenų struktūrų naudojimui. Nepaisant to, sunkumus, kylančius naudojant plintančio fronto technikos algoritmą, lengva identifikuoti, nes dažniausiai jie būna susiję su

- situacijos analizavimu tam tikro taško ar regiono aplinkoje;
- teisingu įvairių primityvų susikirtimo identifikavimu, pvz., patikrinti, ar naujai gautas elementas neišeina už srities ribų;
- optimalaus taško parinkimu iš jau egzistuojančių taškų ir naujai duotajam fronto elementui sukurti taško.

Kitaip tariant, kiekvienas naujai sukurtas taškas bei jį atitinkantis elementas turi tenkinti kokybei keliamus reikalavimus ir apsaugoti tolesnę tinklo generavimo procedūrą nuo galimų sunkumų.

Nauji tinklo elementai kuriami remiantis einamojo fronto briaunomis, t. y. vieno ar kito elemento sukūrimas priklauso nuo to, koku būdu analizuojamas frontas. Remiantis George ir Seveno (George, Seveno, 1994) atliktais tyrimais, fronto primityvas naujam elementui formuoti dažniausiai parenkamas atsižvelgiant į tam tikrus specifinius reikalavimus, pvz., briaunų ilgį, generuojamų elementų plotus, kampus tarp briaunų ir pan. Fronto elementai rūšiuojami ir parenkami elementui generuoti pagal keletą kriterijų. Pavyzdžiui, norint išvengti didelio ploto elementų persidengimo su mažo ploto tinklo elementais, pirmiausia parenkami tie fronto primityvai, kurie duoda mažiausio ploto tinklo elementus (Löhner, 1996). Toks fronto elementų rūšiavimas naudingas tais atvejais, kai dviems besiliečiančioms sritims, kuriose elementų dydžio funkcijos labai skiriasi, reikia sugeneruoti vieną baigtinių elementų tinklą. Kitas ne mažiau populiarus rūšiavimo kriterijus fronto primityvus parenka pagal jų ilgį – pirmiausia tinklui generuoti naudojami mažiausią ilgį turintys fronto elementai, tada ilgesni ir vėliausiai imami ilgiausi fronto elementai (Peraire ir kt., 1992).

Centrinė slenkamojo fronto technikos algoritmo ašis yra optimalaus taško parinkimas duotam fronto elementui. Tarkime, kad vienu ar kitu būdu jau parinkome fronto primityvą, taip pat žinome pageidaujamą elemento dydį tam parinktam elementui. Mūsų tikslas yra suskaidyti šią sritį į tam tikro

dydžio kokybės reikalavimus atitinkančius elementus. Generuojant naują elementą pirmiausia nagrinėjamam fronto primityvui priskiriamas prognozuojamas naujo elemento dydis, apskaičiuotas remiantis gretimų fronto primityvų saugoma informacija (pvz., vidurkinant briaunų ilgį ir pan.), po to sugeneruojamas naujas taškas, kuris kartu su nagrinėjamu fronto primityvu turėtų duoti naują tinklo elementą. Galiausiai patikrinama, ar naujasis taškas atitinka elemento dydžio reikalavimus, jei ne, optimalaus taško ieškoma iš naujo. Paprastai išskiriami trys skirtingi taškų tipai, iš kurių parenkamas vienas taškas, naudojamas naujam elementui konstruoti:

- Nagrinėjamam fronto primityvui sugeneruotas naujas optimalus taškas;
- Jau egzistuojanti tinklo viršūnė, esanti netoli nagrinėjamo fronto primityvo;
- Pagal tam tikras taisykles šalia surasto optimalaus taško papildomai sugeneruoti nauji taškai.

Yra keletas būdų, kaip nagrinėjamai fronto briaunai rasti optimalų tašką. Dvimačiu atveju dažniausiai optimaliu pripažįstamas tas taškas, kurį sujungus su nagrinėjama fronto briauna, gaunamas elementas, kaip galima labiau artimas lygiakraščiam trikampiui. Taip pat labai dažnai optimalus taškas nustatomas pagal elemento dydžio funkcijos $h(P)$ apibrėžimą kiekviename srities taške P – iš pasirinkto fronto primityvo vidurio taško išvedamas norminis vektorius ir ant jo tam tikru atstumu, nusakytu funkcijos $h(P)$, atidedamas naujas optimalus taškas (Peraire ir kt. 1992; Jin, Tanner, 1993).

Tarkime, kad f yra nagrinėjamas fronto primityvas, G – jo vidurio taškas, $h(G)$ – elemento dydžio funkcija taške G . Jei $\vec{n}_f$ yra norminis fronto primityvo f vektorius, tai optimalaus taško P_{opt} pozicija apskaičiuojama taip:

$$\vec{GP}_{opt} = \alpha h(G) \cdot \vec{n}_f, \quad (1.45)$$

čia α – norminis koeficientas, leidžiantis generuoti elementą, kaip galima artimesnį lygiakraščiam trikampiui.

Surastas optimalus taškas P_{opt} automatiškai į tinklą dar neįtraukiamas, pirmiausia patikrinama, ar nėra kitų galimų taškų į optimalaus taško vietą. Jei tokie taškai egzistuoja, jie atrenkami ir surūšiuojami pagal tai, kuris iš jų duotų geresnės formos elementą. Tokie galimi optimalūs taškai parenkami pagal atstumo kriterijų – taškas P gali būti pripažintas optimaliu tašku, jei jis yra pakankamai arti jau surasto taško P_{opt} , t. y. jei taškas P patenka į spindulio $h(P_{opt})$ apskritimą, išvestą iš optimalaus taško P_{opt} :

$$\left\| \vec{P_{opt}P} \right\| \leq h(P_{opt}). \quad (1.46)$$

Galimi optimalūs taškai P_i rūšiuojami didėjimo tvarka pagal atstumą iki optimalaus taško P_{opt} , o pats taškas P_{opt} įtraukiamas į sąrašą pirmuoju numeriu.

Suradus visus galimus taškus į tinklo elemento viršūnę, reikia patikrinti, ar iš jų sukonstruoti elementai atitiktų generuojamo tinklo elemento kokybei keliamus kriterijus. Ši patikra garantuotų, kad gautas galutinis tinklas būtų:

- topologiškai ir geometriškai teisingas,
- sudarytas iš kokybiškų elementų,
- elementų dydžio pasiskirstymas tinkle atitiktų iš anksto nurodytus reikalavimus.

Atliekant praktinius skaičiavimus laikoma, kad taškas P_i atitinka kokybės reikalavimus, jei jis nepatenka nei į vieną iš jau egzistuojančių elementų, ir nei viena iš P_i sugeneruoto elemento briaunų nekerta nagrinėjamo fronto, t. y. iš taškų P_i sukonstruoti elementai laikomi tinkamais, jei nei vienas tų elementų primityvas nekerta nei vieno fronto primityvo ir jei nei į vieną iš tų elementų neįeina joks tinklo primityvas.

Elementų kokybės patikrinimas yra pakankamai sudėtingas ir užima gana daug laiko. Pavyzdžiui, Löhner su bendraautorais (Löhner, Parikh, 1988) įrodė, kad net 80 % tinklo generavimo laiko užima įvairių sankirtų suradimas, t. y. tinklo generavimo greitis labai priklauso nuo to, kokį algoritmą parinksime elementų sankirtai tikrinti. Taip pat įrodyta, kad tinkamos duomenų saugojimo struktūros parinkimas bei atitinkamas duomenų rūšiavimas leidžia 25 % sumažinti skaičiavimo sąnaudas (Löhner, 1988; Jin, Tanner, 1993).

1.3 Išvados

Apžvelgus ir įvertinus kitų mokslininkų, tobulinusių prisitaikančiųjų baigtinių elementų strategijas, darbus, galima pastebėti, kad prisitaikančiosios BE strategijos leidžia gerokai sumažinti varginantį, daug klaidų duodantį rankinį darbą ir yra labai patogus mechaninių konstrukcijų skaičiavimo įrankis, kurį gali taikyti net ir mažai patirties baigtinių elementų srityje turintis projektuotojas. Pristaikančiosioms BE strategijoms labiau tinka nestruktūriniai BE, nes juos lengviau sutankinti ten, kur reikia nustatyti maksimalius įtempimus. Įrodyta, kad plintančio fronto technikos algoritmas atitinka visus kokybiškam tinklo generatoriui keliamus reikalavimus ir prisitaikančiųjų BE strategijų algoritmui tinka labiausiai. Dažniausiai mechanikos uždavinių sprendimui taikomos h -pristaikančiosios baigtinių elementų strategijos, o paklaidų vertinimas atliekamas taikant poprocesorinius paklaidų vertinimo metodus ir mažiausią elementų skaičių galutiniame tinkle duodančius globalius optimalumo kriterijus. Sprendinys singuliarioje srities zonoje dažniausiai reiškiamas per tikrines reikšmes, kurios surandamos taikant p -pristaikančiąsias BE strategijas.

Prisitaikančiųjų baigtinių elementų strategijos – didelių skaičiavimo sąnaudų reikalaujantis procesas, kurio eigoje nepakankamai vertinama:

- srities geometrijos ar kraštinių sąlygų įtaka pradiniam baigtinių elementų tinklui, nuo kurio kokybės priklauso iteracijų, reikalingų galutiniam optimaliam tinklui gauti, skaičius;
- stengiantis išvengti didesnių skaičiavimo sąnaudų galimybė rinktis alternatyvius optimalumo kriterijus, leidžiančius geriau atspindėti narinėjamos konstrukcijos fizines ir topologines savybes, tirta mažai; paprastai skaičiavimams parenkamas pigiausią galutinį tinklą duodantis tolygaus paklaidos pasiskirstymo optimalumo kriterijus;
- geometrinio singularumo įtakos analizei (nagrinėjamos srities singularūs taškai dažniausiai eliminuojami juos užapvalinant).

Dėl šių priežasčių prisitaikančiųjų strategijų tobulinimo uždaviniai vis dar yra aktualūs mokslininkų tarpe ir papildomos metodikos turi būti kuriamos siekiant pagerinti visą prisitaikančiųjų baigtinių elementų strategijų procesą.

2. PRADINIO BEVEIK OPTIMALAUS TINKLO GENERAVIMAS SUDĖTINGIEMS DVIMAČIAMS MECHANINIAMS OBJEKTAMS

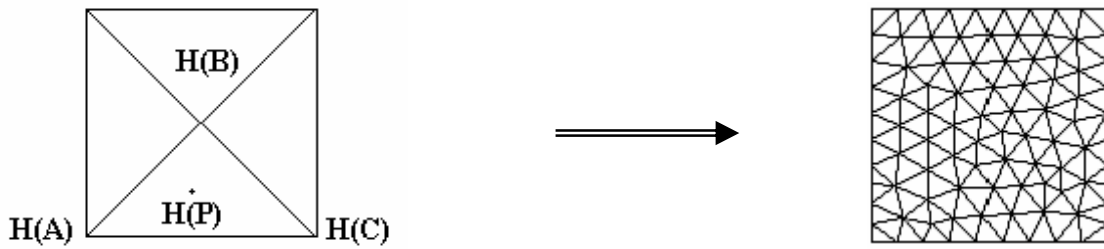
Prisitaikančiųjų BE strategijų efektyvumas tiesiogiai priklauso nuo iteracijų, reikalingų galutiniam tinklui gauti, skaičiaus. Paprastai tinklo generavimas pradedamas nuo reto, rankomis generuoto pradinio tinklo, o elemento dydžio funkcija nustatoma priskiriant tam tikrą dydį kiekvienam to pradinio tinklo elementui. Elemento dydžio funkcijai įtakos turi tik to elemento forma, o informacija apie nagrinėjamos srities topologiją ar įvairias kraštines sąlygas visiškai neįtraukiama. Tinklo generavimo procese elemento dydis naujuose taškuose nusakomas taikant tiesinę interpoliaciją. Tada skaičiuojamas atitinkamas sprendinys ir atliekama paklaidos analizė, leidžianti perskaičiuoti elemento dydį jau iš posteriorinio paklaidos įvertinimo. Toks procesas užima daug laiko ir iteracijų galutiniam rezultatui gauti. Taipogi blogus rezultatus galime gauti ir tada, kai pradinio elementų tinklo elementai yra prastos formos, ar jų dydis keičiasi labai staigiai viso tinklo atžvilgiu. Tokia situacija visada duoda blogesnius rezultatus, nei tikimasi. Pats tinkamiausias tinklas nagrinėjamam uždaviniui spręsti gaunamas suradus kompromisą tarp rezultatų tikslumo ir skaičiavimų efektyvumo. Tikslumą paprastai nusako tinklo elementų skaičius ir dydis – kuo elementai mažesni, tuo tiksliau aproksimuojama nagrinėjama sritis. Tačiau mažėjant elemento dydžiui, didėja jų skaičius, o tai lemia didesnes skaičiavimo sąnaudas. Norėdami rasti pusiausvyrą tarp tikslumo ir efektyvumo, turime sugraduoti tinklą, tačiau šioje vietoje atsiranda vienas sunkumas – įtempimų gradientai pirmojoje prisitaikančiųjų strategijų iteracijoje dar nežinomi, todėl mums reikalinga papildoma technologija, leidžianti nustatyti tinklo tankumą dar prieš prasidedant baigtinių elementų analizei.

Pagerintų pradinių tinklų pirmai prisitaikančiųjų BE strategijų iteracijai kūrimą nagrinėjo E. Kang su K. Haghighi (1995), taip pat E. Stupak ir R. Baušys (2000). Šių autorių darbuose siūloma taikyti ekspertinę žinių sistemą, kuri atsižvelgia į nagrinėjamo uždavinio ypatybes ir leidžia nagrinėjamą sritį automatiškai sudalinti į regionus bei juos surūšiuoti pagal jų kritiškumo laipsnį, nusakantį, kurioje srities dalyje tinklas turi būti tankesnis, o kur retesnis.

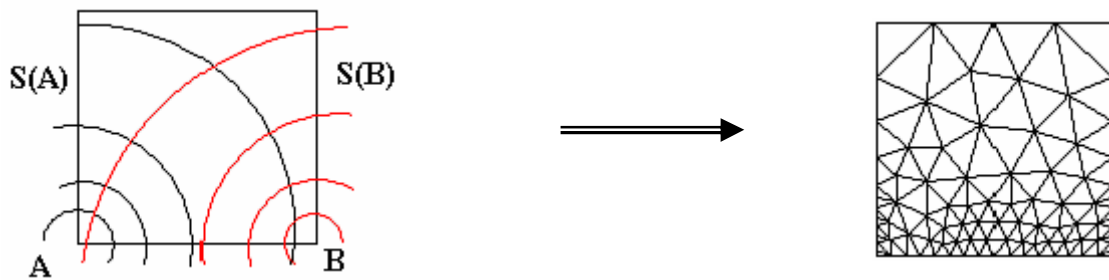
Šiame skyriuje pradinio tinklo parinkimo problemai spręsti pateikiama nauja metodika, automatiškai apimanti visą informaciją apie srities geometriją, kraštines sąlygas ar medžiagos savybes, reikalingas tinklui generuoti. Siekiant sukurti tinklą, kuris dar prieš baigtinių elementų analizę būtų tankesnis tose srityse, kur tikimasi gauti didesnius įtempimų gradientus, praplečiama kontrolinės erdvės sąvoka taip, kad ji apimtų informaciją apie elemento dydžio kitimą nagrinėjamos srities viduje. Įtrauktas į prisitaikančiųjų baigtinių elementų strategijų algoritmą, šis metodas sumažins laiko ir resursų sąnaudas, reikalingas galutiniam tinklui gauti.

Šio skyriaus tyrimų medžiaga 2001 metais pristatyta „Modern Building Materials, Structures and Techniques“ ir „Mathematical Modeling and Analysis“ konferencijose, o 2002 metais publikuota žurnale „Informatica“ (Vasiliauskienė, Baušys, 2002).

2.1 Pradinio beveik optimalaus tinklo generavimas



2.1 pav. Tradicinis kontrolinės erdvės apibrėžimas



2.2 pav. Praplėstos kontrolinės erdvės apibrėžimas

Paprastai tinklas graduojamas taikant kontrolinės erdvės sąvoką (Frey, George, 2000).

1 apibrėžimas. Sakoma, kad (D, H) yra tinklo T , dengiančio nagrinėjamą sritį Ω , kontrolinė erdvė, jei

- $\Omega \in D$, čia D yra srities Ω dengiančioji trianguliacija,
- tinklo dydžio nustatymo funkcija $H(P)$ susiejama su kiekvienu srities tašku $P \in D$ ir randama taikant tiesinę interpoliaciją:

$$H(P) = \sum_{i=0}^2 w_i H(P_i), \quad (2.1)$$

čia w_i yra taško P baricentrinės koordinatės nagrinėjamo trikampio atžvilgiu.

Kontrolinė erdvė sujungia du susijusius komponentus: pirmiausia apibrėžiama dengiančioji trianguliacija D , o paskui nustatoma funkcija H (2.1 pav.). Tinkamas šių parametrų parinkimas leidžia tam tikras fizines ar geometrines srities savybes atvaizduoti generuojamojo tinklo elementais.

Norėdami išspręsti grubaus pradinio tinklo parinkimo problemą, tinklui generuoti taikykime išplėstos kontrolinės erdvės apibrėžimą. Skirtingai nei 1-ame apibrėžime, tinklo dydžio funkciją susiesime ne su kiekvienu dengiančios trianguliacijos D tašku, bet su kritiškiausiu srities Ω regionu. Kaip matysime toliau, šis patobulinimas leis mums tiksliau atspindėti topologines ir fizines srities savybes generuojamame tinkle, bei suteiks daugiau laisvės tinklo gradavimui kontroliuoti.

Tinklo dydžiui nusakyti apie kiekvieną kritinį srities regioną įvesime kontrolinės sferos sąvoką.

2 apibrėžimas. Sfera

$$S(P, R) = \{x \mid \|P - x\| = R\} \quad (2.2)$$

yra tam tikro taško P kontroline sfera, jei jos spindulys R apibrėžiamas pagal formulę

$$R = \sum_{i=1}^n (1.2)^i kh', \quad (2.3)$$

čia h' – didžiausias pageidaujamas elemento dydis. Daugiklis $0 < k \leq 1$ reguliuoja mazgų pasiskirstymo tolygumą ir apima visą informaciją apie apkrovas, medžiagas ir kraštines sąlygas. Mažos k reikšmės duoda tankesnę tinklą, o didesnės reikšmės – retesnę; n – sferų skaičius apie tašką P , skaičius n apibrėžiamas kaip santykio $\frac{d}{h'}$ sveikoji dalis; d – nagrinėjamąsios srities įstrižainė. Pagal tokį apibrėžimą spindulys R nusako atstumą tarp bet kurių dviejų gretimų sferų, generuotų apie tašką P .

Tokiu pačiu principu apibrėžiame kontrolinę sferą apie atkarpą AB :

$$S(AB, R) = (1-t)S(A, R) + tS(B, R), \quad t \in [0, 1], \quad (2.4)$$

čia $S(A, R)$ ir $S(B, R)$ yra atkarpos AB galų kontrolinės sferos.

Dabar kontrolinės sferos apibrėžimą galime išplėsti bendruoju atveju.

3 apibrėžimas. Sferų aibė

$$\left\{ S(g(t), R) = \sum_{i=0}^n j_i S(P_i, R) \right\} \quad (2.5)$$

yra vadinama kreivės $g(t) = \sum_{i=0}^n j_i P_i$ kontroline sfera, jei visi sferų centrai yra ant tos kreivės, o sferų spindulys apibrėžtas pagal 2 apibrėžimą. Šis apibrėžimas apima ankstesnius kontrolinės sferos apie tašką bei atkarpą atvejus ir gali būti naudojamas tinklo dydžiui nustatyti apie bet kokią laisvos formos kreivę nusakytą kritinį regioną.

Turėdami bendrą kontrolinės sferos apibrėžimą, galime suformuluoti naują kontrolinės erdvės sąvoką.

4 apibrėžimas. Sakysime, kad (D, S) yra tinklo T , dengiančio nagrinėjamąją sritį Ω , kontrolinė erdvė (2.2 pav.), jei

- $\Omega \in D$, čia D yra srities Ω dengiančioji trianguliacija,
- S yra kontrolinė sfera, susieta su kiekvienu kritiniu duotos srities Ω regionu.

Praktikoje kontrolinės sferos apie tam tikrą tašką apibrėžimas taikomas tinklo elementų dydžiui nusakyti apie singuliariuosius taškus, o kontrolinės sferos apie tam tikrą kreivę apibrėžimas taikomas nusakyti kraštinių sąlygų daromą įtaką generuojamam tinklui.

Kontrolinės erdvės sąvoka leidžia visiškai kontroliuoti generuojamo tinklo tankumą, tačiau dar lieka viena neišspręsta problema – norint tinkle išvengti blogos formos elementų, reikia įvertinti elementų kokybę. Paprastai tokie įvertinimai daromi imant įvairių elemento parametrų santykius: kraštinių ilgius, statmenis, kampus ir t. t. Nemažai mokslininkų (Babuška, 1976; Field, 2000) tyrinėjo įvairius elementų kokybės matavimus. Šiame darbe naudojamas (Bank, 1996; Bhatia, Lawrence, 1990) apibrėžtas elemento formos įvertis, kuris dėl savo sferinių kontūrų yra suderinamas su minėtu kontrolinės sferos apibrėžimu ir leidžia išvengti netaisyklingos formos elementų ir staigių elemento dydžio pokyčių generuojamame tinkle:

$$Q_K = \frac{4\sqrt{3}A}{l_1^2 + l_2^2 + l_3^2}, \quad (2.6)$$

čia l_1 , l_2 ir l_3 – nagrinėjamo trikampio K kraštinės, A – trikampio plotas.

Taigi pora (D, S) , apimanti įvairią informaciją apie kraštinės sąlygas, medžiagos savybes ir geometrinius srities ypatumus, leidžia sukonstruoti pradinį beveik optimalų baigtinių elementų tinklą, kuris dar prieš atliekant baigtinių elementų analizę yra labiau sutankinamas tose srityse, kur tikimasi gauti didesnius įtempimų gradientus. Tokiu būdu BE analizė pradedama jau nuo beveik optimalaus tinklo, kas leidžia tikėtis, kad reikės mažiau sąnaudų galutiniam rezultatui gauti.

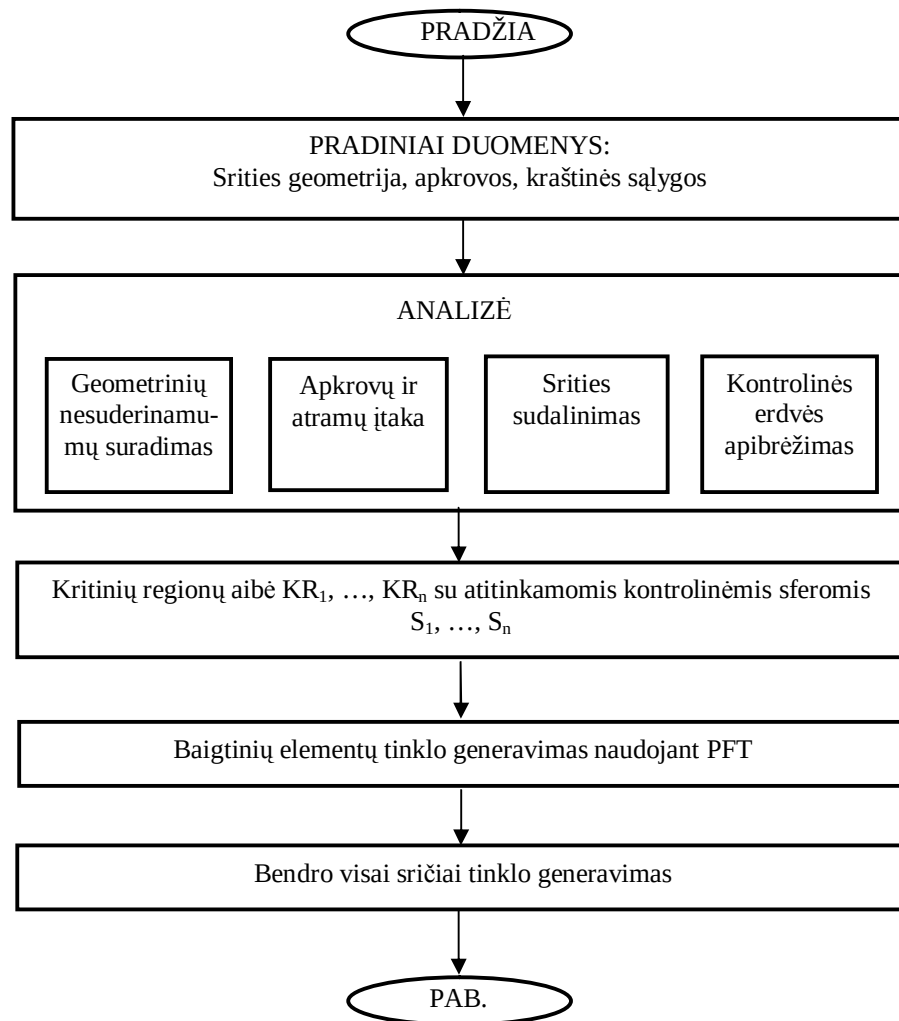
2.2 Pradinio beveik optimalaus tinklo generavimo algoritmas

Šiame skyriuje aprašysime algoritmą, leidžiantį generuoti puikios kokybės pradinį tinklą nagrinėjamai plokštuminei sričiai. Algoritmas tinklo generavimą pradeda nuo uždavinio analizės bei duomenų paruošimo PFT trianguliacijai – visų pirma, sudėtinga pradinė sritis suskaidoma į keletą paprastesnių dalių, tada iš turimų kraštinių sąlygų bei medžiagos savybių kiekvienai gautai srities daliai yra konstruojamos kontrolinės erdvės bei generuojami pradiniai beveik optimalūs tinklai. Algoritmas darbą baigia sujungdamas gautus pradinius beveik optimalius tinklus į vieną tinklą.

Pagrindiniai šio algoritmo žingsniai yra tokie (2.3 pav.):

1. Pradinis paruošimas: srities geometrijos, kraštinių sąlygų bei medžiagos savybių apibrėžimas.
2. Analizės dalis:

- (a) Srities geometrijos tikrinimas: nuosekliai peržiūrint visus geometrinius primityvus (taškus, atkarpas, lankus ir t. t.), nustatome, kurie iš jų galėtų būti pripažinti kritiniais regionais (kritiniais taškais, atkarpomis, kreivėmis). Paprastai kritinėmis pripažįstamos tos sritys, kurios turi įvairios



2.3 pav. Algoritmo schema

formos skylių, įtrūkimų, kampų ir kitų geometrinių nesuderinamumų, kuriuose galime tikėtis gauti didesnius įtempimų gradientus. Suformuojame sąrašą $\{KR_1, KR_2, \dots, KR_n\}$, čia KR_i nusako i -ąjį regioną, o n yra bendras tokių regionų skaičius.

- (b) Kraštinių sąlygų analizė: patikriname apkrovų bei atramų įtaką visiems įtartiniams kritiniams regionams KR_i , jei nėra tokios apkrovos ar atramos, kuri duotų ypatingai didelę įtempimų koncentraciją tame regione, išmetame šią KR_i iš kritinių regionų sąrašo.
- (c) Srities skaidymas: daliname originalią sritį į dalis taip, kad kiekvienoje gautoje dalyje būtų tik vienas kritinis taškas ar regionas. Formuojame sąrašą $\{\{D_1, KR_1\}, \{D_2, KR_2\}, \dots, \{D_m, KR_m\}\}$, čia D_i – einamoji sritis, o KR_i yra šios srities kritinis regionas, $m \leq n$ yra tikrai kritinių sričių skaičius.
- (d) Kontrolinės erdvės apibrėžimas: pagal 4 apibrėžimą kiekvienai sričiai D_i apibrėžiame kontrolinę sferą CS_i . Sudarome sąrašą $\{\{D_1, KR_1, KS_1\}, \{D_2, KR_2, KS_2\}, \dots, \{D_m, KR_m, KS_m\}\}$, kuriame kiekviena kontrolinė sfera CS_i laiko atitinkamą informaciją apie nagrinėjamos srities dalies D_i savybes bei elementų dydžio pasiskirstymą toje dalyje D_i .

3. Tinklo generavimas: taikydami plintančio fronto technikos algoritmą, generuojame pradinį beveik optimalų pradinį tinklą T_i kiekvienai srities daliai D_i :

(a) Naudodami informaciją, užkoduotą kontrolinėje erdvėje CS_i , apibrėžiame pradinį kraštą F_i PFT trianguliacijai.

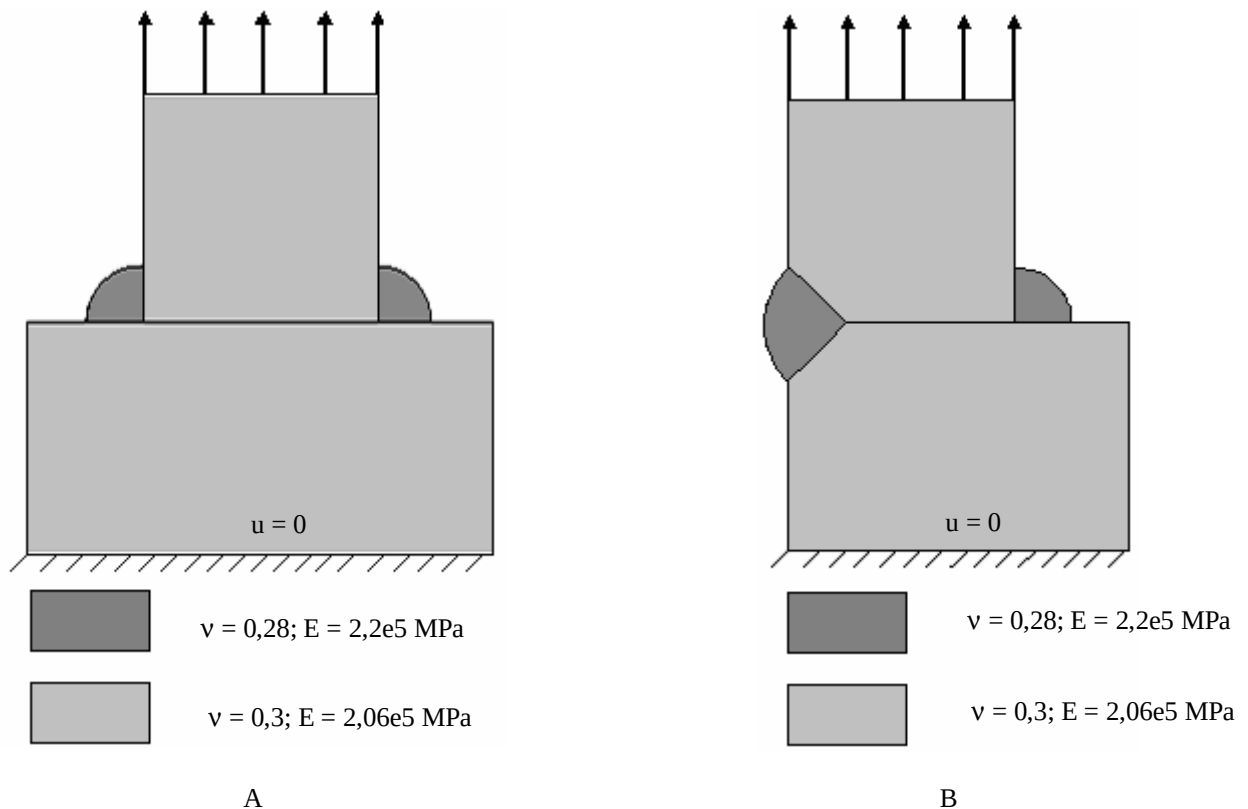
(b) Nagrinėjame kraštą F_i :

- pagal tam tikrus kriterijus, pasirenkame krašto elementą f ;
- sukuriame optimalų tašką P_{opt} elementui f ;
- patikriname, ar jau egzistuoja toks tinklo taškas V , kuris galėtų būti naudojamas vietoje taško P_{opt} . Jei taip, tai P_{opt} priskiriame tašką V ;
- iš krašto elemento f ir P_{opt} sugeneruojame naują elementą K ;
- patikriname, ar sugeneruotasis elementas atitinka elemento dydžio reikalavimus.

(c) Jei krašte yra dar nenaudotų elementų, grįžtame į (b);

4. Galutinio tinklo generavimas: gautus pradinius tinklus $\{T_1, T_2, \dots, T_n\}$ sujungiame ir gauname bendrą pradinį tinklą T , dengiantį visą nagrinėjamą sritį Ω .

2.3 Mechaninių konstrukcijų modeliavimas taikant pasiūlytą algoritmą



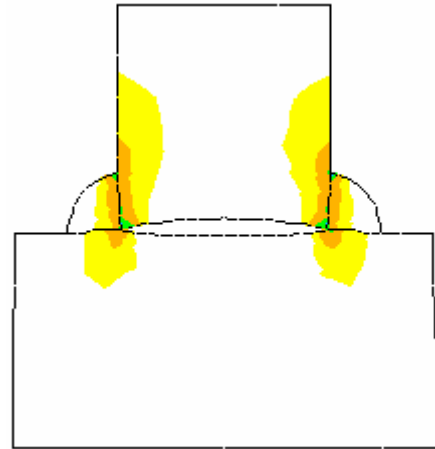
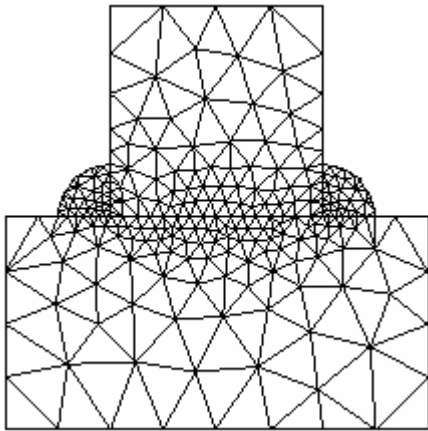
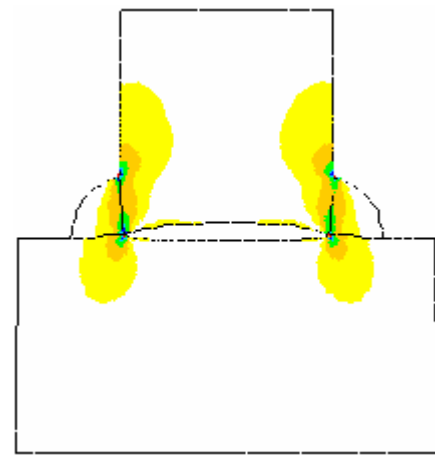
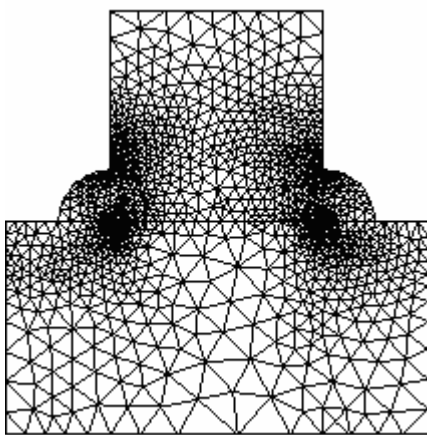
2.4 pav. Konstrukcijų A ir B, susidedančių iš dviejų skirtingų medžiagų, schemas

Pasiūlyto pradinio beveik optimalaus tinklo generavimo algoritmo pranašumo demonstravimui parinkime dvi skirtingas mechanines konstrukcijas, susidedančias iš dviejų plieninių medžiagų: pirmoji medžiaga aprašoma Jungo moduliu $E = 2,06 \cdot 10^5$ MPa ir Puasono koeficientu $n = 0,3$, antroji medžiaga atitinkamai nusakoma $E = 2,2 \cdot 10^5$ MPa ir $n = 0,28$. Kiekvienos konstrukcijos viršuje uždedame $p = 1000$ kN apkrovą. Uždavinių geometrija bei kraštinės sąlygos atvaizduotos 2.4 paveiksle. Abiems uždaviniams parenkame leistiną sprendinio diskretizacijos paklaidą $h = 10$ %.

Taikant pasiūlytą pradinio beveik optimalaus tinklo generavimo algoritmą vartotojo sąveika apsiriboja tik objekto geometrijos apibrėžimu, medžiagos bei kraštinių sąlygų nusakymu. Pradiniam beveik optimaliam tinklui generuoti naudojama praplėsta kontrolinė erdvė, kuri informaciją apie elemento dydį gauna iš kontrolinių sferų apie tašką ir apie atkarpą. Visų pirma informaciją apie skirtingas medžiagas įtraukiame į elemento dydžio nustatymo funkciją, o tada generuojame konstrukciją A ir B pradinius beveik optimalius tinklus. Gauname, kad konstrukcija A turi 490 elementų (2.5 pav.), o konstrukcija B – 415 (2.7 pav.). Palyginimui abiem konstrukcijoms generuojame tolygius pradinius tinklus, turinčius tą patį elementų skaičių (2.6 pav. ir 2.8 pav.). Iš viso nagrinėjame keturis tinklus, pagal kuriuos atliekame baigtinių elementų analizę ir surandame atitinkamus sprendinius. Lygindami gautus duomenis, matome, kad Mizeso įtempimai, gauti taikant pradinį beveik optimalų tinklą (2.5 pav.), yra labiau sukoncentruoti dviejų medžiagų sąlyčio vietose, nei įtempimai, gauti naudojant tolygų tinklą (2.6 pav.). Vadinasi, taikydami pradinį beveik optimalų tinklą, jau per pirmąją prisitaikančiųjų strategijų žingsnį galime tikėtis mažesnės paklaidos, o tai reiškia, kad visam procesui užbaigti reikės mažiau iteracijų, nei naudojant tolygų tinklą.

Norėdami galutinai įsitikinti, kad pradinio beveik optimalaus tinklo parinkimas prisitaikančiųjų baigtinių elementų analizei tikrai leidžia sumažinti iteracijų skaičių, konstrukcijoms A ir B suraskime galutinius optimalius tinklus dviem atvejais: pirmai iteracijai parinkdami pradinį beveik optimalų tinklą ir pradinį tolygų tinklą. Skaičiavimų rezultatus surašome į 2.1 lentelę. Baigtinių elementų tinklai bei juos atitinkančios įtempimų pasiskirstymo diagramos pavaizduotos 2.5, 2.6, 2.7 ir 2.8 paveiksluose.

Pradėkime nuo konstrukcijos A. Baigtinių elementų sprendinius pradinio beveik optimalaus tinklo ir pradinio tolygaus tinklo atvejais šiai konstrukcijai mes jau suskaičiavome, dabar remdamiesi gautais sprendiniais, įvertiname paklaidas. Pradinio beveik optimalaus tinklo atveju gauname, kad paklaida yra $\bar{h}_{opt} = 28,63$ %, o tolygaus pradinio tinklo atveju gauname, kad $\bar{h} = 43,46$ %. Matome, kad tolygaus pradinio tinklo atveju paklaidos reikšmė yra didesnė, nei pradinio beveik optimalaus tinklo atveju, tačiau abiem atvejais gautosios paklaidos yra didesnės, nei leistina vartotojo paklaida, todėl perskaičiuojame naujas elemento dydžio funkcijas ir generuojame naujus tinklus. Tada vėl atliekame baigtinių elementų analizę ir įvertiname diskretizacijos paklaidas. Po antrosios iteracijos gauname, kad pradinio beveik

Pirmoji iteracija**Antroji iteracija**

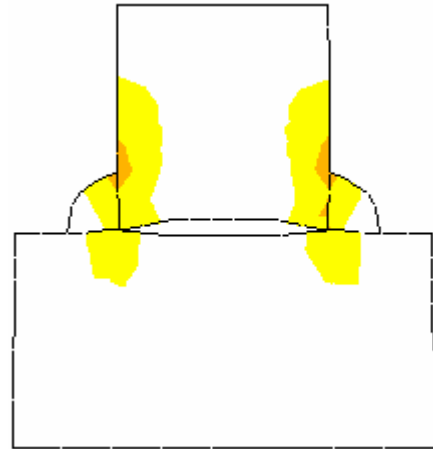
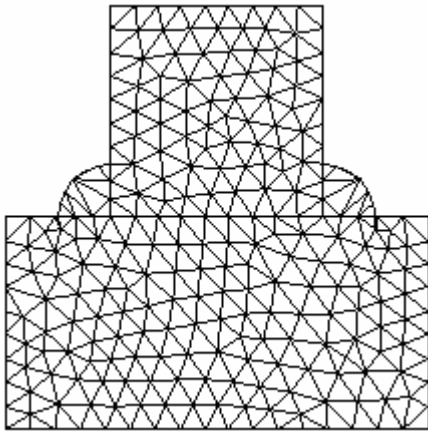
2.5 pav. Pradinio beveik optimalaus tinklo atvejis konstrukcijai A: BE tinklai ir įtempimų pasiskirstymo diagramos.

optimalaus tinklo atveju $\bar{h}_{opt} = 9,22 \%$, t.y. leistina vartotojo paklaida jau tenkinama, todėl skaičiavimus šiam tinklui stabdome. Tuo tarpu tolygaus pradinio tinklo atveju $\bar{h} = 15,18 \%$, todėl atliekame trečiąją prisitaikančiųjų baigtinių elementų analizės iteraciją. Lygindami gautus rezultatus, matome, kad pradinio beveik optimalaus tinklo atveju galutinio optimalaus tinklo gavimui prireikė dviejų iteracijų, o tolygaus pradinio tinklo atveju dviejų iteracijų galutinio tinklo radimui buvo mažai.

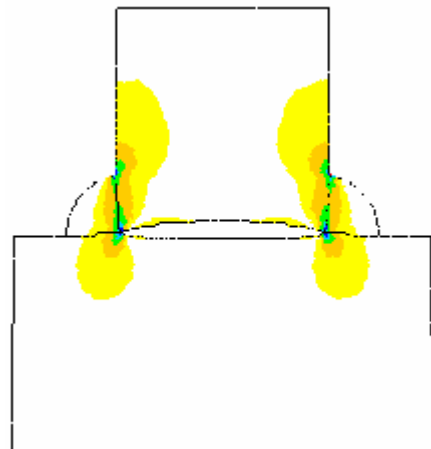
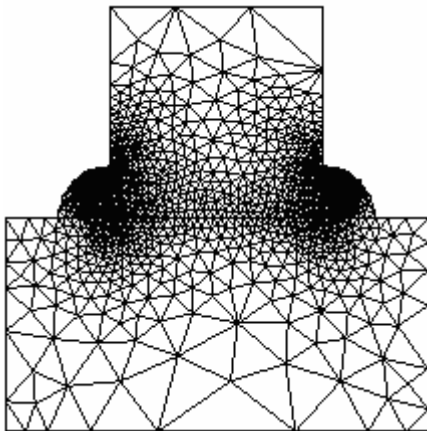
2.1 lentelė. Mechaninių konstrukcijų A ir B skaičiavimo rezultatai, leidžiama paklaida $h = 10 \%$

		1-oji iteracija		2-oji iteracija		3-oji iteracija	
		$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
A	Tolygus pradinis tinklas	43,46	450	15,18	3707	9,45	6344
A	Pradinis beveik optimalus tinklas	28,63	450	9,22	5089	-	-
B	Tolygus pradinis tinklas	25,61	490	18,15	2127	9,98	6117
B	Pradinis beveik optimalus tinklas	23,12	490	9,31	3599	-	-

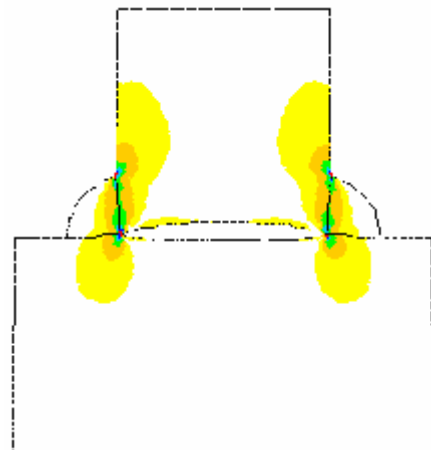
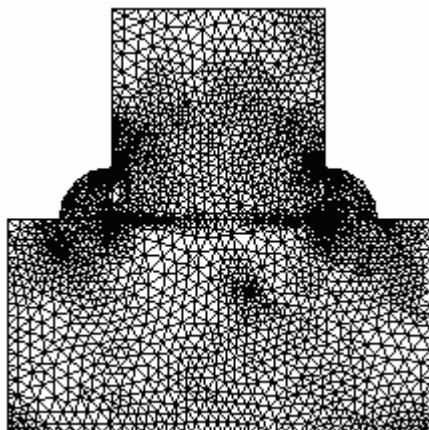
Pirmoji iteracija



Antroji iteracija



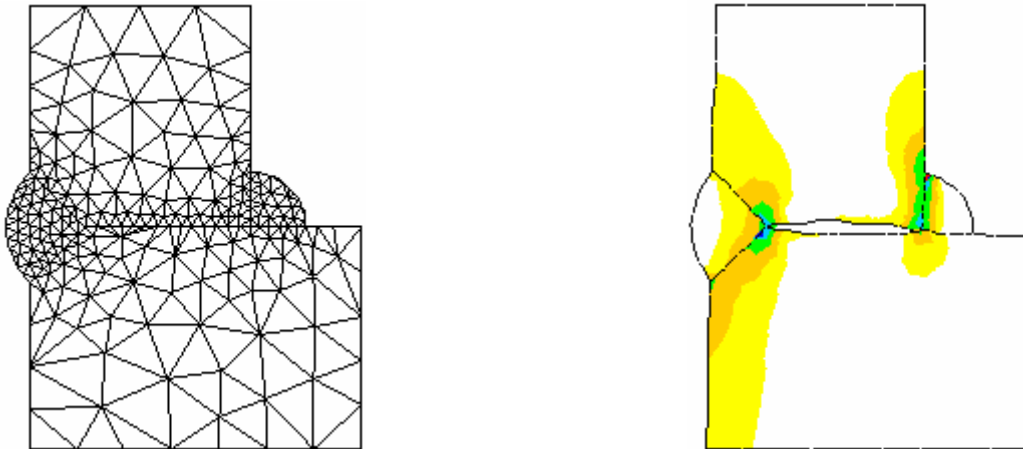
Trečioji iteracija



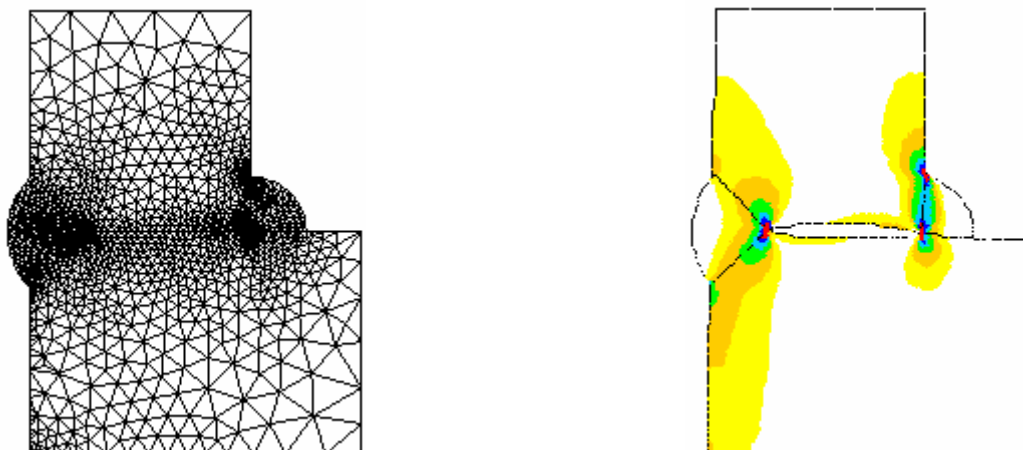
2.6 pav. Tolygaus pradinio tinklo atvejis konstrukcijai A: BE tinklai ir įtempimų pasiskirstymo diagramos.

Analogiškus skaičiavimus atliekame konstrukcijai B. Įvertinus paklaidas pirmoje iteracijoje gauname, kad pradinio beveik optimalaus tinklo atveju paklaida $\bar{h}_{opt} = 23,12 \%$, o tolygaus pradinio tinklo atveju $\bar{h} = 25,61 \%$. Nors paklaidą, gautą parinkus pradinį beveik optimalų tinklą ir yra šiek tiek artimesnė vartotojo nustatytai leistinai paklaidai, tačiau $\bar{h} > h$ sąlyga nėra tenkinama, todėl abiem atvejais atliekame antrą prisitaikančiųjų baigtinių elementų analizės iteraciją: vėl perskaičiuojame naują elemento dydžio funkciją, generuojame naują tinklą bei apskaičiuojame baigtinių elementų sprendinį. Gauname, kad po antros iteracijos pradinio beveik optimalaus tinklo atveju paklaida $\bar{h}_{opt} = 9,31 \%$, o tolygaus tinklo atveju paklaida yra $\bar{h} = 18,15 \%$. Matome, kad parinkus pradinį beveik optimalų tinklą, galutinio tinklo suradimui prireikė tik dviejų iteracijų. Atliekant prisitaikančiųjų baigtinių elementų analizę su pradiniu tolygiu tinklu, galutinio optimalaus tinklo radimui reikia trečios iteracijos.

Pirmoji iteracija

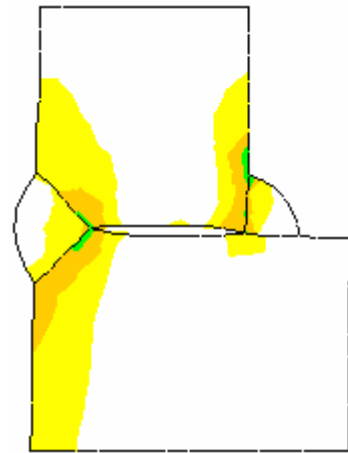
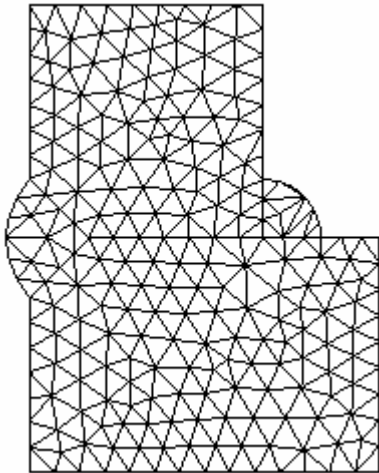


Antroji iteracija

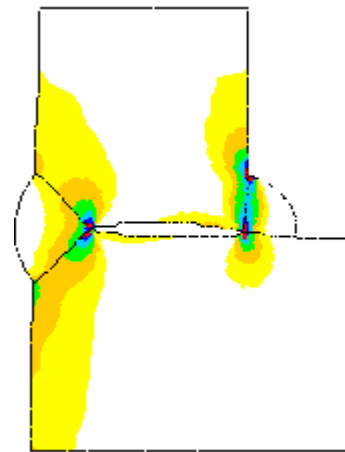
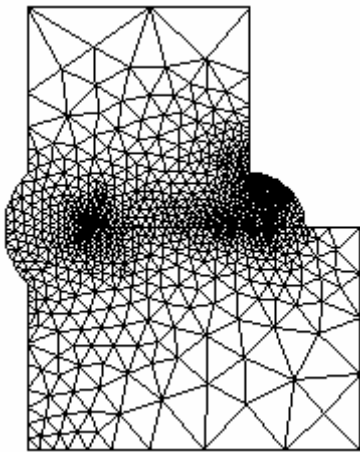


2.7 pav. Pradinio beveik optimalaus tinklo atvejis konstrukcijai B: BE tinklai ir įtempimų pasiskirstymo diagramos.

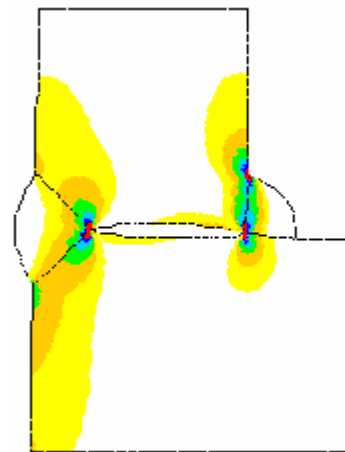
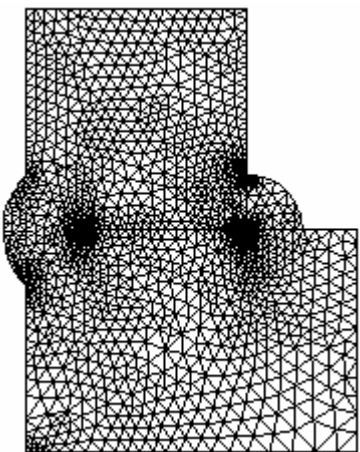
Pirmoji iteracija



Antroji iteracija



Trečioji iteracija



2.8 pav. Tolygaus pradinio tinklo atvejis konstrukcijai B: BE tinklai ir įtempimų pasiskirstymo diagramos.

Atlikę baigtinių elementų analizę bei įvertinę diskretizacijos paklaidas, matome, kad, kaip ir buvo tikėtasi, taikant pradinį beveik optimalų tinklą jau pirmoje iteracijoje diskretizacijos paklaida gaunasi artimesnė vartotojo leistinai paklaidai, nei taikant pradinį tolygų tinklą. Pagal pradinį beveik optimalų tinklą, artimesnes paklaidas leistinai reikšmei gauname ir antroje bei trečioje iteracijose, lyginant su tolygiu pradiniu tinklu. Be to, gauname mažesnę elementų skaičių galutiniame tinkle. Vadinasi, skaičiavimo sąnaudos sumažėjo.

2.4 Išvados

Šiame skyriuje pasiūlytas nauja strategija, leidžianti dar prieš pradedant baigtinių elementų analizę sugeneruoti pradinį beveik optimalų baigtinių elementų tinklą. Atlikti skaičiavimai parodė, kad:

- prisitaikančiųjų baigtinių elementų analizės pirmai iteracijai parinkus pradinį beveik optimalų tinklą, sugeneruotą pagal pasiūlytą metodą, galutinio optimalaus tinklo gavimui reikia tik dviejų iteracijų;
- prisitaikančiųjų baigtinių elementų analizę pradėjus nuo pradinio beveik optimalaus tinklo, elementų skaičius galutiniame tinkle sumažėja 20–40 %;
- pateikti pavyzdžiai rodo, kad pasiūlyta tinklo generavimo procedūra suteikia daugiau laisvės generuojant tinklus sudėtingoms konstrukcijoms, bei leidžia tikėtis, kad norimas paklaidos lygis bus pasiektas su mažesniu iteracijų ir elementų skaičiumi.

3. GLOBALIŲJŲ BEI LOKALIŲJŲ OPTIMALUMO KRITERIJŲ TAIKYMAS PRISITAIKANČIOSE BAIGTINIŲ ELEMENTŲ TINKLO GENERAVIMO PROCEDŪROSE

Diskretizacijos paklaidos įvertinimas bei tinkamo tinklo konstravimas taikant prisitaikančiąsias baigtinių elementų strategijas yra vieni iš svarbiausių baigtinių elementų uždavinių. Prisitaikančiųjų baigtinių elementų strategijų efektyvumas tiesiogiai priklauso nuo iteracijų, reikalingų optimaliam tinklui gauti, skaičiaus, todėl visų pirma turime atsakyti į klausimą – koks baigtinių elementų tinklas jau laikomas optimaliu? Formaliai optimalų tinklą galime apibrėžti taip: elementų skaičius optimaliame tinkle yra kaip galima mažesnis, o sprendinio paklaida neturi viršyti leistinos paklaidos ribos (Li, Bettess, 1995; Li ir kt., 1995). Vadinas, norėdami optimizuoti prisitaikančiųjų strategijų algoritmą, turime įvykdyti dvi sąlygas – sumažinti iteracijų, reikalingų optimaliam tinklui gauti, skaičių, bei kaip galima labiau minimizuoti elementų skaičių galutiniame tinkle atsižvelgdami į leidžiamą sprendinio paklaidą. Tokiu būdu gausime baigtinių elementų sprendinį, kuris ne tik tenkins jam parinktą paklaidą, bet ir atitiks tam tikrą su juo susijusį optimalų tinklą.

Optimaliam tinklui apibrėžti taikomos dvi metodologijos. Pirmoji optimalumo kriterijų metodika, dar kitaip vadinama globaliaisiais optimalumo kriterijais, sprendinio kokybę įvertina remdamasi paklaidos energetinės normos apibrėžimu (Zienkiewicz, Zhu, 1987; Onate, Bugeda, 1993; Fuenmayor, Oliver, 1996). Pagal šias strategijas globaliajai tinklo paklaidai kontroliuoti įvedami atskiri globalieji ir lokalieji parametrai. Antroji optimalumo kriterijų metodika paremta lokaliųjų įtempimų paklaidos įverčių bei elemento gerinimo parametrų skaičiavimu tam tikruose strategiškai svarbiuose srities taškuose (Bugeda, Onate, 1994). Tokie optimalumo kriterijai vadinami lokaliaisiais.

Šiame skyriuje pristatomas prisitaikančiųjų strategijų algoritmas, reikalaujantis mažiau iteracijų galutiniam tinklui gauti, lyginant su tradiciniu algoritmu. Tikslui pasiekti, 2.2 skyriuje pasiūlytą pradinio beveik optimalaus tinklo generavimo algoritmą įkomponuosime į tradicinį prisitaikančiųjų strategijų algoritmą. Toks patobulinimas leis sumažinti galutiniam tinklui gauti reikalingą iteracijų skaičių. Norėdami kontroliuoti elementų dydį bei tankumą galutiniame optimaliame tinkle, į prisitaikančiųjų strategijų algoritmą įkomponuosime skirtingus globaliuosius bei lokaliuosius optimalumo kriterijus. Skyriaus gale pateikti skaitiniai pavyzdžiai vaizdžiai iliustruos pateikto algoritmo privalumus lyginant su tradiciniu prisitaikančiųjų strategijų algoritmu.

Šios skyriaus tyrimų medžiaga 2003 metais pristatyta „Mechanika – 2003“ ir „Mathematical Modeling and Analysis“ konferencijose ir tais pačiais metais publikuota „Mechanikos“ žurnale (Vasiliauskienė, Baušys, 2003).

3.1 Sprendinio tinkamumo įvertinimas

Sakoma, kad sprendinys yra tinkamas, jei globalioji energijos paklaida yra mažesnė už tam tikrą bendrosios deformacijos energijos dalį:

$$\|e\| \leq h \|u\|, \quad (3.1)$$

čia η – vartotojo užduota leidžiama santykinė globalioji paklaida. Remiantis šia lygybe, globalusis paklaidos vertinimo parametras x_g apibrėžiamas taip:

$$x_g = \frac{\|e\|}{h \|u\|}. \quad (3.2)$$

Jei $x_g \leq 1$, tai reiškia, kad globalusis paklaidos kriterijus tenkinamas, o jei $x_g \geq 1$, tai reikalingas tolesnis tinklo smulkinimas ar stambinimas. Tinklo tobulinimo strategijos, besiremiančios x_g parametru, duos tolygų tinklo elementų smulkinimą ar stambinimą.

Elementų pasiskirstymas tinkle lemia tam tikro tinklo optimalumo kriterijaus atitikimą, kitaip tariant:

$$\|e\|_i = \|e\|_{r_i}, \quad (3.3)$$

čia $\|e\|_i$ – i -ojo elemento paklaida, o $\|e\|_{r_i}$ – reikiama elemento paklaida. Tada lokalusis paklaidos indikatorius $\bar{x}_i$ apibrėžiamas kaip $\|e\|_i$ ir $\|e\|_{r_i}$ santykis:

$$\bar{x}_i = \frac{\|e\|_i}{\|e\|_{r_i}}. \quad (3.4)$$

Jei $\bar{x}_i = 1$, tai i -asis elementas yra optimalus, o jei ne, tai reikalinga papildoma tinklo tobulinimo procedūra. Reikiamos paklaidos $\|e\|_{r_i}$ apibrėžimas kiekvienam tinklo elementui yra labai svarbus tolesnei baigtinių elementų analizei, nes, kaip matysime kitose dalyse, šios paklaidos pagrindu yra apibrėžiami kiti optimalumo kriterijai, stipriai veikiantys elementų dydį bei tankumą galutiniame tinkle.

Bendruoju atveju turi būti tenkinami tiek lokalusis, tiek globalusis paklaidų vertinimo kriterijai. Ši prielaida leidžia suformuluoti tinklo tobulinimo parametro apibrėžimą, taikant tiek globalųjį paklaidos vertinimo kriterijų, tiek lokalųjį paklaidos vertinimo indikatorius:

$$x_i = \bar{x}_i x_g = \frac{\|e\|_i}{\|e\|_{r_i}} \cdot \frac{\|e\|}{h \|u\|} = \frac{\|e\| \|e\|_i}{h \|u\| \|e\|_{r_i}}. \quad (3.5)$$

Šį elemento tobulinimo parametą pirmasis pristatė Zienkiewicz ir Zhu (Zienkiewicz, Zhu, 1987) ir nuo to laiko jis laikomas pagrindu apibrėžiant naują elemento dydį prisitaikančiose tinklo tobulinimo strategijose.

3.2 Globalieji ir lokalieji optimalumo kriterijai

Vienas populiariausių eliptinių uždavinių optimalumo kriterijų yra apibrėžiamas taip: tinklas yra optimalus, jei energetinės normos pasiskirstymas yra tolygus visų elementų atžvilgiu (Zhu, Zienkiewicz, 1988). Remiantis šiuo apibrėžimu, reikiama elemento paklaida apibrėžiama kaip globaliosios paklaidos ir tinklo elementų santykis:

$$\|e\|_{r_i} = \frac{\|e\|}{\sqrt{n}}. \quad (3.6)$$

Tada lokalusis paklaidos vertinimo parametras apibrėžiamas taip:

$$\bar{x}_i = \frac{\|e\|_i}{\|e\|\sqrt{n}} \quad (3.7)$$

ir elemento gerinimo parametras išreiškiamas lygybe

$$x_i = \bar{x}_i x_g = \frac{\|e\|_i}{g\|u\|\sqrt{n}}. \quad (3.8)$$

Dydį x_i dabar galime interpretuoti kaip elemento paklaidos ir leidžiamos paklaidos pasiskirstymo tinkle santykį. Jei $x_i > 1$, tai reiškia, kad elemento dydį reikia dar kartą perskaičiuoti, o jei $x_i \leq 1$, tai reiškia, kad tiek lokaliosios, tiek globaliosios paklaidos sąlygos yra tenkinamos.

Jei žinome einamąjį elemento dydį h_i , tai naujas elemento dydis $\bar{h}_i$ yra perskaičiuojamas taip (Zhu, Zienkiewicz, 1988):

$$\bar{h}_i = \frac{h_i}{b_i}, \quad b_i = (x_i)^{\frac{1}{m}}, \quad (3.9)$$

čia m – formos funkcijos daugianario laipsnis.

Kitas globalusis paklaidos vertinimo kriterijus teigia, kad tinklas yra optimalus, jei paklaidos kvadratas ploto vienetui yra vienodas visame tinkle:

$$\frac{\|e\|_i^2}{\Omega_i} = \frac{\|e\|^2}{\Omega}, \quad (3.10)$$

čia Ω_i ir Ω yra atitinkamai elemento ir visos srities plotai.

Reikiamos paklaidos normos bei elemento dydžio perskaičiavimo parametrai šiam kriterijui apibrėžiami taip:

$$\|e\|_{r_i} = \|e\| \left(\frac{\Omega_i}{\Omega} \right)^{\frac{1}{2}} \quad (3.11)$$

ir

$$x_i = \bar{x} x_g = \frac{\|e\|_i}{h\|u\|} \left(\frac{\Omega}{\Omega_i} \right)^{\frac{1}{2}}. \quad (3.12)$$

Naujas elemento dydis randamas pagal tą pačią (3.9) formulę. Šis optimalumo kriterijus leidžia sukcentruoti daugiau ir mažesnių elementų tose zonose, kuriose sprendinio nežinomųjų gradientai yra didžiausi. Ši tolygos paklaidos pasiskirstymo kriterijų pirmasis pristatė ir struktūrinės analizės uždaviniams pritaikė Bugeda su bendraautoriais (Bugeda, Onate, 1992; Bugeda, Oliver, 1991).

Kita optimalumo kriterijų grupė remiasi lokaliuoju paklaidos įvertinimu ir yra vadinama lokaliaisiais optimalumo kriterijais. Jų esmę sudaro tam tikrų specifinių dydžių, skaičiuojamų tam tikrose srities vietose, radimas ir kontroliavimas apribojant juos tam tikrais iš anksto numatytais dydžiais.

Pirmasis tokių optimalumo kriterijų pavyzdys remiasi Mizeso įtempimais (Bugeda, 2000, 2001):

$$e_{\bar{s}} = \sqrt{\frac{(s_1 - s_2)^2 + (s_2 - s_3)^2 + (s_3 - s_1)^2}{2}} - \sqrt{\frac{(\hat{s}_1 - \hat{s}_2)^2 + (\hat{s}_2 - \hat{s}_3)^2 + (\hat{s}_3 - \hat{s}_1)^2}{2}}, \quad (3.13)$$

čia s_1, s_2 ir s_3 – svarbiausieji poprocesorinio sprendinio įtempimai, $\hat{s}_1, \hat{s}_2, \hat{s}_3$ – svarbiausieji BE sprendinio įtempimai. Žinant maksimalią Mizeso įtempimų paklaidą $e_{\bar{s}_{max}}$, kiekviename tinklo taške elemento dydžio perskaičiavimo parametą galime apibrėžti taip:

$$x = \frac{e_{\bar{s}}}{e_{\bar{s}_{max}}}. \quad (3.14)$$

Antrasis lokaliųjų optimalumo kriterijų pavyzdys yra vadinamasis maksimalus įtempimų paklaidos procentinis santykis (Bugeda, Onate, 1994). Šiuo atveju nagrinėjame maksimalios įtempimų paklaidos ir maksimalaus įtempimo kiekviename taške santykį. Įtempimų paklaida e_s skaičiuojama kaip tikslaus ir aproksimuoto įtempimų tenzorijų skirtumas:

$$e_s = \begin{bmatrix} e_{s_x} & e_{t_{xy}} \\ e_{t_{xy}} & e_{s_y} \end{bmatrix} = \begin{bmatrix} s_x - \hat{s}_x & t_{xy} - \hat{t}_{xy} \\ t_{xy} - \hat{t}_{xy} & s_y - \hat{s}_y \end{bmatrix}. \quad (3.15)$$

Maksimali įtempimų paklaida e_{s_I} kiekviename taške yra apribota tam tikru maksimalaus svarbiausiojo įtempimo procentiniu santykiu $abs(s_I)$:

$$abs(e_{s_I}) \leq hab(s_I). \quad (3.16)$$

Šis kriterijus leidžia nagrinėti didesnes įtempimų paklaidas tose zonose, kur įtempimai yra didesni. Elemento dydžio perskaičiavimo parametras apibrėžiamas kaip santykis tarp $abs(e_{s_I})$ ir $hab(s_I)$:

$$x = \frac{abs(e_{s_I})}{hab(s_I)}. \quad (3.17)$$

Taikant lokaliuosius optimalumo kriterijus, sprendinio kokybė vertinama remiantis lokaliaisiais parametrais, o tai leidžia kontroliuoti tokius aiškiai fizikinę prasmę turinčius dydžius, kaip įtempimai kiekviename taške ir t. t.

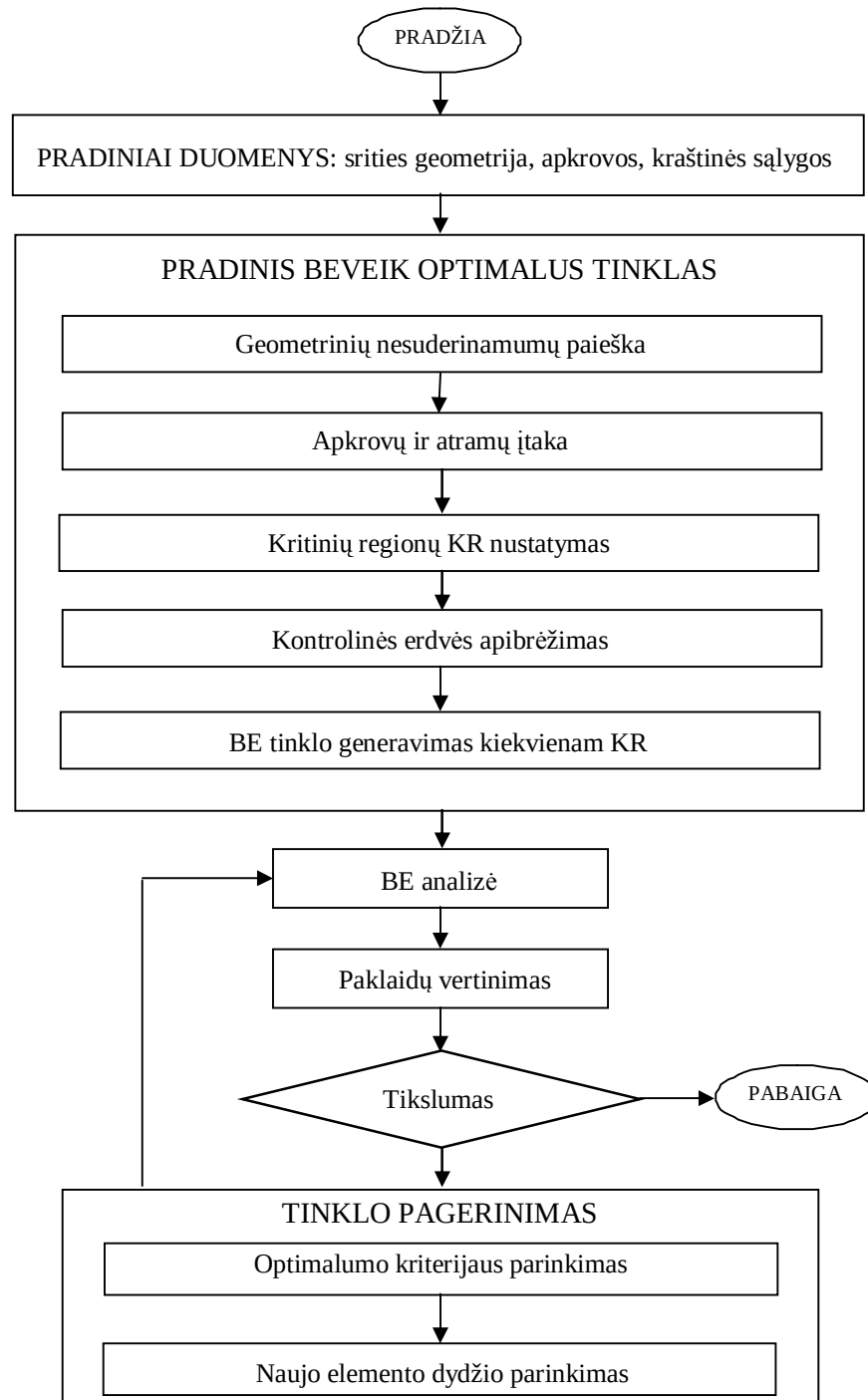
3.3 Optimalumo kriterijų įraukimas į baigtinių elementų tinklo generavimo procedūrą

Šioje dalyje pristatomas praplėstas tinklo gerinimo algoritmas, galintis ne tik minimaliu iteracijų skaičiumi sugeneruoti puikios kokybės galutinį baigtinių elementų tinklą, bet ir turintis vartotojo pageidaujamą elementų skaičių. Algoritmo pagrindą sudaro 1 skyriuje aprašytas prisitaikančiųjų baigtinių elementų strategijų algoritmas, papildytas dviem punktais. Pirmasis patobulinimas yra paruošti gerą beveik optimalų pradinį tinklą baigtinių elementų analizei atlikti, o kitas – parinkti atitinkamą optimalumo kriterijų, aprašytą 3.2 skyriuje. Kaip matysime toliau, pirmasis patobulinimas leis sumažinti prisitaikančiųjų strategijų proceso iteracijų skaičių, o antrasis – leis daryti įtaką galutinio tinklo elementų skaičiui bei tankumui.

Praplėtę tradicinį prisitaikančiųjų strategijų algoritmą, gausime tokią veiksmų seką (3.1 pav.):

1. Pradinio beveik optimalaus tinklo paruošimas tolesnei baigtinių elementų analizei.
 - a) Paruošiamoji informacija: nusakoma objekto geometrija, kraštinės sąlygos ir medžiagos savybės.
 - b) Analizės dalis: objekto geometrijos bei kraštinių sąlygų patikrinimas, srities sudalinimas bei kritinio regiono (KR) nustatymas, kontrolinės sferos (KS) apibrėžimas kiekvienam kritiniam regionui.
 - c) Tinklo generavimas: taikant plintančio fronto technikos metodą generuojamas pradinis beveik optimalus pradinis tinklas kiekvienam kritiniam regionui atskirai, tada gauti regionų tinklai sujungiami į vieną taip, kad padengtų visą nagrinėjamą sritį.
2. Baigtinių elementų sprendinio skaičiavimas paruoštam tinklui.
3. Tinklo optimalumo kriterijaus parinkimas tolesnei analizei.

4. Globaliosios ir lokalsios diskretizacijos paklaidų įvertinimas.
5. Globaliosios paklaidos patikrinimas. Jei ji tenkina leistiną paklaidą, skaičiavimus stabdome ir procedūrą baigiame, jei ne – naudodamiesi įvertintąja paklaida bei leidžiama paklaida perskaičiuojame naujam tinklui generuoti reikalingą informaciją.
6. Pageriname seną arba pergeneruojame naują tinklą ir grįžtame į antrąjį žingsnį.



3.1 pav. Praplėstasis optimalaus tinklo generavimo algoritmas

3.4 Praktinis globaliųjų ir lokaliųjų optimalumo kriterijų taikymas inžinerinių uždavinių sprendimui

Ankščiau paminėtam metodui iliustruoti parinksime tris skirtingo sudėtingumo uždavinius, turinčius skirtingas topologines savybes. Vartotojo sąveika su šiuo algoritmu apsiriboja objekto geometrijos bei kraštinių sąlygų apibrėžimu. Visi trys uždaviniai pradedami spresti naudojant tolygų pradinį tinklą. Skaičiavimo metu pritaikome keturis optimalumo kriterijus ir gauname keturis skirtingus galutinius tinklus. Analogišką procedūrą kartojame imdami beveik optimalų pradinį tinklą. Norėdami palyginti ir įvertinti gautus rezultatus, skaičiavimų duomenis surašome į lenteles bei iliustruojame grafiškai.

Rezultatams aprašyti įveskime tokius žymėjimus:

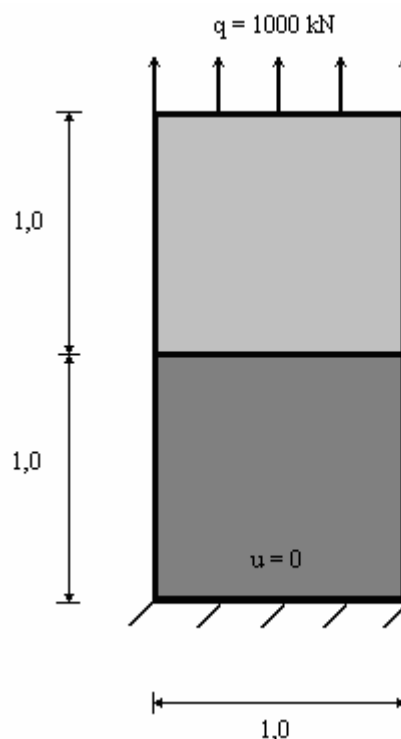
Kriterijus A – remiasi globaliuoju optimalumo kriterijumi, sakančiu, kad globaliosios energetinės normos pasiskirstymas yra tolygus visų elementų atžvilgiu;

Kriterijus B – remiasi globaliuoju optimalumo kriterijumi, sakančiu, kad tinklas yra optimalus, jei paklaidos kvadratas ploto vienetui yra vienodas visame tinkle;

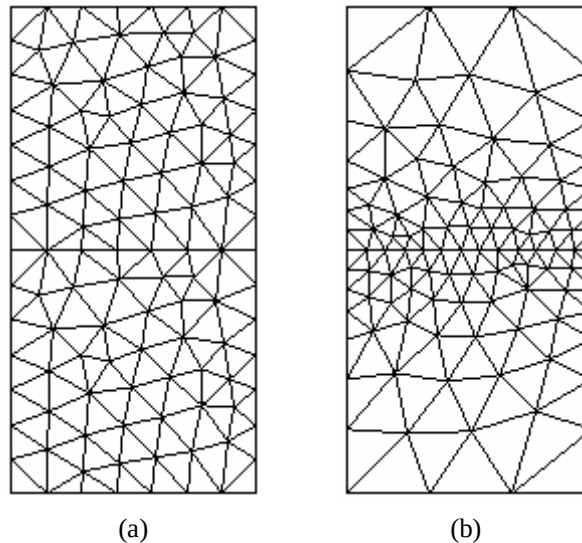
Kriterijus C – remiasi Mizeso įtempimais;

Kriterijus D – remiasi lokaliuoju optimalumo kriterijumi, skaičiuojamu kaip maksimalus įtempimų paklaidos procentinis santykis.

3.4.1 Srities, sudarytos iš dviejų skirtingų medžiagų, uždavinys



3.2 pav. Dviejų medžiagų uždavinys



3.3 pav. Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus tinklas

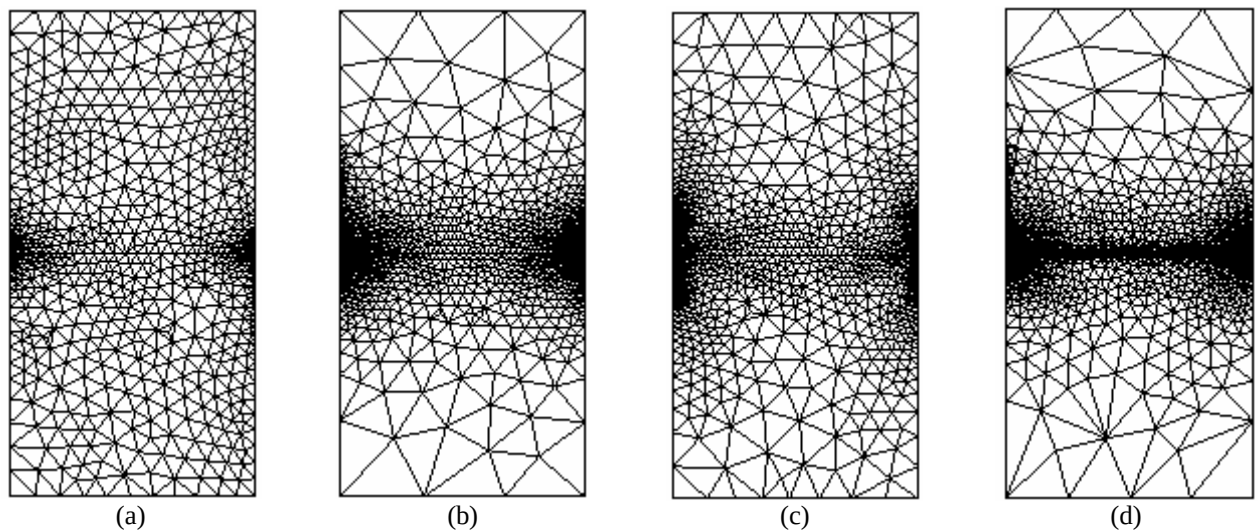
Visų pirma išnagrinėsime sritį, susidedančią iš dviejų skirtingų medžiagų. Pirmoji medžiaga nusakoma Jungo modulių $E = 2,06 \cdot 10^5$ MPa ir Puasono koeficientu $\nu = 0,28$. Atitinkamai antroji medžiaga nusakoma $E = 2,20 \cdot 10^5$ MPa ir $\nu = 0,30$. Srities viršus apkraunamas $q = 1000$ N dydžio tolygia apkrova. Nagrinėjamos srities dimensijos bei įtvirtinimo sąlygos parodytos 3.2 paveiksle.

Šiam uždaviniui spręsti paruošime tolygų pradinį baigtinių elementų tinklą, susidedantį iš 184 elementų (3.3 pav., a). Tada naudodamiesi praplėsto prisitaikančiųjų strategijų algoritmo schema bei parinkę optimalumo kriterijų A, gauname galutinį optimalų baigtinių elementų tinklą. Tą pačią procedūrą kartojame su optimalumo kriterijais B, C ir D. Gauti galutiniai baigtinių elementų tinklai pavaizduoti 3.4 paveikslo a, b, c ir d variantuose.

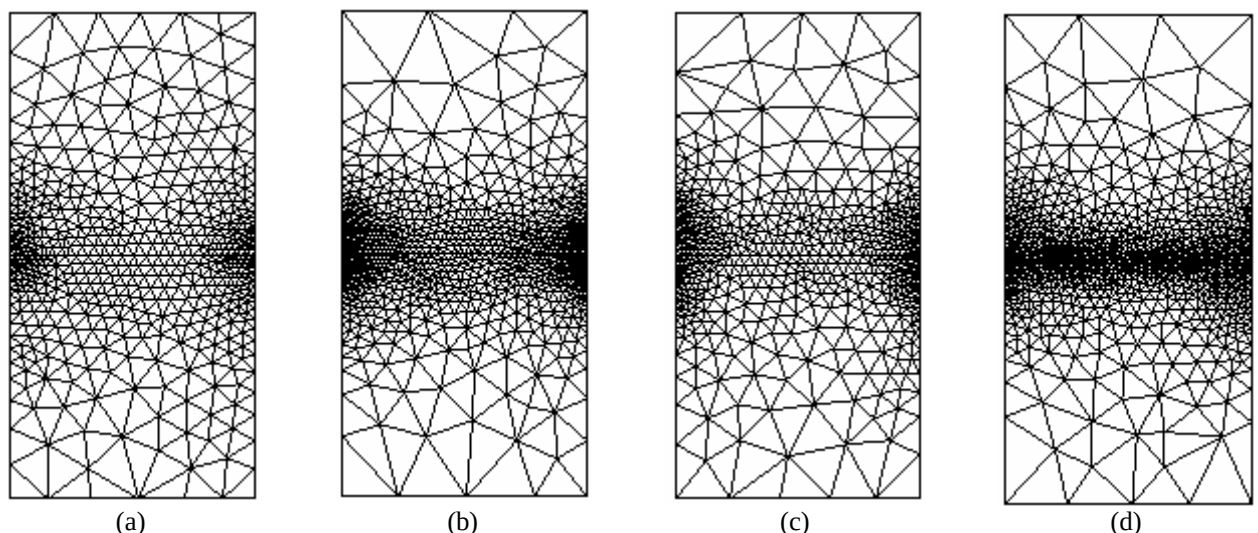
Dabar taikydami praplėstąją kontrolinės erdvės sąvoką, paruošime pagerintą beveik optimalų pradinį tinklą (3.3 pav., b), turintį tą patį elementų skaičių (184). Informacija apie elemento dydį gaunama konstruojant kontrolines sferas apie medžiagų sąlyčio sritis. Atlikę tą pačią prisitaikančiųjų strategijų procedūrą su visais keturiais optimalumo kriterijais, gauname keturis galutinius baigtinių elementų tinklus, pavaizduotus 3.5 paveikslo a, b, c ir d variantuose.

Skaiciavimų rezultatus surašome į 3.1 lentelę. Palyginę rezultatus, gautus taikant skirtingas strategijas, nesunkiai pastebime, kad optimalumo kriterijaus parinkimas stipriai veikia galutinį baigtinių elementų tinklą. Su tolygiu pradiniu tinklu optimalumo kriterijui A gauname 1476 elementus galutiniame tinkle, kriterijui B gauname 3847 elementus, kriterijui C – 2507 ir kriterijui D – 5800 elementus. Atitinkamai pradiniam beveik optimaliam tinklui gauname 1063, 2125, 1158 ir 2400 elementus galutiniame tinkle. Mažiausią elementų skaičių gauname taikydami tolygaus paklaidos pasiskirstymo optimalumo kriterijų – 1476 elementus pagal tolygų pradinį tinklą ir 1063 elementus pagal pradinį beveik optimalų tinklą. Didžiausią elementų skaičių šiam uždaviniui gavome taikydami optimalumo kriterijų D – 3,92 karto daugiau elementų nei pagal kriterijų A taikant tolygų pradinį tinklą ir 2,26 karto daugiau

elementų taikant pradinį beveik optimalų tinklą. Taip pat galima pastebėti, kad elementų skaičius pagal visus optimalumo kriterijus yra mažesnis, kai prisitaikančiųjų baigtinių elementų analizė atlikta parenkant pradinį beveik optimalų tinklą. Matome, kad elementų skaičius bei tankumas visuose tinkluose taip pat labai skiriasi – elementų dydis ir tankis tinkluose, gautuose taikant A optimalumo kriterijų kinta ne taip smarkiai, kaip tinkluose, gautuose pagal optimalumo kriterijus B, C ir D, kur daugiau smulkesnių elementų koncentruojama didesnių įtempimų gradientų zonose. Taigi, naudojant pagerintą beveik optimalų pradinį BE tinklą vietoj įprastinio tolygaus, galime sumažinti adaptyvaus proceso iteracijų skaičių, o parinkdami skirtingus optimalumo kriterijus, galime kontroliuoti elementų skaičių bei pasiskirstymą galutiniame tinkle.



3.4 pav. Galutiniai tinklai, gauti pagal tolygų pradinį tinklą: (a) – taikant optimalumo kriterijų A; (b) – taikant optimalumo kriterijų B; (c) – taikant optimalumo kriterijų C; (d) – taikant optimalumo kriterijų D.

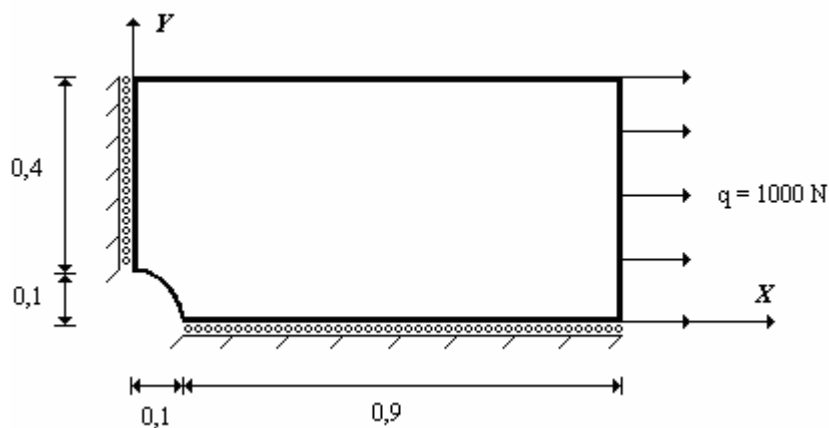


3.5 pav. Galutiniai tinklai, gauti pagal pradinį beveik optimalų tinklą: (a) – taikant optimalumo kriterijų A; (b) – taikant optimalumo kriterijų B; (c) – taikant optimalumo kriterijų C; (d) – taikant optimalumo kriterijų D.

3.1 lentelė. Dviejų medžiagų uždavinio skaičiavimo rezultatai, leidžiama paklaida $h = 10\%$

		1-oji iteracija		2-oji iteracija		3-oji iteracija	
		$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
A	Tolygus pradinis tinklas 1	27,85	92	10,03	603	9,88	726
	Tolygus pradinis tinklas 2	27,86	92	10,35	622	9,43	750
	Pradinis beveik optimalus tinklas 1	23,41	92	9,43	531	-	-
	Pradinis beveik optimalus tinklas 2	23,64	92	9,59	532	-	-
B	Tolygus pradinis tinklas 1	27,85	92	10,01	610	8,30	1798
	Tolygus pradinis tinklas 2	27,86	92	10,57	610	7,80	2049
	Pradinis beveik optimalus tinklas 1	23,41	92	8,25	1041	-	-
	Pradinis beveik optimalus tinklas 2	23,64	92	8,88	1084	-	-
C	Tolygus pradinis tinklas 1	27,85	92	11,61	405	7,63	1222
	Tolygus pradinis tinklas 2	27,86	92	13,36	360	7,66	1285
	Pradinis beveik optimalus tinklas 1	23,41	92	9,82	547	-	-
	Pradinis beveik optimalus tinklas 2	23,64	92	9,93	611	-	-
D	Tolygus pradinis tinklas 1	27,85	92	11,58	574	10,16	2619
	Tolygus pradinis tinklas 2	27,86	92	12,09	628	9,84	3181
	Pradinis beveik optimalus tinklas 1	23,41	92	9,39	1221	-	-
	Pradinis beveik optimalus tinklas 2	23,64	92	9,10	1179	-	-

3.4.2 Tempiamos stačiakampės plokštės su skylė uždavinys



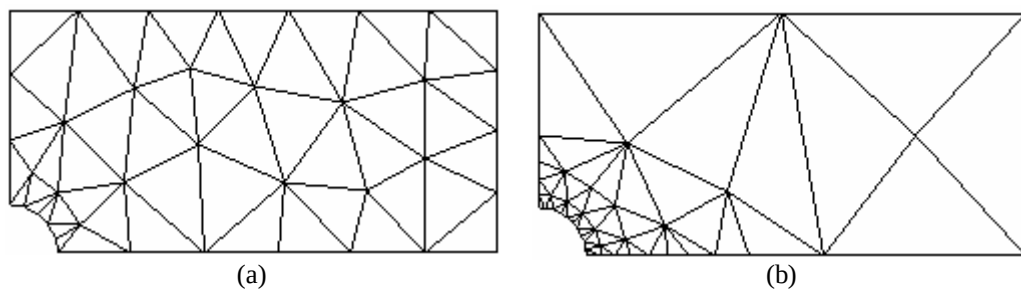
3.6 pav. Stačiakampės plokštės su skylė uždavinys

Rezultatams patvirtinti nagrinėjame stačiakampę plokštelę su skylė, kuri tempiama X ašies kryptimi. Medžiagos savybės aprašomos Jungo moduliu $E = 1000$ Pa ir Puasono koeficientu $\nu = 0,3$. Uždavinio dimensijos bei kraštinės sąlygos pavaizduotos 3.6 paveiksle.

Kaip ir ankstesniame pavyzdyje, taikome prisitaikančiųjų strategijų algoritmą pagal tolygų ir pagal pagerintą beveik optimalų tinklą, parinkdami visus keturis optimalumo kriterijus. Gauti duomenys

L. Vasiliausienė. Pristatantiosios baigtinių elementų strategijos plokštuminiams tamprumo teorijos uždaviniams

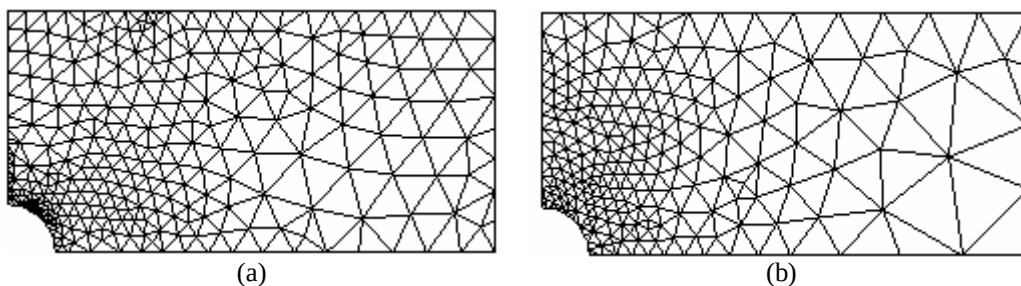
surašomi į 3.2 lentelę, galutiniai tinklai, gauti pagal tolygų tinklą, pavaizduoti 3.8, 3.9, 3.10 ir 3.11 paveikslų *a* variantuose, o tinklai, gauti pagal pagerintą beveik optimalų tinklą – 3.8, 3.9, 3.10 ir 3.11 paveikslų *b* variantuose. Lygindami gautus rezultatus, matome, kad imant pagerintą beveik optimalų pradinį baigtinių elementų tinklą, galutiniam tinklui gauti reikia tik dviejų iteracijų, o taikant kriterijų A, gauname mažiausią elementų skaičių tame galutiniame tinkle – 532 elementus pagal tolygų pradinį tinklą ir 391 elementus pagal pradinį beveik optimalų tinklą. Imdami pradinį beveik optimalų tinklą didžiausią elementų skaičių šiam uždaviniui gavome taikydami optimalumo kriterijų C – 3,64 karto daugiau elementų nei pagal kriterijų A, o imdami tolygų pradinį tinklą didžiausią elementų skaičių gavome optimalumo kriterijui B – 5,46 karto daugiau daugiau elementų nei taikant optimalumo kriterijų A.



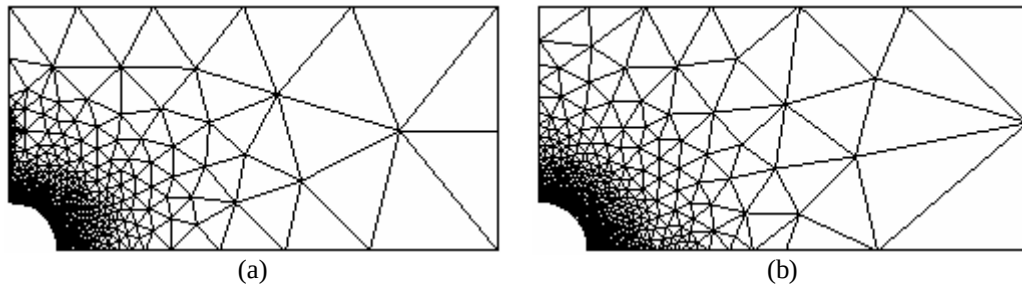
3.7 Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus pradinis tinklas

3.2 lentelė. Stačiakampės plokštės su skylė skaičiavimo rezultatai, leidžiama paklaida $h = 4\%$

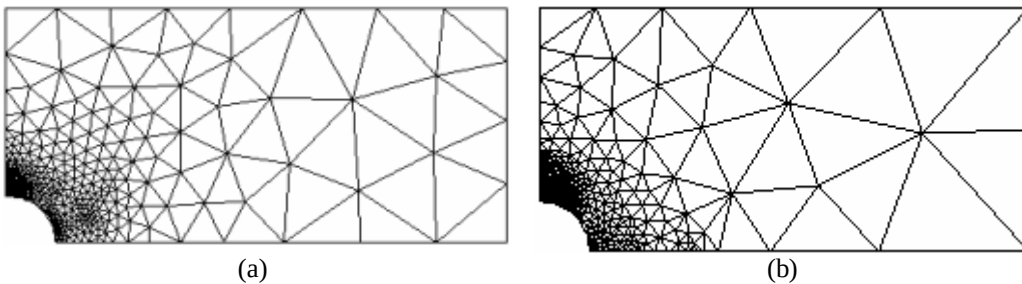
		1-oji iteracija		2-oji iteracija		3-oji iteracija	
		$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
A	Tolygus pradinis tinklas	19,49	51	4,99	379	3,24	532
	Pradinis beveik optimalus tinklas	13,08	51	3,47	391	-	-
B	Tolygus pradinis tinklas	19,49	51	4,61	915	2,58	2906
	Pradinis beveik optimalus tinklas	13,08	51	2,61	1359	-	-
C	Tolygus pradinis tinklas	19,49	51	5,44	198	2,58	2001
	Pradinis beveik optimalus tinklas	13,08	51	3,41	1424	-	-
D	Tolygus pradinis tinklas	19,49	51	8,21	902	3,94	1210
	Pradinis beveik optimalus tinklas	13,08	51	3,36	609	-	-



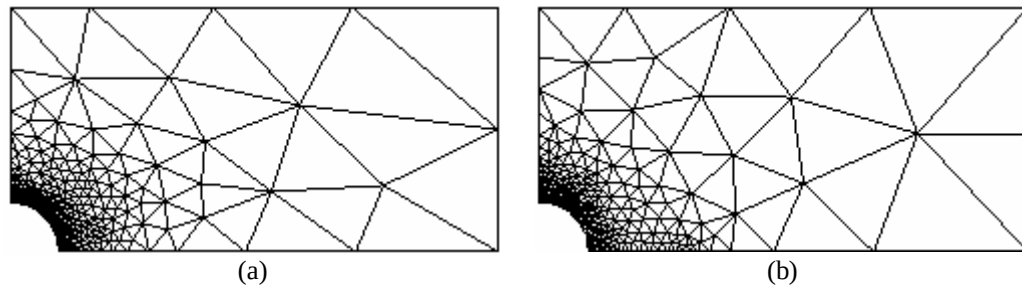
3.8 pav. Galutiniai tinklai, gauti pagal kriterijų A: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui



3.9 pav. Galutiniai tinklai, gauti pagal kriterijų B: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui



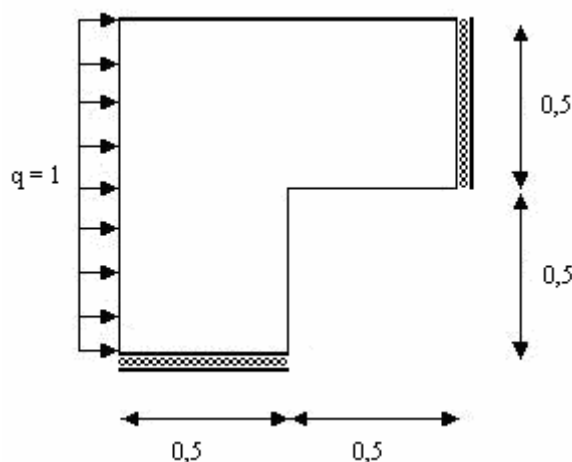
3.10 pav. Galutiniai tinklai, gauti pagal kriterijų C: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui



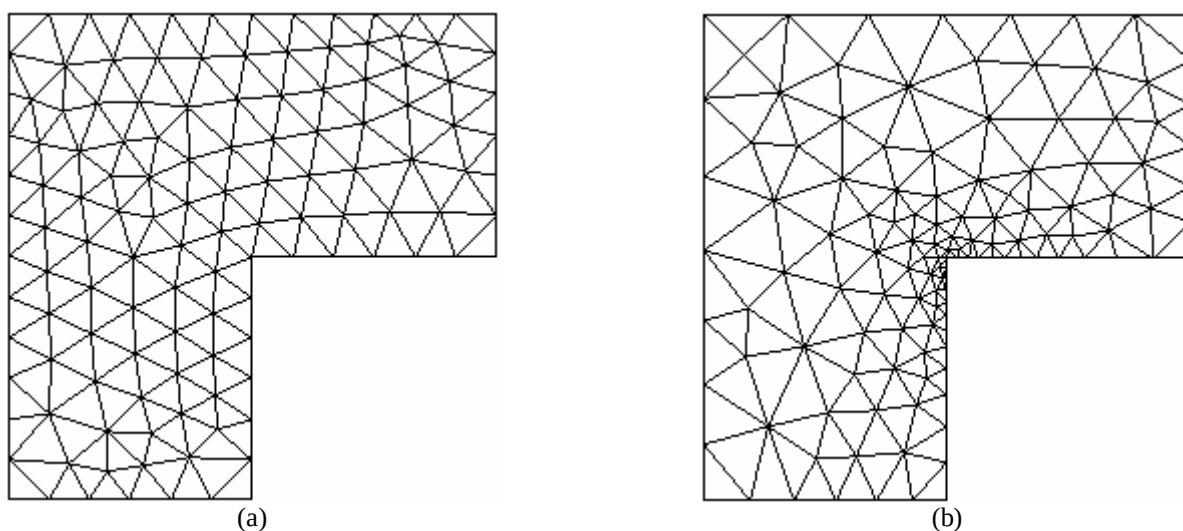
3.11 pav. Galutiniai tinklai, gauti pagal kriterijų D: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui

3.4.3 L-formos plokštės uždavinys

Imkime dar vieną pavyzdį: L-formos plokštę, kurios medžiagos savybės nusakomos Jungo modulių $E = 10^5$ Pa ir Puasono koeficientu $\nu = 0,3$. Plokštės dimensijos, įtvirtinimo sąlygos bei apkrovos pavaizduotos 3.12 paveiksle. Kaip ir pirmais dviem atvejais, taikome prisitaikančiųjų baigtinių strategijų algoritmą pagal tolygų (3.13 pav., a) ir pagal pagerintą beveik optimalų tinklą (3.13 pav., b) visiems keturiems optimalumo kriterijams A, B, C ir D. Gautus skaičiavimo rezultatus surašome į 3.3 lentelę. Galutiniai tinklai, gauti pagal tolygų tinklą pavaizduoti 3.14, 3.15, 3.16 ir 3.17 paveikslų a variantuose, o galutiniai tinklai, gauti pagal pradinį beveik optimalų tinklą – 3.14, 3.15, 3.16 ir 3.17, b.



3.12 pav. L-formos plokštės uždavinys



3.13 pav. Pradiniai tinklai: (a) – tolygus tinklas, (b) – beveik optimalus pradinis tinklas

Lygindami gautus rezultatus, matome, kad skaičiuojant pagal optimalumo kriterijų A, galutinis tinklas, gautas pagal tolygų pradinį tinklą, turi 3858 elementus, o galutinis tinklas, gautas pagal pradinį beveik optimalų tinklą, turi 3682 elementų (176 elementais arba 4,56 % mažiau), be to, abiem atvejais galutiniam optimaliam tinklui rasti prireikė trijų iteracijų. Šiuo atveju, skirtingai nei sprendžiant dviejų medžiagų ir plokštės su skylė uždavinius, pradinio optimalaus tinklo naudojimas didesnio pagerėjimo nedavė.

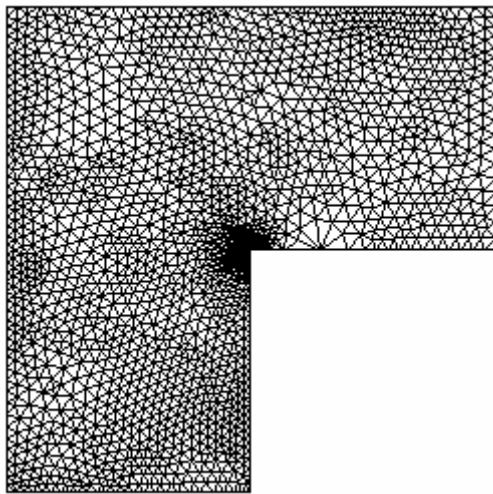
Skaičiuojant pagal kriterijų B, galutinis tinklas, gautas pagal tolygų tinklą turi 6667 elementus, o pagal pradinį beveik optimalų – 5729 elementus. Nors elementų skirtumas galutiniuose optimaliuose tinkluose skaičiuojant pagal tolygaus paklaidos pasiskirstymo ploto vienetui optimalumo kriterijų yra didesnis (938 elementais arba 14,07 %), tačiau galutiniams tinklams gauti vėl prireikė trijų iteracijų. Vadinasi, globaliųjų optimalumo kriterijų taikymas sprendžiant L-formos uždavinį, priešingai, nei sprendžiant dviejų medžiagų bei plokštės su skylė uždavinius, nei iteracijų, nei elementų skaičiaus nesumažina.

Skaičiuojant pagal lokaliuosius kriterijus, situacija daug geresnė – pagal kriterijų C galutinis optimalus tinklas tolygaus pradinio tinklo atveju turi 9686 elementus, o pradinio optimalaus tinklo atveju – 5982 elementus (3704 elementais arba 38,24 % mažiau), be to, pradinio beveik optimalaus tinklo atveju galutiniam optimaliam tinklui rasti reikėjo viena iteracija mažiau. Labai panašius rezultatus gauname optimalumo kriterijui D – galutinis tinklas, gautas pagal tolygų pradinį tinklą turi 9565 elementus, o galutinis tinklas, gautas pagal pradinį beveik optimalų tinklą – 5651 (3914 elementais arba 41 % mažiau). Iteracijų skaičius esant pradiniam beveik optimaliam tinklui taip pat mažesnis, nei tolygaus pradinio tinklo atveju.

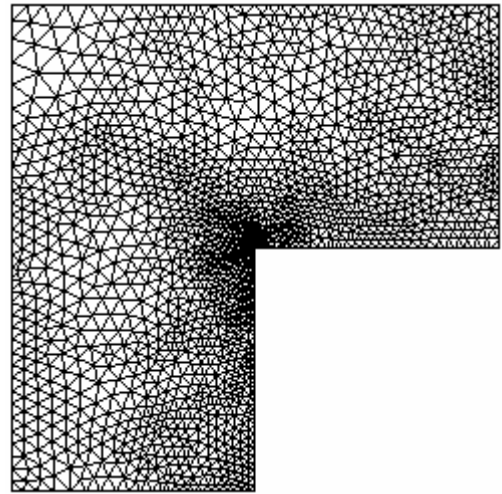
Apibendrinant gautus rezultatus, galima konstatuoti, kad globaliųjų optimalumo kriterijų taikymas sprendžiant L-formos uždavinį norimo rezultato neduoda, t. y. priešingai, negu sprendžiant dviejų medžiagų bei plokštės su skylė uždavinius, nei iteracijų, nei elementų skaičiaus parenkant pradinį beveik optimalų tinklą nesumažėja. Taip gauname todėl, kad L-formos plokštė turi tipinį singuliarųjį tašką, darantį didelę įtaką galutiniam rezultatui. Norint pagerinti prisitaikančiąsias baigtinių elementų strategijas singuliariojo taško aplinkoje, reikalingos papildomos procedūros, kurios kaip tik ir bus nagrinėjamos kitame skyriuje.

3.3 lentelė. L-formos plokštės skaičiavimo rezultatai, leidžiama paklaida $h = 5\%$

		1-oji iteracija		2-oji iteracija		3-oji iteracija		4-oji iteracija	
		$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
A	Tolygus pradinis tinklas	26,35	206	5,81	2831	4,74	3858	-	-
	Pradinis beveik optimalus tinklas	19,39	206	5,15	3267	4,88	3682	-	-
B	Tolygus pradinis tinklas	26,35	206	8,16	2841	4,68	6667	-	-
	Pradinis beveik optimalus tinklas	19,39	206	5,06	4646	4,76	5729	-	-
C	Tolygus pradinis tinklas	26,35	206	8,50	2493	5,64	3806	4,21	9686
	Pradinis beveik optimalus tinklas	19,39	206	5,98	3595	4,74	5982	-	-
D	Tolygus pradinis tinklas	26,35	206	5,26	4390	4,17	9565	-	-
	Pradinis beveik optimalus tinklas	19,39	206	4,50	5651	-	-	-	-

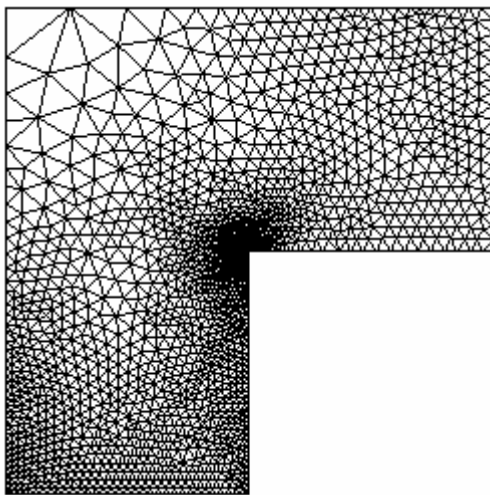


(a)

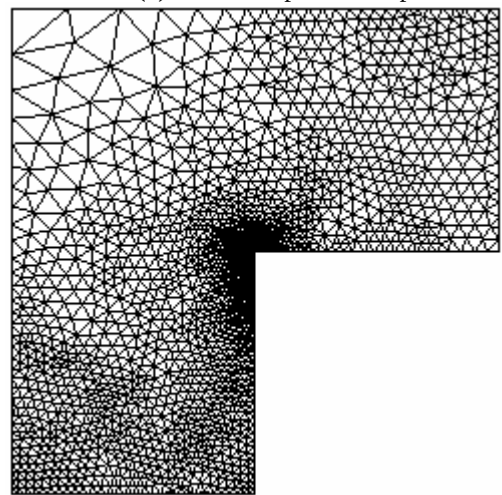


(b)

3.14 pav. Galutiniai tinklai, gauti pagal kriterijų A: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui

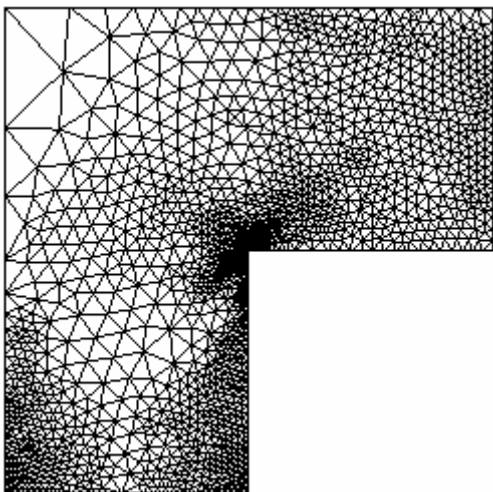


(a)

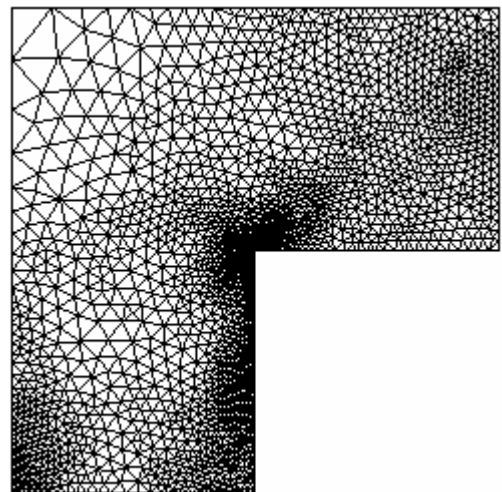


(b)

3.15 pav. Galutiniai tinklai, gauti pagal kriterijų B: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui

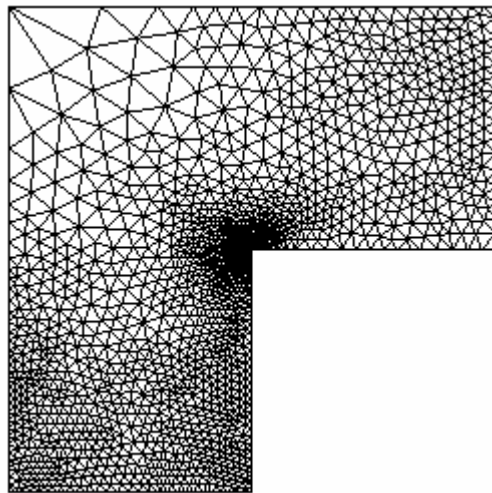


(a)

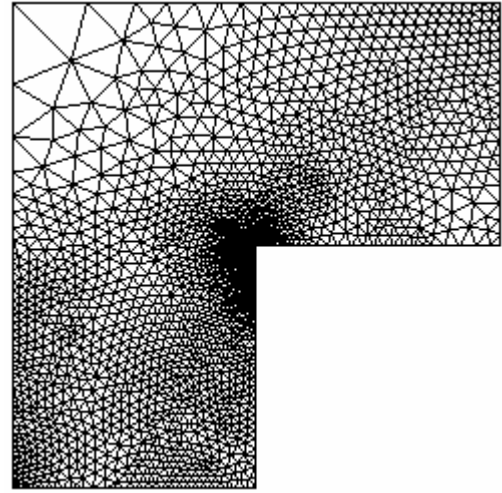


(b)

3.16 pav. Galutiniai tinklai, gauti pagal kriterijų C: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui



(a)



(b)

3.17 pav. Galutiniai tinklai, gauti pagal kriterijų D: (a) – tolygiam pradiniam tinklui, (b) – beveik optimaliam pradiniam tinklui

3.5 Išvados

Šiame skyriuje išnagrinėta globaliųjų ir lokaliųjų optimalumo kriterijų įtaka galutinių optimalių tinklų radimui taikant prisitaikančias baigtinių elementų strategijas. Siekiant labiau optimizuoti prisitaikančias baigtinių elementų strategijas, skaičiavimai atlikti ne tik su keturiais optimalumo kriterijais, bet ir su dviem skirtingais pradiniais tinklais: pradiniu tolygiu ir anksčiau pasiūlytu pradiniu beveik optimaliu. Apibendrinant nagrinėtų mechaninių konstrukcijų skaitinius rezultatus, galima teigti, kad pradinio tinklo bei optimalumo kriterijaus parinkimas smarkiai įtakoja elementų ir iteracijų, reikalingų galutiniam tinklui gauti, skaičių, elementų dydį bei jų pasiskirstymą galutiniame tinkle:

- mažiausią elementų skaičių galutiniame optimaliame tinkle gauname taikydami tolygaus paklaidos pasiskirstymo optimalumo kriterijų (A);
- galutiniuose optimaliuose tinkluose, gautuose pagal tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui (B), Mizeso įtempimų (C) ir maksimalaus įtempimų paklaidos procentinio santykio (D) optimalumo kriterijus, elementų skaičius gali 5 kartus didesnis, nei tinkluose, gautuose pagal tolygaus paklaidos pasiskirstymo optimalumo kriterijų (A);
- elementų dydžio ir tankio kitimas galutiniuose optimaliuose tinkluose, gautuose pagal tolygaus paklaidos pasiskirstymo optimalumo kriterijų (A), yra ne toks žymus, kaip galutiniuose optimaliuose tinkluose, gautuose taikant kitus tris optimalumo kriterijus, pagal kuriuos didesnis kiekis smulkesnių elementų koncentruojami tose konstrukcijos vietose, kur tikimasi gauti didžiausius įtempimų gradientus;
- pradinio beveik optimalaus tinklo parinkimas iteracijų ir elementų skaičių sumažinti leido tik tuose uždaviniuose, kurie neturėjo tipinių singuliarių taškų;
- uždaviniuose, turinčiuose tipinių singuliarių taškų, pradinio optimalaus tinklo parinkimas pirmai iteracijai sumažėjusį iteracijų ir elementų skaičių (~40 %) galutiniame optimaliame tinkle duoda tik

taikant lokalius optimalumo kriterijus, kai tuo tarpu taikant globalius optimalumo kriterijus iteracijų skaičius liko toks pat, o elementų skaičius sumažėjo tik 4–14 %.

Vadinasi, naudojant pagerintą beveik optimalų pradinį BE tinklą vietoj įprastinio tolygaus, galime sumažinti adaptyvaus proceso iteracijų skaičių, o parinkdami skirtingus optimalumo kriterijus, galime kontroliuoti elementų skaičių bei pasiskirstymą galutiniame tinkle, tačiau siekiant pagerinti prisitaikančiąsias baigtinių elementų strategijas singuliaraus taško aplinkoje, reikalingos papildomos procedūros, kurios kaip tik ir bus nagrinėjamos kitame skyriuje.

4. SINGULIARIŲ TAŠKŲ ĮTAKOS VERTINIMAS H-PRISITAIKANČIŲJŲ BAIGTINIŲ ELEMENTŲ ANALIZĖJE SPRENDŽIANT TAMPRUMO TEORIJOS UŽDAVINIUS

Singuliarių taškų įtaka dažniausiai nagrinėjama sprendžiant medžiagų irimo mechanikos uždavinius, kuriuose reikia tiksliai nustatyti įtempimų ir deformacijų būvį tam tikro plyšio viršūnės aplinkoje, tačiau šio darbo apimtis apsiriboja geometrinio singularumo nagrinėjimu sprendžiant sudėtingos geometrijos uždavinius.

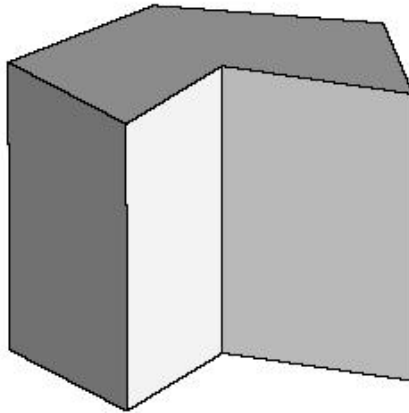
Visi 1.1.4 skyriuje paminėti metodai gana efektyviai ir tiksliai skaičiuoja sprendinį singuliaraus taško aplinkoje, tačiau jų taikymas h -prisitaikančiųjų baigtinių elementų strategijų analizei reikalauja vienokių ar kitokių pakeičimų. Pavyzdžiui, Aalto pasiūlytų singuliariųjų juostų konstravimas yra nepatogus tuo, kad vietoj standartinių formos funkcijų reikėtų imti singuliarus, o toks pakeitimas būtų sunkiai suderinamas su standartinėmis baigtinių elementų programomis. Tokiu atveju vartotojas turėtų pats pasirašyti specialią programą baigtinių elementų analizei atlikti. Szabo ir Yosibash (1996) pasiūlytas metodas tinka p -baigtinių elementų analizei, bet netinka h -baigtinių elementų analizei. Tinkamiausias h -baigtinių elementų analizei būtų Apel pasiūlytas lokalo tinklo gradavimo singuliarinio taško aplinkoje metodas, tačiau jis veikia tik su parametriniais tinklais, kurie nėra tinkami prisitaikančiosioms baigtinių elementų strategijoms.

Dėl šių priežasčių šiame skyriuje sprendiniui singuliaraus taško aplinkoje skaičiuoti pateikiamas naujas h -prisitaikančioms strategijoms tinkantis algoritmas. Pasiūlyto metodo esmę sudaro prielaida, kad singuliarinio taško aplinkoje bendrą sprendinį galime išskaidyti į reguliariąją ir singuliariąją dalis. Taip ieškodami tikslesnio sprendinio be reguliariosios papildomai rasime ir singuliariąją sprendinio dalį. Singuliariosios sprendinio dalies radimas realizuotas modifikuojant mažiausiųjų kvadratų metodą: fizinių koordinačių matricos elementus pakeičiame singuliariaisiais monomais. Pasiūlyto sprendinio gerinimo metodo efektyvumą iliustruosime atlikdami L-formos plokštės uždavinio analizę. Norint pagerinti prisitaikančiąsias baigtinių elementų strategijas, vien pagerinti sprendinį neužtenka, todėl pasiūlytą sprendinio gerinimo singuliarinio taško aplinkoje metodiką įkomponuosime į prisitaikančiųjų strategijų algoritmą. Galiausiai į vieną algoritmą sujungsime ir kitus anksčiau pasiūlytus prisitaikančiųjų strategijų gerinimo būdus: pradinio beveik optimalaus tinklo generavimo algoritmą bei optimalumo kriterijų taikymą generuojant naują elementų tinklą.

Šio skyriaus tyrimų medžiaga 2005 metais pristatyta „Mechatronic Systems and Materials – 2005“ konferencijoje ir publikuota „Solid State Phenomena“ žurnale (Vasiliauskienė, Baušys, 2006).

4.1 Bendra sprendinio išraiška singuliariose srities zonose

Šiame skyrelyje aprašysime matematinį uždavinį, kuriuo remsimės ieškodami pagerinto sprendinio singuliarinio taško aplinkoje. Tarkime, kad Ω yra prizminis trimatis objektas (4.1 pav.), aprašomas formule:



4.1 pav. Prizminė trimatė sritis su singuliaria briauna

$$\Omega = G \times Z, \quad (4.1)$$

čia $G \subset R^2$ – plokštuminis daugiakampis, o $Z = (0, z_0) \subset R^1$ – atkarpa. Daugiakampis G privalo turėti bent vieną viršūnę C , kurios vidinis kampas $w > p$. Visi kiti daugiakampio kampai turi būti mažesni, nei p (4.1 pav.). Viršūnę C vadinsime singuliariuoju srities Ω tašku. Apibrėžiame vadinamąją singuliariąją briauną, einančią per tašką C :

$$E = C \times Z. \quad (4.2)$$

Žinoma (Kufner, Sändig, 1987), kad briaunos E aplinkoje sprendinys u gali būti išskaidomas į reguliariąją u_r ir singuliariąją u_s dalis:

$$u = u_s + u_r, \quad (4.3)$$

čia

$$u_s = x(r)g(x)r^l\Phi(f), \quad (4.4)$$

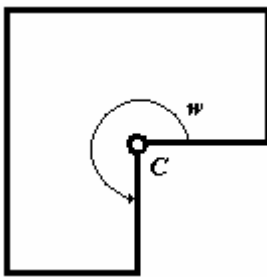
$$l = \frac{p}{w},$$

$r = \text{dist}(x, E)$, $f = (0, w)$ – polinės koordinatės plokštumoje, statmenoje briaunai E ,

$$x(r) = \begin{cases} 1, & \text{kai } r < R_0, \\ 0, & \text{kai } r > 2R_0, \end{cases} \quad (4.5)$$

$g(x)$ – koeficientų funkcija,

$$\Phi(f) = \sin lf.$$



4.2 pav. Plokštuminis daugiakampis su singuliaruoju tašku C

Koeficientų funkcija $g(x)$, dar kartais vadinama *įtempimų intensyvumo pasiskirstymu*, priklauso nuo visų trijų erdvės koordinačių ir išreiškiama konvoliucijos integralu:

$$g(r, x_3) = \frac{1}{p} \int_R \frac{r}{r^2 + s^2} q(x_3 - s) ds. \quad (4.6)$$

Tuo atveju glodumo koeficientas q išreiškiamas per l Besovo erdvėje (Grisvard, 1995).

Plokštumos atveju, koeficientų funkcija $g(x)$ yra konstanta. Be to, tam tikrais atvejais (pvz., esant sudėtingoms kraštinėms sąlygoms) singuliarioji dalis u_s gali būti užrašoma kaip singuliariųjų funkcijų suma:

$$u_s = x(r) \sum_i g_i(x) r^{l_i} F_i(f). \quad (4.7)$$

Taip pat egzistuoja tam tikri atvejai, kai priklausomai nuo kampo ω , į šią išraišką reikia įtraukti ir logaritminius narius. Apibendrinant galima teigti, kad kuo sudėtingesnis uždavinys, tuo sunkesnis singuliariosios dalies u_s radimas. Tokių sudėtingesnių atvejų apžvalgą galima rasti šiose monografijose (Dauge, 1988; Grisvard, 1985).

Kadangi šio darbo tikslams įgyvendinti nagrinėjami tik plokštuminiai uždaviniai, tai anksčiau aprašytas uždavinys supaprastėja. Nagrinėkime (4.1) lygtimi aprašyto trimačio prizminio kūno plokštuminį variantą (4.2 pav.):

$$\Omega = G, \quad (4.8)$$

čia $G \subset \mathbb{R}^2$ – plokštuminis daugiakampis, turintis viršūnę C su vidiniu kampu $w > p$. Visi kiti daugiakampio kampai turi būti mažesni nei p . Viršūnę C vadinsime singuliaruoju daugiakampio tašku. **Pastaba:** nagrinėjamoji sritis parenkama taip, kad joje būtų tik vienas singularusis taškas, priešingu atveju sritis būtų skaidoma į dalis tol, kol kiekvienoje dalyje liktų tik po vieną singuliarųjį tašką.

Analogiškai anksčiau aprašytam atvejui, sprendinys singuliarinio taško aplinkoje yra skaidomas į reguliariąją u_r ir singuliariąją u_s dalis:

$$u = u_s + u_r = x(r)g(x)r^l \sin l f + u_r, \quad (4.9)$$

$$l = \frac{p}{w},$$

$r = \text{dist}(0, x)$, $f \in (0, w)$ – polinės koordinatės nagrinėjamoje plokštumoje,

$$g = \text{const} \in R,$$

$$x(r) = \begin{cases} 1, & \text{kai } r < R_0 \\ 0, & \text{kai } r \geq R_0. \end{cases}$$

4.2 Popprocesorinis paklaidos vertinimas įtraukiant singuliariojo taško įtaką

4.2.1 Superkonvergencinio elementų fragmento atkūrimo metodo panaudojimas singuliaraus sprendinio radimui

Sprendiniui gerinti singuliariojo taško aplinkoje bus reikalingas superkonvergencinis elementų fragmento atkūrimo metodas, kurio esmę sudaro naujo tiesioginių baigtinių elementų kintamųjų lauko radimas. Surastas naujasis kintamųjų laukas yra aukštesnės tikslumo klasės. Jis apskaičiuojamas mažiausiųjų kvadratų metodu, naudojant elementų fragmentus. Taikant mažiausiųjų kvadratų metodą naudojami popprocesorinio sprendinio superkonvergencinių taškų rezultatai. Superkonvergenciniai taškai – tai specialiai parinkti diskretizavimo srities taškai, kuriuose baigtinių elementų sprendinys yra aukštesnės tikslumo klasės, negu apskaičiuotasis pagal apriorinius paklaidų įvertinimus.

Naujasis tiesioginių kintamųjų laukas $\mathbf{u}_i^*$ elemente $t \in T_h$ išreiškiamas $p+1$ eilės baigtinių elementų formos funkcijomis:

$$\mathbf{u}_i^*(\mathbf{x}) = \sum_r N_r^*(\mathbf{x})(\mathbf{u}_r^*)_i + \sum_s N_s^*(\mathbf{x})(\mathbf{u}_s^*)_i. \quad (4.10)$$

čia r žymi originalius baigtinių elementų mazgus, o s – papildomus elemento mazgus, kuriuose apskaičiuojamos atkurto sprendinio reikšmės, $N_r^*(\mathbf{x})$ ir $N_s^*(\mathbf{x})$ – lokalsios $p+1$ eilės formos funkcijos, atitinkančios originalius ir papildomus elemento mazgus.

Originalių elemento mazgų reikšmės yra lygios popprocesorinio sprendinio mazgų reikšmėms $(\mathbf{u}_r^*)_i \equiv (\mathbf{u}_r^h)_i$ ir atkurto kintamųjų lauko reikšmei $(\mathbf{u}_s^*)_i$, apskaičiuotai papildomuose mazguose. Jos randamos sprendžiant mažiausių kvadratų uždavinį atskirame elementų fragmente Ω_τ :

$$\text{rasti tokį } \mathbf{u}_i^* \in R_{p+1}, \text{ kad} \quad (4.11)$$

$$J_{W_t}(\mathbf{u}_i^*) = \min_{(\mathbf{u}^{f*})_i \in R_{p+1}} J_{W_t}(\mathbf{u}^{f*})_i,$$

čia

$$J_{w_t}(\mathbf{u}^{f*})_i = \sum_{j=1}^{ns} w_j^2 \mathbf{R}_u^T(\mathbf{x}_j) \mathbf{R}_u(\mathbf{x}_j), \quad (4.12)$$

čia $\mathbf{R}_u(\mathbf{x}_j)$ – liekamasis narys, apskaičiuotas superkonvergenciniuose taškuose, $w(\mathbf{x})$ – teigiama glodžioji svorio funkcija, įgyjanti reikšmę 1 nagrinėjamo elemento ribose ir reikšmę, mažesnę nei 1 – už nagrinėjamo elemento ribų.

Liekamojo nario išraiška:

$$\mathbf{R}_u = (\mathbf{u}^{f*})_i - (\mathbf{u}_r^h)_i. \quad (4.13)$$

O $(\mathbf{u}^{f*})_i$ aprašomas $p+1$ eilės daugianariu

$$(\mathbf{u}^{f*})_i = \mathbf{Q}(\mathbf{x}) \mathbf{b}, \quad (4.14)$$

čia

$$\mathbf{Q} = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 \end{bmatrix} \quad (4.15)$$

yra fizinių koordinačių $p+1$ eilės monomų matrica, o $\mathbf{b}$ – ieškomų dydžių vektorius.

Taigi, suformulavus mažiausiųjų kvadratų uždavinį, reikia spręsti algebrinę lygčių sistemą:

$$\left(\sum_j w_j^2 (\mathbf{Q}(\mathbf{x}_j))^T \mathbf{Q}(\mathbf{x}_j) \right) \mathbf{b}_i = \left(\sum_j w_j^2 (\mathbf{Q}(\mathbf{x}_j))^T \right) (\mathbf{u}_j^h)_i. \quad (4.16)$$

Norėdami taikyti (4.9) formules geresniam sprendiniui singuliarinio taško aplinkoje rasti, visų pirma turėsime nuspręsti, kaip skaičiuosime reguliariąją ir singuliariąją sprendinio dalis. Kadangi bendruoju atveju analitinė sprendinio išraiška nėra žinoma, vietoj reguliariosios dalies imsime baigtinių elementų sprendinį:

$$\mathbf{u}_r = \mathbf{u}_{FEM}. \quad (4.17)$$

Singuliarajai sprendinio $\mathbf{u}$ daliai rasti modifikuosime 4.2.1 skyriuje aprašytą superkonvergencinį mažiausiųjų kvadratų metodą. Pagrindinė mažiausiųjų kvadratų metodo pakeitimo idėja – pakeisti fizinių koordinačių vektorius į tokį vektorius, kuris turėtų singuliariųjų monomų (elementų). Tai reiškia, kad artimoje singuliarinio taško aplinkoje vietoj standartinio kvadratinio daugianario imame vektorius, susidedantį iš vieno singuliarinio nario:

$$\mathbf{Q}_{SING}(r, f) = [x(r) r^l \sin l f]. \quad (4.18)$$

Išsprendę (4.18) lygčių sistemą su (4.18) singuliarinio daugianariu, galėtume rasti singuliariąją sprendinio $\mathbf{u}$ dalį $\mathbf{u}_s$:

$$u_s = Q_{SING} \cdot b_{SING}. \quad (4.19)$$

Taip pat nesunkiai rastume bendrą sprendinio u reikšmę singuliaraus taško aplinkoje:

$$u = u_s + u_{FEM}. \quad (4.20)$$

Tokiu būdu parenkant daugianarius, sistemos (4.18) sprendimas apimtų du atvejus: kai taškas x_j patenka į singuliariąją zoną, patikslintiems poslinkiams skaičiuoti naudojamas singularusis daugianaris, o kai taškas x_j išeina iš singuliariosios zonos ribų, poslinkiams nustatyti bus naudojamas standartinis kvadratinis daugianaris:

$$Q = \begin{cases} Q(x), & \text{kai } \text{dist}(C, N_j) \geq R_0 \\ Q_{SING}(r, j), & \text{kai } \text{dist}(C, N_j) < R_0 \end{cases} \quad (4.21)$$

4.2.2 Praktinio algoritmo konstravimas sprendinio suradimui singuliaraus taško aplinkoje

Taigi turime plokštuminę prizminę sritį Ω , kurios vidinis kampas $w > p$, C – singularusis srities Ω taškas. Spindulys R_0 apibrėžia sritį, kurioje singuliarioji sprendinio u dalis $u_s \neq 0$. Tarkime, sritį Ω dengia N trikampių elementų tinklas:

$$T = \{T_i | i = 1, \dots, N\}. \quad (4.22)$$

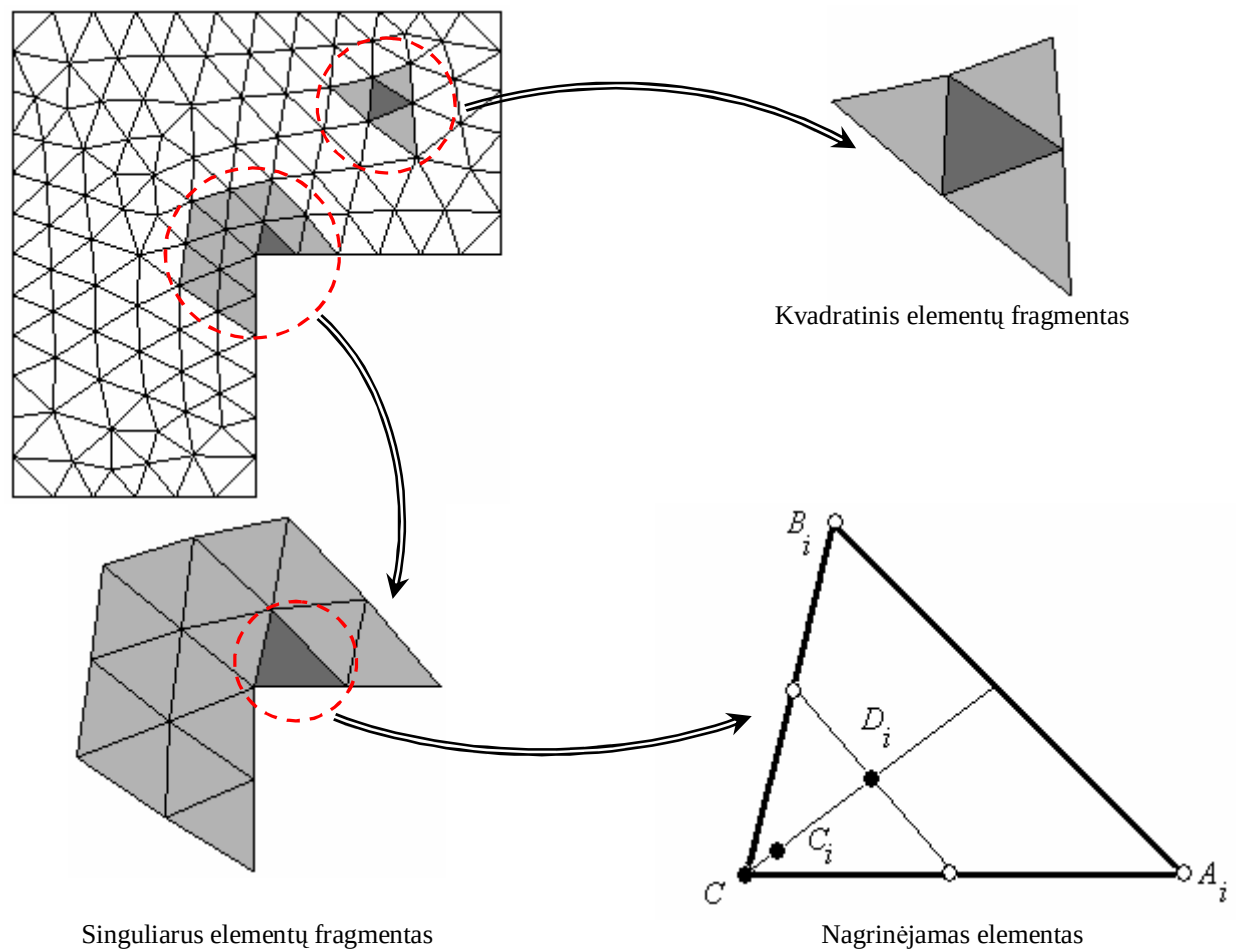
Tarsime, kad trikampis T_i yra singularusis, jei bent vienas jo mazgas nuo taško C yra nutolęs atstumu, mažesniu už R_0 . Aibė tokių singuliarių elementų sudaro singuliarųjį elementų fragmentą, naudojamą atliekant tolesnius sprendinio skaičiavimus (4.3 pav.):

$$W_{SING} = \{T_i | \|Node_j(T_i)\| \leq R_0, j = 1, 2, 3\}. \quad (4.23)$$

Kadangi singuliarijame taške $r = 0$ ir $f \in [0, 2p]$ daugianaris (4.18) negali būti apibrėžtas korektiškai, tai visiems singuliariojo elementų fragmento elementams, besiribojantiems su singuliariuoju tašku C

$$T_C = \{T_{Ci} | \{T_i \cap C\} \neq \emptyset, i = 1, \dots, M\} \quad (4.24)$$

surasime papildomus taškus $C'_i, i = 1, \dots, M$, esančius labai arti originalaus taško C :



4.3 pav. Singuliariojo elemento fragmento formavimas, naujo singuliariojo taško parinkimas

$$C_i = (1-t) \cdot C + t \cdot D_i, \quad (4.25)$$

čia

$$D_i = 0,5 \cdot (0,5 \cdot A_i + 0,5 \cdot C) + 0,5 \cdot (0,5 \cdot B_i + 0,5 \cdot C),$$

$$t = \frac{h}{\sqrt{N}} \cdot \frac{const}{\|CD_i\|},$$

čia A_i ir B_i – singuliariojo elemento T_{Ci} mazgai; D_i – papildomas taškas, esantis ant taško C pusiaukampinės; N – tinklo elementų skaičius; h – mažiausio elemento, priklausančio singuliariajam fragmentui, dydis; $\|CD_i\|$ – atstumas nuo C iki D_i ; $const \in (0,1]$ – glodumo konstanta. Toks parametro t apibrėžimas leidžia kontroliuoti papildomą tašką C_i padėti: santykis $\frac{\sqrt{h}}{N}$ užtikrina, kad C_i tikrai įeis į singuliarųjį elementą T_{Ci} , o dalyba iš $\|CD_i\|$ užtikrina, kad C_i tikrai bus labai arti originalaus taško C .

Atstumas nuo singuliaraus taško C iki papildomai sugeneruotų taškų C_i yra vadinamas *kampo aštrumu*, parodančiu kokioje singuliaraus taško aplinkoje dar gaunami stabilūs rezultatai. Praktinių skaičiavimų metu nustatyta, kad apibrėžus parametą t pagal aukščiau nurodytas formules, atstumas nuo singuliaraus taško C iki papildomai gautų taškų C_i priklausomai nuo tinklo elementų dydžio ir skaičiaus gali kisti nuo 0,5 mm iki 0,01 mm.

Kitų autorių, nagrinėjusių uždavinius su plyšiais ir skylėmis, darbuose parenkami panašūs kampo aštrumai. Pavyzdžiui, Kačianausko su bendraautoriais (Kačianauskas, Stupak, Kienzler, 2002) nagrinėtuose uždaviniuose parinko 0,078 ir 0,448 mm kampo aštrumą, Stupak (Stupak, Kačianauskas, 2000; Stupak, 2004) įvardija 0,07, 0,018 bei 0,09 mm kampo aštrumą, o Žarnovskij (Žarnovskij, Kačianauskas, 2003) taiko 0,001 mm kampo aštrumą.

Visiems singuliariojo fragmento Ω_{SING} mazgams suformuojame fizinių koordinačių vektorius $x_{SING}(x_k, y_k)$ ir turėdami omenyje, kad $x_k = r_k \cos j_k$ ir $y_k = r_k \sin j_k$, apibrėžiame singuliarųjį daugianarį $Q_S(r, j) = [x(r) r^l \sin l f]$. Kadangi singuliarusis daugianaris skaičiuojamas polinėse koordinatėse, poslinkių vektorių (u_j^h) taip pat reikia pervesti į polines koordinates: $(u_j^h) \rightarrow (u_{SING,j}^h)$. Singuliarajam elementų fragmentui Ω_{SING} gauname tokią lygčių sistemą:

$$\left(\sum_j w_j^2 (Q_{SING}(x_{SING,j}))^T Q_{SING}(x_{SING,j}) \right) b_{SING,i} = \left(\sum_j w_j^2 (Q_{SING}(x_{SING,j}))^T \right) (u_{SING,j}^h)_i. \quad (4.26)$$

Išsprendę šią lygčių sistemą, randame ieškomus koeficientus b_{SING} ir singuliarią sprendinio dalį u_s .

Pastaba: kadangi u_s buvo rastas polinėse koordinatėse, prieš skaičiuojant toliau būtina atlikti atvirkštinę transformaciją ir pervesti u_s į dekartio koordinates.

Taip apibrėžtam taškui C_j spindulys ir kampas jau nebus lygūs nuliui, singuliarusis daugianaris Q_S bus apibrėžiamas korektiškai, todėl patikslintam sprendiniui rasti jau galėsime taikyti singuliarųjį elementų fragmento atkūrimo metodą. Iš viso randama tiek papildomų taškų, kiek yra elementų, besiribojančių su singuliariuoju tašku C .

Turėdami n papildomų taškų iš labai artimos singuliariojo taško aplinkos bei suradę pagerintus poslinkius juose, galime rasti pagerintą poslinkį pačiame singuliarijame taške ekstrapoliuojant gautus n papildomų poslinkių. Taikydami tą patį mažiausiųjų kvadratų metodą, sprendžiame (4.16) lygčių sistemą, naudodami rastus poslinkius $u_j(C_j)$ papildomuose C_j taškuose:

$$\left(\sum_j w_j^2 (Q(x_j))^T Q(x_j) \right) b_i = \left(\sum_j w_j^2 (Q(x_j))^T \right) (u_j^h(C_j)). \quad (4.27)$$

Suradę ieškomą vektorių b , suskaičiuojame poslinkius singuliariajame taške.

Kadangi visi šie skaičiavimai yra atliekami sričiai su iš anksto žinomu singuliariuoju kampu, todėl atlikus baigtinių elementų analizę ir suradus pagerintą sprendinį naujas tinklo elementų dydis apskaičiuojamas taikant (1.44) išraišką:

$$h'_k = \frac{h_k}{x^{1/l}}. \quad (4.28)$$

Pasiūlytojo metodo veiksmingumui iliustruoti atliksime du skirtingus eksperimentus. Pirmojo eksperimento metu palyginsime tradicinio poprocesorinio sprendinio, apskaičiuoto naudojant Zienkiewicz ir Zhu pasiūlytą algoritmą (Zienkiewicz, Zhu, 1987), bei pagerinto singuliaraus poprocesorinio sprendinio konvergavimą. Antrojo eksperimento metu pasiūlytą metodą įdėsime į prisitaikančiųjų baigtinių elementų strategijos algoritmą ir pažiūrėsime, kokią įtaką pagerintas sprendinys daro iteracijų skaičiui ir galutiniam elementų skaičiui tinkle.

4.3 Tradicinio poprocesorinio sprendinio ir pagerinto poprocesorinio singuliaraus sprendinio palyginimas nagrinėjant L-formos plokštės uždavinį

Nagrinėkime L-formos plokštę, aprašytą 3.4.3 skyriuje. Leistina paklaida $h = 5\%$. Kadangi L-formos plokštės uždaviniui tikslus analitinis sprendinys nėra žinomas, palaipsniui didindami elementų skaičių generuojame tolygius baigtinių elementų tinklus tol, kol santykinė įtempimų paklaida dviems vienas po kito einantiems tinklams bus mažesnė, nei 2 %. Esant 25000 elementų tinklui gauname sukonvergavusį sprendinį, kurio poslinkių reikšmes laikysime tiksliais ir naudosime nagrinėjamų pavyzdžių rezultatams įvertinti. Norėdami įsitikinti, kad pasiūlytas singuliariojo poprocesorinio sprendinio skaičiavimo algoritmas tikrai pagreitina sprendinio konvergavimą lyginant su tradiciniu poprocesoriniu sprendiniu, atliksime tris bandymus.

Pirmajam bandymui atlikti parinksime keturis reguliarius tolygius tinklus, kurių elementų skaičius ir dydis kinta pagal tokį dėsnį:

$$N_i = \begin{cases} 150, & \text{kai } i = 1 \\ 4 * N_{i-1} & \text{kai } 1 < i \leq 4 \end{cases}, \quad (4.29)$$

$$h_i = \begin{cases} 0.1, & \text{kai } i = 1 \\ 0.5 * h_{i-1} & \text{kai } 1 < i \leq 4 \end{cases}. \quad (4.30)$$

Pirmasis tinklas susideda iš 150 $h = 0,1$ dydžio elementų, antrasis tinklas susideda iš 600 dvigubai mažesnių $h = 0,05$ dydžio elementų, trečiasis tinklas susideda iš 2400 $h = 0,025$ dydžio elementų, o

4.1 lentelė. Reguliarių tolygių tinklų su vienoda singuliaria sritimi rezultatai

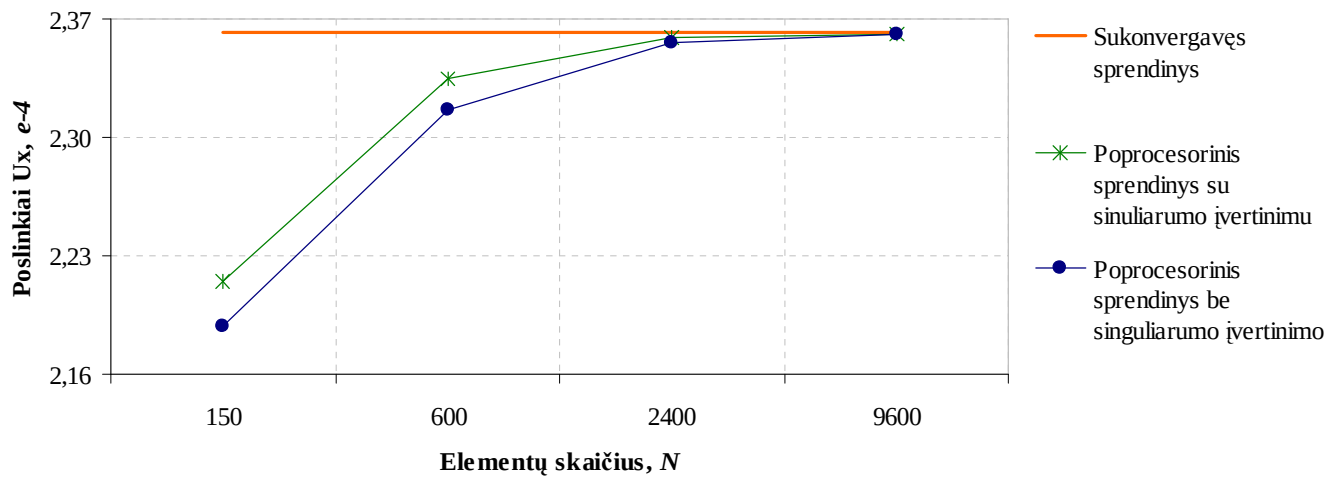
Poslinkių UX reikšmės, e-4				
Elementų skaičius	150	600	2400	9600
Poprocesorinis sprendinys be singuliarumo įvertinimo	2,188283	2,316560	2,355673	2,360946
Poprocesorinis sprendinys su singuliarumo įvertinimu	2,214648	2,334816	2,358444	2,361133
Konvergavęs sprendinys	2,362	2,362	2,362	2,362
Poslinkių UY reikšmės, e-4				
Elementų skaičius	150	600	2400	9600
Poprocesorinis sprendinys be singuliarumo įvertinimo	1,246261	1,410350	1,475205	1,499787
Poprocesorinis sprendinys su singuliarumo įvertinimu	1,294452	1,437034	1,480956	1,500151
Konvergavęs sprendinys	1,511	1,511	1,511	1,511
Sprendinio paklaidos norma, %,				
Elementų skaičius	150	600	2400	9600
Paklaida be singuliarumo įvertinimo	23,301	14,042	7,642	4,474
Paklaida su singuliarumo įvertinimu	24,899	14,974	7,727	4,475
Leidžiama paklaida	5,000	5,000	5,000	5,000

ketvirtasis tinklas – iš 9600 $h = 0,0125$ dydžio elementų. Visais keturiais atvejais spindulio $R = 0,1$ apibrėžta singuliari sritis yra to paties ploto (4.7 pav., 4.8 pav., 4.9 pav. ir 4.10 pav.).

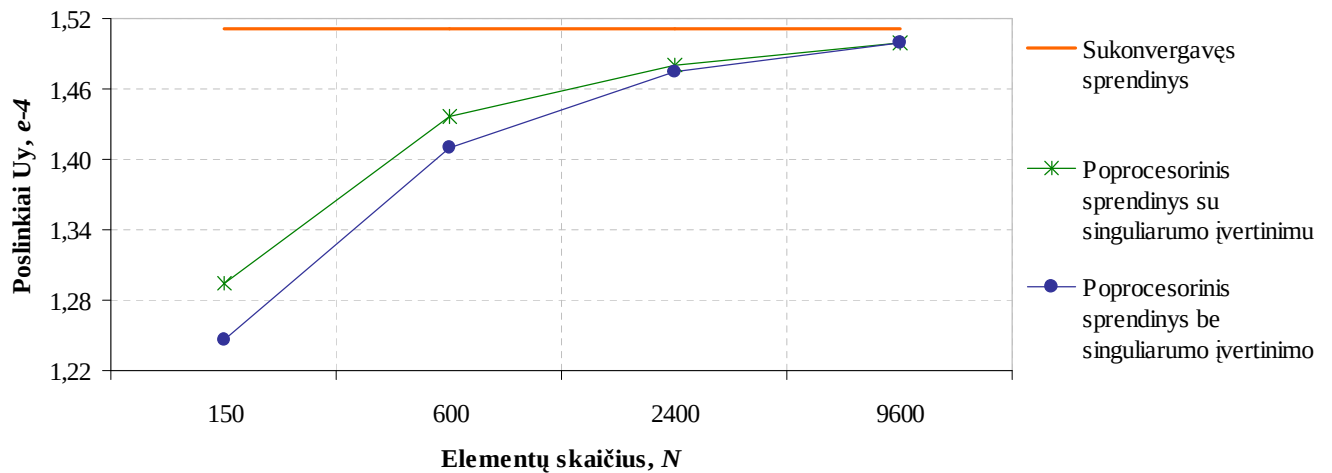
Kiekvienam iš šių keturių tinklų atliekame baigtinių elementų analizę ir randame baigtinių elementų sprendinį. Tada remdamiesi gautu baigtinių elementų sprendiniu, atliekame poprocesorinį paklaidos vertinimą atsižvelgdami į singuliariosios srities įtaką ir neatsižvelgdami į singuliarios srities įtaką. Tokiu būdu gauname du poprocesorinius sprendinius: kai singuliarumas įvertintas ir kai singuliarumas neįvertintas. Abiem atvejais gautos sprendinių bei paklaidos reikšmės surašytos į 4.1 lentelę. Poslinkių U_x , U_y konvergavimo grafikai pateikti 4.4 ir 4.5 paveiksluose, o paklaidų, gautų taikant abu metodus, palyginamieji grafikai pavaizduoti 4.6 paveiksle.

Lygindami gautus grafikus, matome, kad tiek U_x (4.4 pav.), tiek U_y (4.5 pav.) reikšmės, gautos taikant poprocesorinį paklaidų vertinimo metodą kai įvertinamas singuliarumas, kiekvienam nagrinėjamam tinklui yra artimesnės tiksliam sprendiniui, nei sprendinio, gauto pagal tradicinį paklaidų vertinimo metodą. Taip pat lygindami paklaidų grafikus (4.6 pav.), apskaičiuotus sumuojant visos nagrinėjamos srities paklaidas, matome, kad paklaidos reikšmė, apskaičiuota nevertinant singuliarumo yra mažesnė, nei paklaidos reikšmė, kai singuliarumas įvertintas. Pvz., 150 elementų tinklui paklaida, gauta taikant tradicinį poprocesorinį paklaidų vertinimo metodą, yra 23,301 %, o paklaida, gauta taikant singuliarų poprocesorinį paklaidų vertinimo metodą, yra 24,899 % (1,598 % didesnė). Tai reiškia, kad patikslinus sprendinio reikšmę, t. y. labiau ją priartinus prie tikslaus sprendinio, poprocesorinio paklaidos skaičiavimo metu gaunamas didesnis skirtumas tarp baigtinių elementų sprendinio ir pagerinto poprocesorinio singuliaraus sprendinio. Be to, didesnė paklaidos reikšmė leidžia tiksliau generuoti

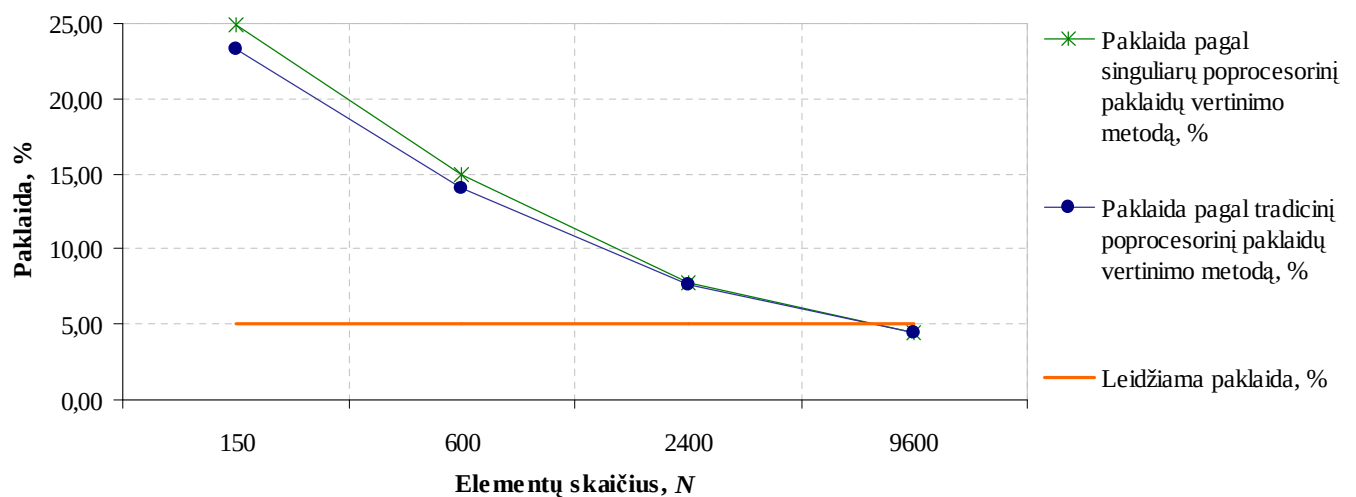
baigtinių elementų tinklą. Režiumuojant galima teigti, kad tolygių reguliarių tinklų atveju pasiūlytas metodas tikrai leidžia pagerinti sprendinio reikšmę singuliariame taške.



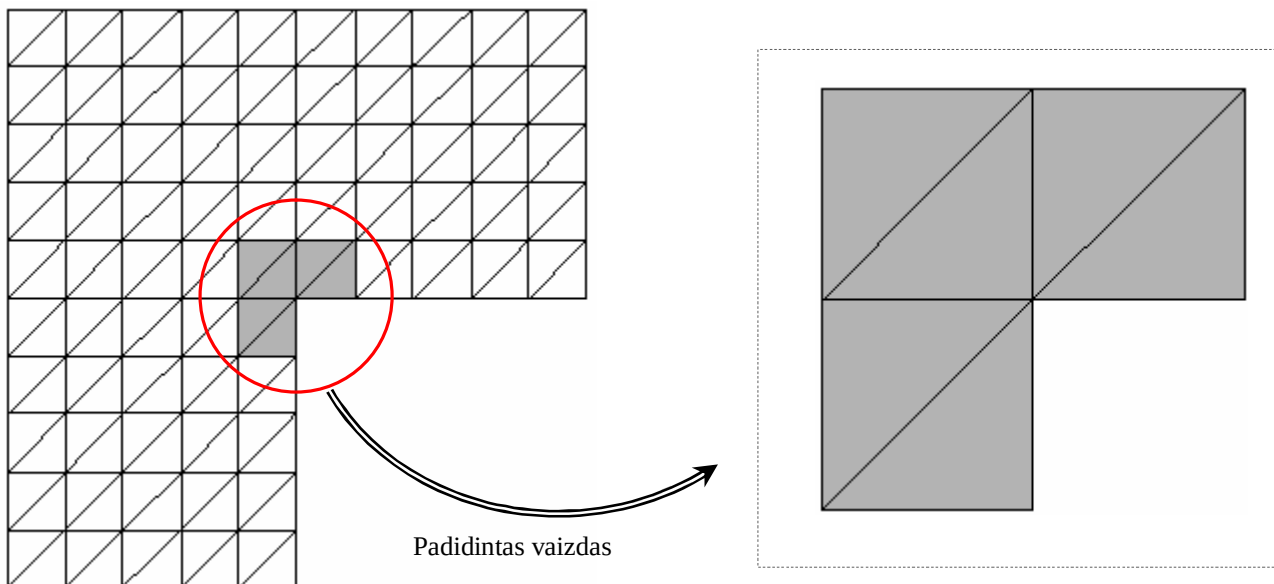
4.4 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_x konvergavimo grafikas



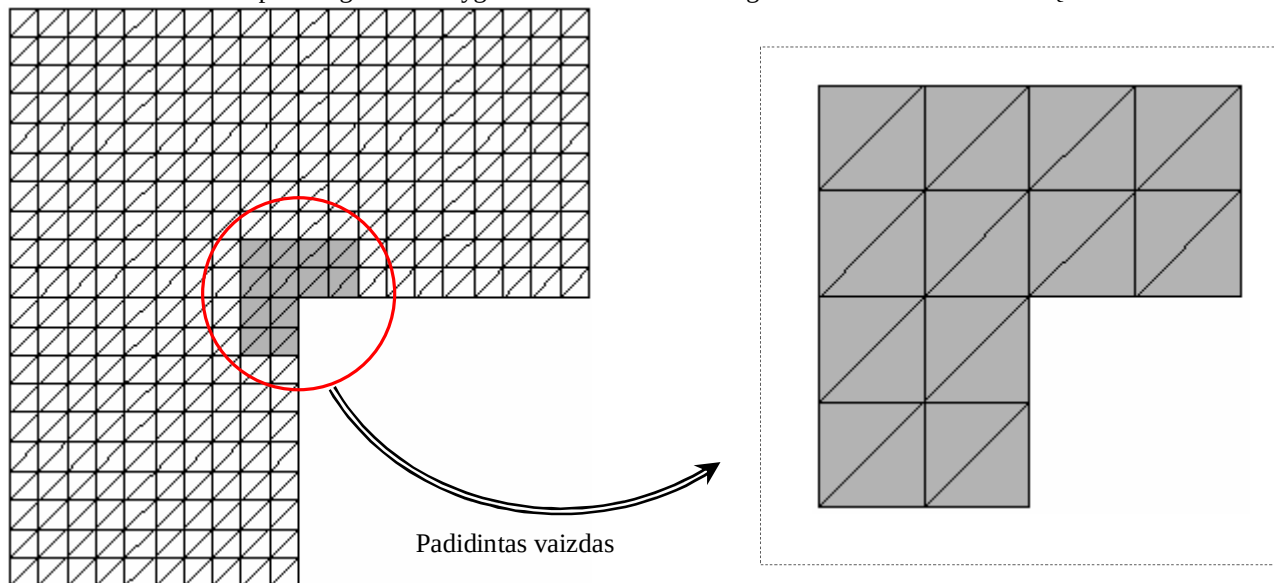
4.5 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_y konvergavimo grafikas



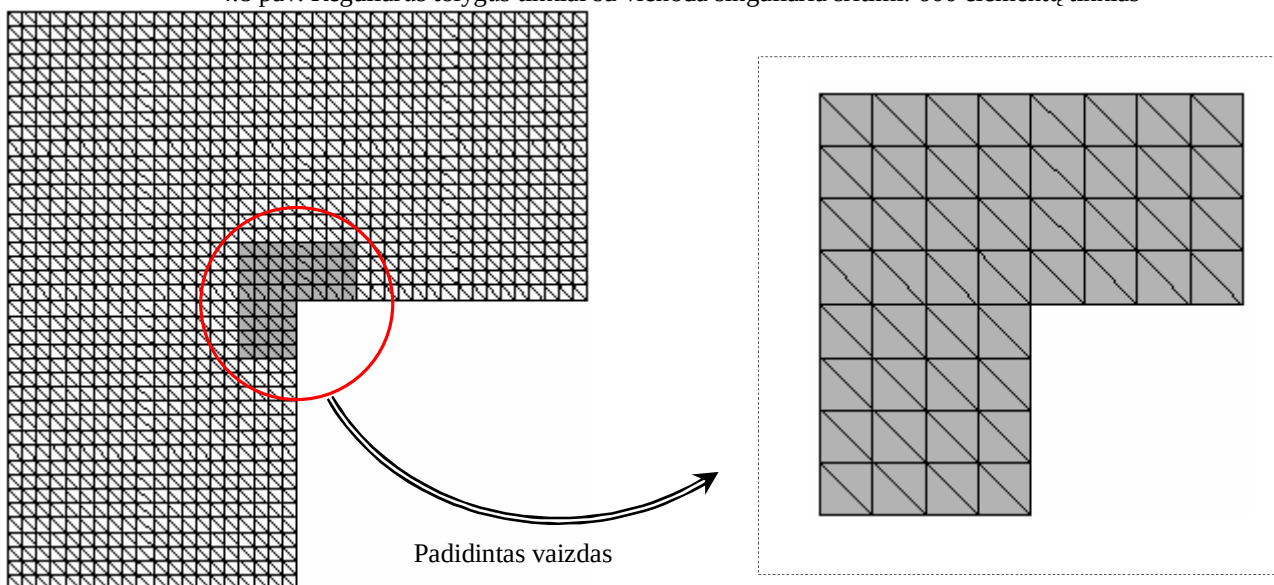
4.6 pav. Regularūs tolygūs tinklai su vienoda singuliaria sritimi: visos nagrinėjamos srities Ω paklaidų konvergavimo palyginimas



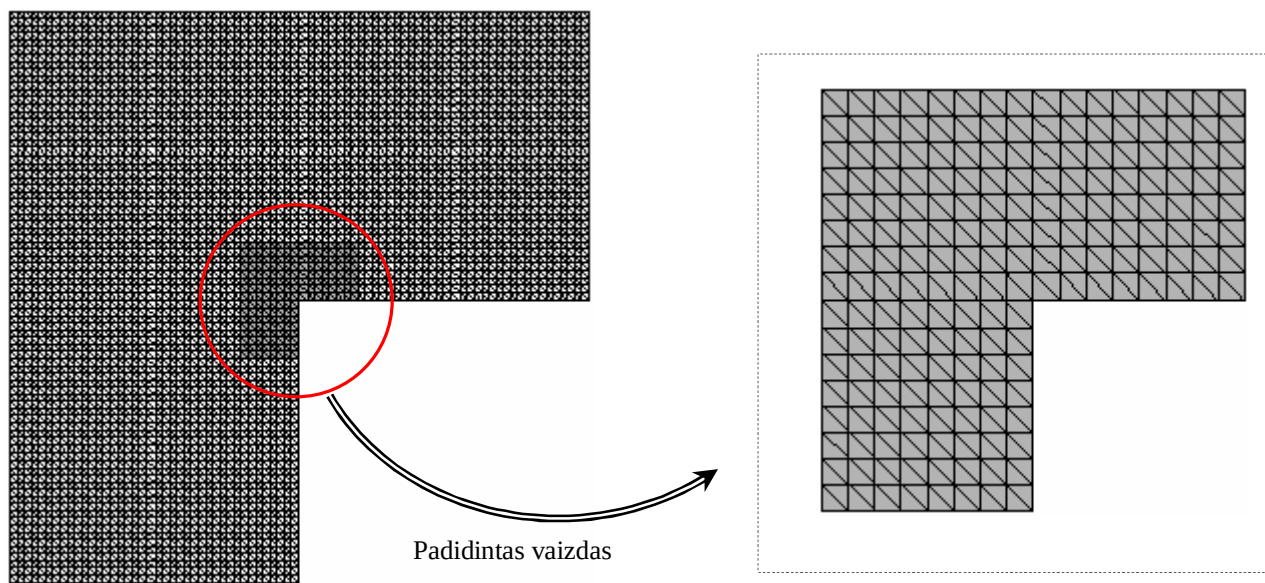
4.7 pav. Reguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 150 elementų tinklas



4.8 pav. Reguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 600 elementų tinklas



4.9 pav. Reguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 2400 elementų tinklas



4.10 pav. Reguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 9600 elementų tinklas

Atlikdami antrąjį eksperimentą parinkime šiek tiek sunkesnį uždavinį. Nagrinėkime tą pačią L-formos plokštę, tačiau vietoj tolygių reguliarių tinklų imkime bendresnį atvejį – tolygius neregulius tinklus, sugeneruotus taip, kad kiekvieno nagrinėjamo tinklo (4.23) formule apibrėžtas singularus elementų fragmentas apimtų tą patį srities plotą (pirmieji keturi šios sekos tinklai pavaizduoti nuo 4.14 pav. iki 4.17 pav.):

$$N_i = \begin{cases} 200, & \text{kai } i = 1 \\ 2 * N_{i-1}, & \text{kai } 1 \leq i \leq 6, \\ 9500, & \text{kai } i = 7 \end{cases} \quad (4.31)$$

$$h_i = \sqrt{\frac{2 * W_L}{N_i}}, \quad (4.32)$$

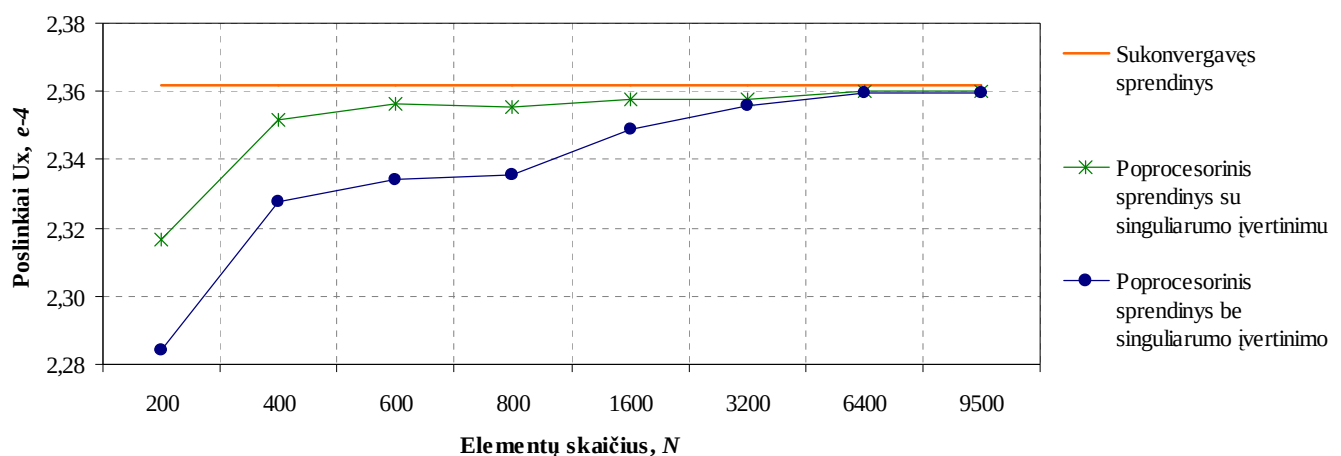
čia $\Omega_L = 0,75$ – plokštės plotas, $i = 1, \dots, 6$ – tinklo eilės numeris

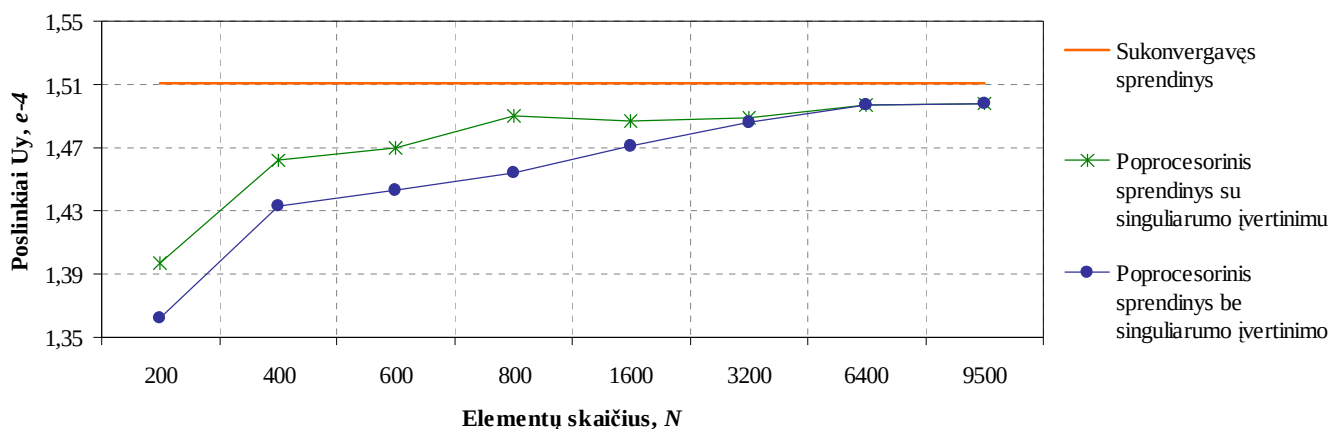
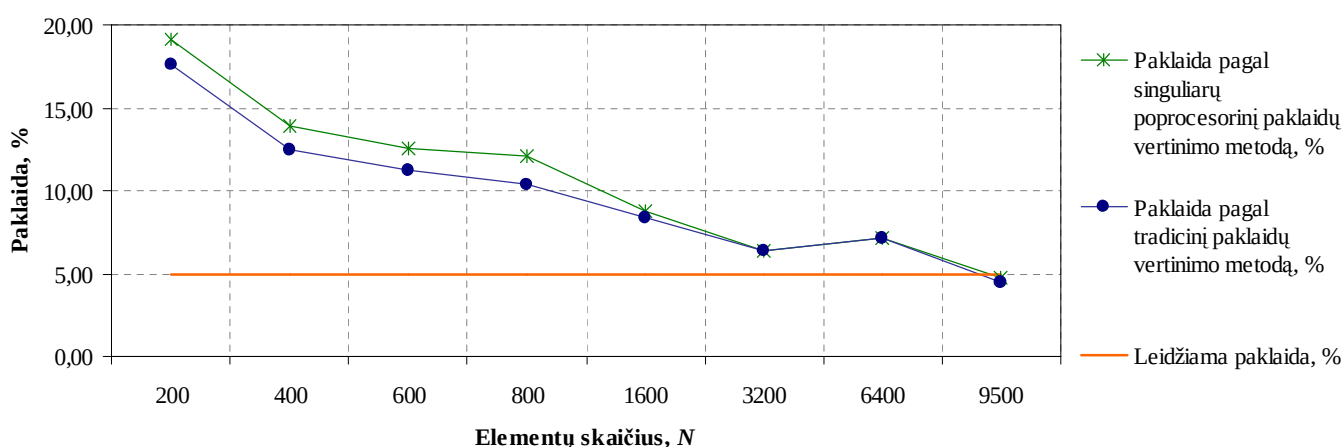
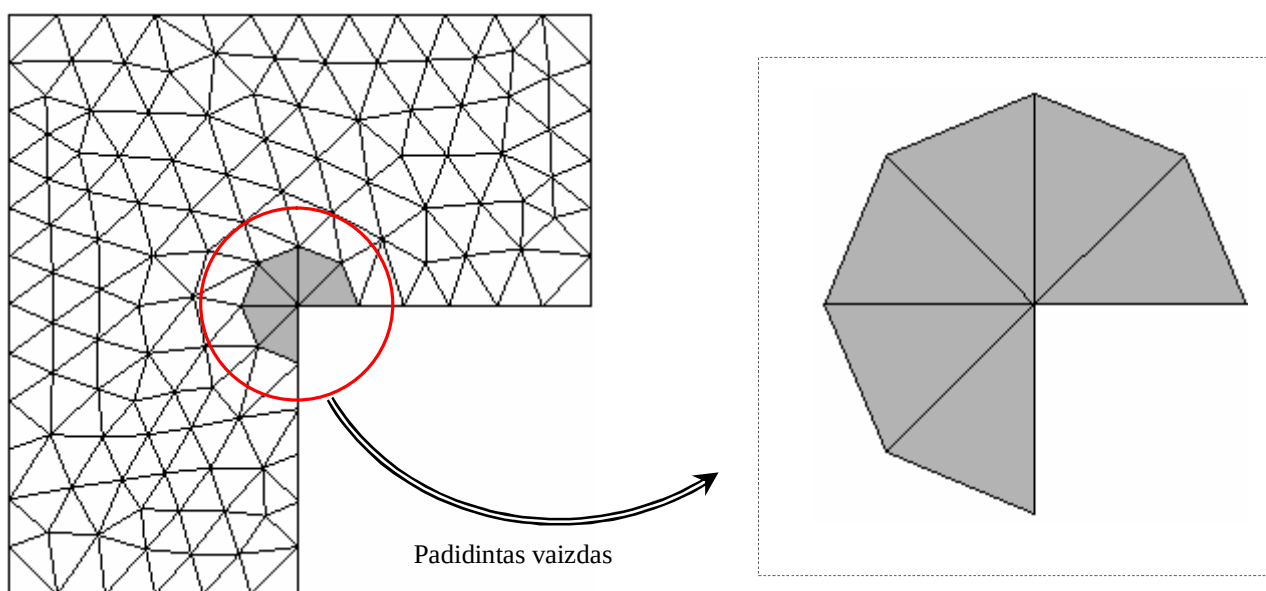
Pagal šias formules, pirmasis nagrinėjamas tinklas susidės iš 200 elementų, apytikslis elemento dydis $h \approx 0,08660$. Antrasis nagrinėjamas tinklas turės dvigubai daugiau - 400 elementų ($h \approx 0,06124$), trečiasis vėl dvigubai daugiau – 800 elementų ($h \approx 0,05303$), ketvirtasis – 1600 ($h \approx 0,03062$), penktasis – 3200 ($h \approx 0,02165$), šeštasis – 6400 ($h \approx 0,01531$). Kadangi esant 6400 elementų skaičiui dar netenkinama leistina 5 % paklaida, tai paimamas dar vienas papildomas 9500 elementų tinklas. Analogiškai pirmajam bandymui, skaičiuojame sprendinius bei paklaidas taikydami tradicinį poprocesorinį paklaidų vertinimo metodą bei singuliarių poprocesorinį paklaidų vertinimo metodą. Gautus poslinkių ir paklaidų duomenis surašome į 4.2 lentelę ir atvaizduojame atitinkamai 4.11, 4.12 ir 4.13 paveiksluose.

Lygindami grafikus matome, kad taikant singuliarių poprocesorinį paklaidų vertinimo metodą, U_x ir U_y reikšmės prie tikslaus sprendinio artėja greičiau, nei sprendinio U_x ir U_y reikšmės, gautos taikant tradicinį poprocesorinį paklaidų vertinimo metodą. Analogiškai pirmajam bandymui, paklaidos reikšmė dėl susidariusio didesnio skirtumo tarp baigtinių elementų sprendinio ir singuliaraus poprocesorinio sprendinio gaunama didesnė (200 elementų tinklui gauname 19,153 %) lyginant su paklaida, gauta pagal tradicinį poprocesorinį paklaidų vertinimo metodą (200 elementų tinklui gauname 17,592 %, arba 1,561 % mažesnę paklaidą). Paklaidų lyginimo grafikai, apskaičiuoti imant visos srities paklaidas, pavaizduoti 4.13 paveiksle. Gauti rezultatai dar kartą įrodo, jog taikant singuliaraus poprocesorinio sprendinio skaičiavimo metodą gaunami tikslesni skaičiavimo rezultatai, nei pagal tradicinį poprocesorinį paklaidų vertinimo metodą.

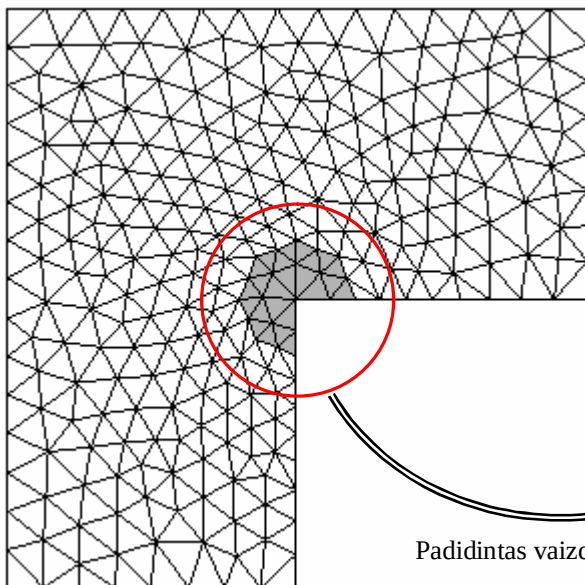
4.2 lentelė. Nereguliarių tolygių tinklų su vienoda singuliaria sritimi skaičiavimo rezultatai

Poslinkių U_X reikšmės, $e-4$								
Elementų skaičius	200	400	600	800	1600	3200	6400	9500
Poprocesorinis sprendinys be singuliarumo įvertinimo	2,283949	2,327799	2,334021	2,335659	2,349062	2,355834	2,359410	2,359757
Poprocesorinis sprendinys su singuliarumo įvertinimu	2,3163499	2,351875	2,356399	2,355657	2,357798	2,357879	2,359890	2,359904
Konvergavęs sprendinys	2,362	2,362	2,362	2,362	2,362	2,362	2,362	2,362
Poslinkių U_Y reikšmės, $e-4$								
Elementų skaičius	200	400	600	800	1600	3200	6400	9500
Poprocesorinis sprendinys be singuliarumo įvertinimo	1,362351	1,433356	1,442919	1,453558	1,470936	1,486038	1,496759	1,497821
Poprocesorinis sprendinys su singuliarumo įvertinimu	1,397178	1,461627	1,469592	1,489843	1,486863	1,489066	1,497404	1,498278
Konvergavęs sprendinys	1,511	1,511	1,511	1,511	1,511	1,511	1,511	1,511
Sprendinio paklaidos norma, %,								
Elementų skaičius	200	400	600	800	1600	3200	6400	9500
Paklaida be singuliarumo	17,592	12,506	11,235	10,350	8,336	6,363	7,101	4,520
Paklaida su singuliarumu	19,153	13,915	12,607	12,105	8,781	6,371	7,103	4,780
Leidžiama paklaida	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000

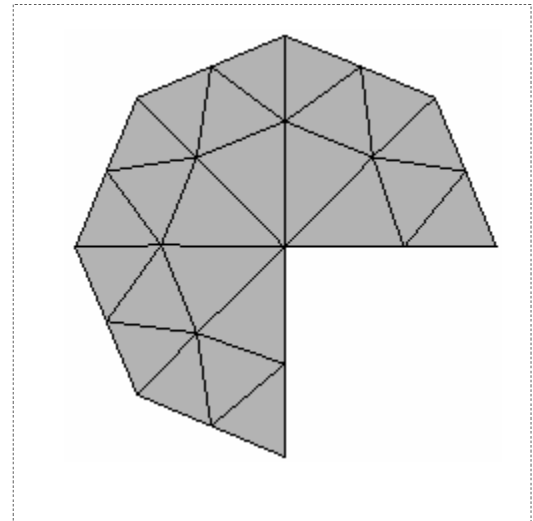
4.11 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_x konvergavimo grafikas


4.12 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: poslinkių U_y konvergavimo grafikas

4.13 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: visos nagrinėjamos srities Ω paklaidų konvergavimo palyginimas


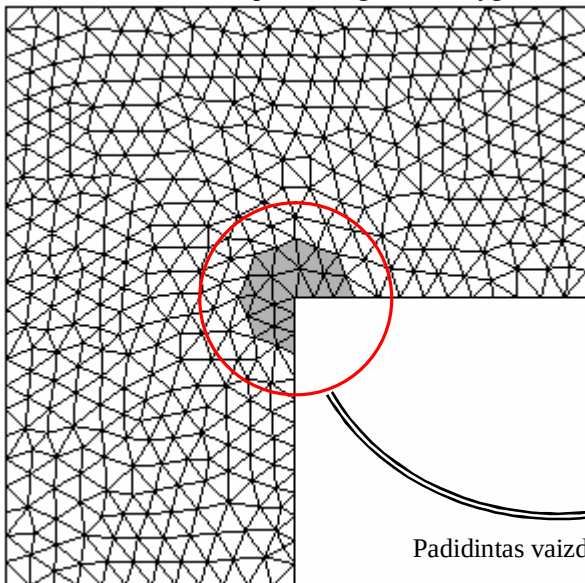
4.14 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 200 elementų tinklas



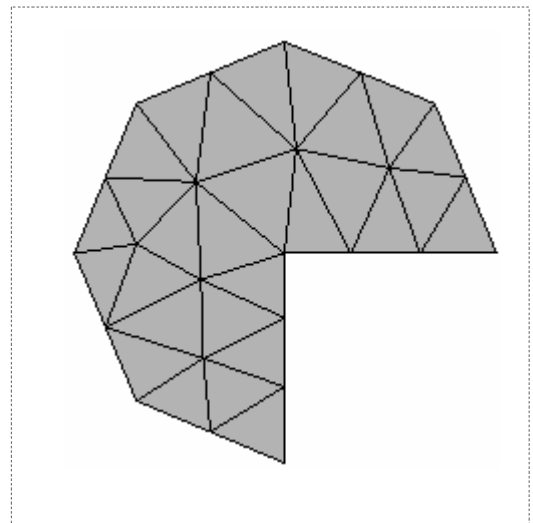
Padidintas vaizdas



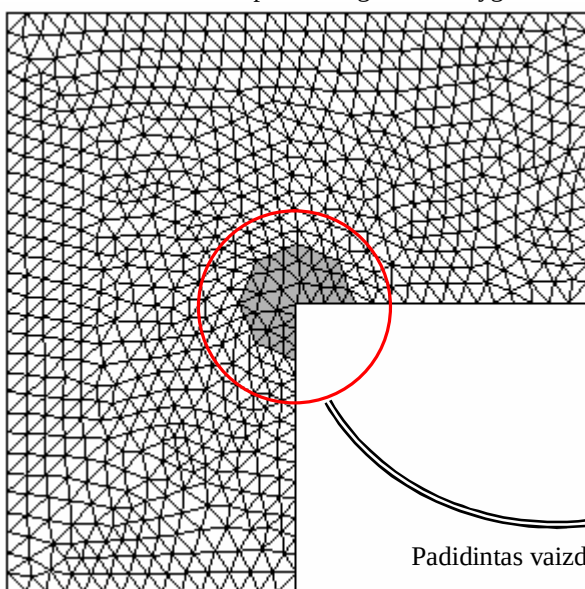
4.15 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 400 elementų tinklas



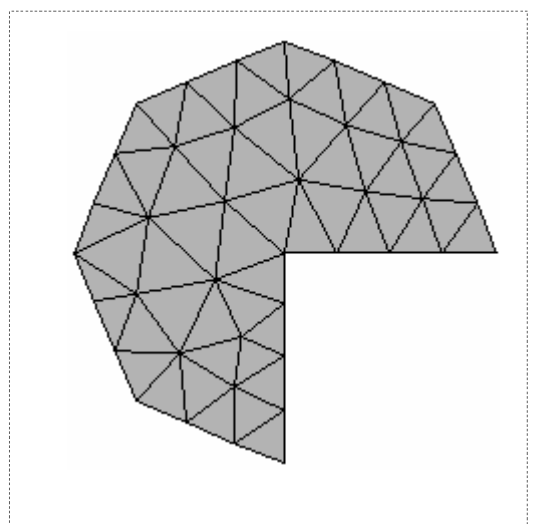
Padidintas vaizdas



4.16 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 800 elementų tinklas



Padidintas vaizdas



4.17 pav. Nereguliarūs tolygūs tinklai su vienoda singuliaria sritimi: 1600 elementų tinklas

Trečiajam eksperimentui parinkime bendriausią atvejį – imsime tolygius bet nereguliuosius tinklus, sugeneruotus neatsižvelgiant, ar (4.23) formule nusakytas singuliarus elementų fragmentas duos tą pačią singuliarią sritį kiekvienam nagrinėjamam tinklui, ar ne. Šiuo atveju kiekvieno tinklo singuliari sritis šiek tiek skirsis (į ją įeis tie L-formos sritį dengiančio trikampių elementų tinklo elementai, kurių bent vienas mazgas bus nutolęs atstumu, mažesniu nei singuliarią sritį nusakantis spindulio R dydis).

Atlikę pirmąjį eksperimentą su tolygiais reguliariais tinklais galime pastebėti, kad esant 2400 elementų skaičiui jau nėra labai didelio skirtumo tarp baigtinių elementų sprendinio ir pagerinto singuliarus sprendinio reikšmių. Antruoju atveju panašus rezultatas gaunamas esant 3200 elementų tinklui, todėl trečiajam bandymui imsime ne dvigubai mažesnio dydžio elementų tinklus, kaip kad pirmuoju atveju darėme, ir ne dvigubai tankesnius elementų tinklus, kaip kad turėjome antruoju atveju, o parinktus taip, kad aiškiau matytųsi sprendinio gerinimo rezultatai esant mažam elementų skaičiui (sprendinio pagerinimo metodų paieškos esmė ir yra gauti kuo geresnį rezultatą su kuo mažesniu elementų skaičiumi ir taip sumažinti skaičiavimo sąnaudas – juk esant labai tankiam baigtinių elementų tinklui ir taip turime sprendinį, labai artimą tikslui).

Elementų skaičius nagrinėjamuose tinkluose šį kartą kis pagal tokį dėsnį:

$$N_i \approx \begin{cases} N_0 + i * 200, & \text{kai } i \leq 8 \\ 2 * N_{i-1}, & \text{kai } 8 < i \leq 10 \\ 9500, & \text{kai } i = 11 \\ 16222, & \text{kai } i = 12 \end{cases}, \quad (4.33)$$

čia $N_0 = 206$ – pirmojo tinklo elementų skaičius, $i = 1, \dots, 12$ – tinklo eilės numeris.

Taigi pirmasis elementų tinklas turės 206 elementų (pastaba: šis tinklas yra tas pats tinklas, kaip ir 3.4.3 skyriuje nagrinėtas pradinis tolygius elementų tinklas), antrasis tinklas turės maždaug 200 elementų daugiau – 418 elementus, trečiasis turės dar 200 elementų daugiau – 576, ketvirtasis 814, penktasis – 1004, šeštasis – 1280, septintasis – 1426, aštuntasis – 1648 elementus. Kiti keturi tinklai yra sudaryti iš atitinkamai 3200, 6400, 9500 ir 16222 elementų. Pirmieji keturi šios sekos tinklai pavaizduoti 4.23 ir 4.26 paveiksluose.

Analogiškai pirmiems dviem atvejams skaičiuojame sprendinio U_x ir U_y reikšmes pagal abu minėtus poprocesorinius paklaidų vertinimo metodus: singuliarių ir tradicinį. Skaičiavimo rezultatai pateikti 4.3 lentelėje, o sprendinių U_x ir U_y konvergavimo grafikų palyginimas pateiktas 4.18 ir 4.19 paveiksluose, o paklaidų palyginimas – 4.20 ir 4.21 paveiksluose..

Lygindami poslinkių X kryptimi grafikus (4.18 pav.), matome, kad sprendinio reikšmė, rasta pagal pasiūlytą singuliarių poprocesorinį paklaidų vertinimo metodą, jau esant 1004 elementų skaičiui pasiekia reikšmę $UX_{SING,1004} = 2,35785e^{-4}$, artimą tikslui sprendiniui $UX_{TIKSLUS} = 2,362e^{-4}$ ir toliau tankinant tinklą kinta labai mažai:

$$UX_{TIKSLUS} - UX_{SING,1004} = 0,0415e^{-4} \approx 0,00417e^{-4} = UX_{TIKSLUS} - UX_{SING,16222}$$

arba

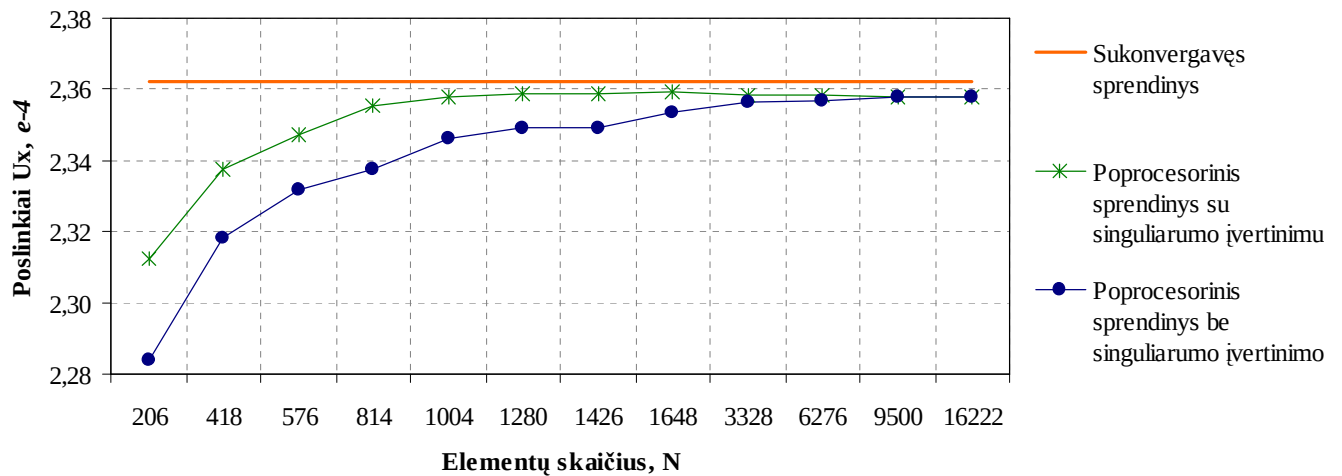
$$\lim_{N \rightarrow \infty} (UX_{TIKSLUS} - UX_{SING,N}) = 0.$$

Analizuojant poslinkių Y kryptimi grafikus (4.19 pav.), matome, kad reikšmę, artimą tiksliai sprendiniui $UX_{TIKSLUS} = 1,511e^{-4}$ gauname tik esant 6276 elementų tinklui, tačiau sprendžiant uždavinį su mažesnių elementų skaičių turinčiais tinklais, matome, kad UY reikšmės, rastos pagal singuliarių poprocesorinį paklaidų vertinimo metodą, mažiau skiriasi nuo tikslios sprendinio reikšmės, nei tradicinis poprocesorinis sprendinys. Pvz., 1004 elementų turinčio tinklo atveju:

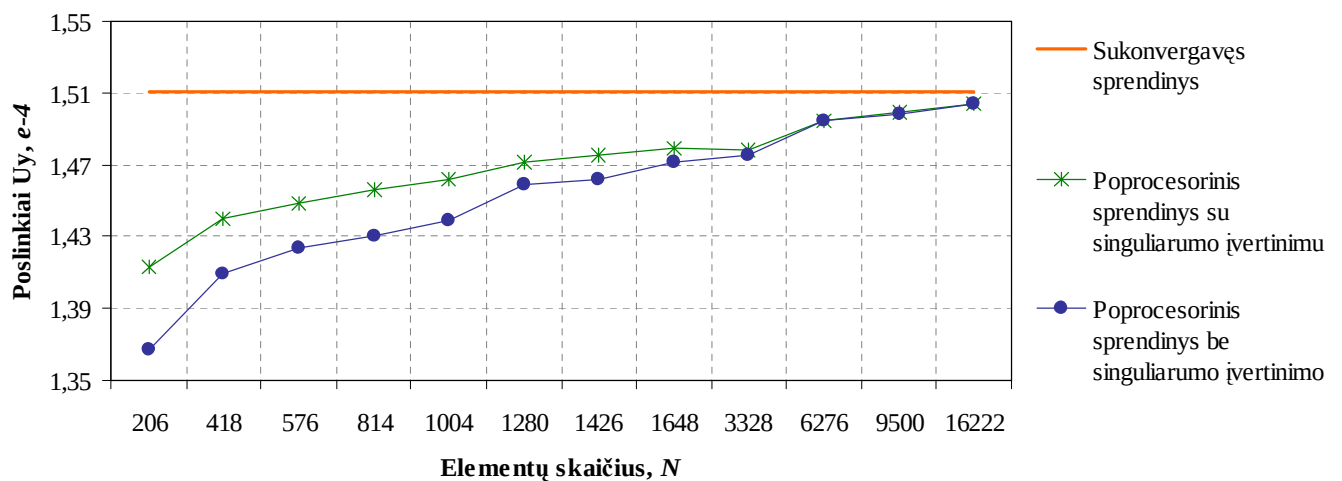
$$UY_{TIKSLUS} - UY_{SING,1004} = 0,04887e^{-4} \ll 0,07217e^{-4} = UY_{TIKSLUS} - UY_{SING,16222}.$$

Be to, kaip ir pirmais dviem atvejais, visos nagrinėjamos srities įvertintos paklaidos reikšmė, gauta pagal singuliarių poprocesorinį paklaidų vertinimo metodą, yra didesnė, nei paklaidos reikšmė, gauta pagal tradicinį poprocesorinį paklaidų vertinimo metodą (4.20 pav.). Taikydami tradicinį poprocesorinį paklaidų vertinimo metodą 206 elementų tinklui gauname 26,349 %, o naudojant singuliarių poprocesorinį paklaidų vertinimo metodą gauname 27,617 %, arba 1,268 % daugiau. Tai reiškia, kad patikslinus sprendinio reikšmę, t. y. labiau ją priartinus prie tikslaus sprendinio, poprocesorinio paklaidos skaičiavimo metu gaunamas didesnis skirtumas tarp baigtinių elementų sprendinio ir pagerinto poprocesorinio singuliaraus sprendinio, kas vėliau leidžia tiksliau generuoti naują baigtinių elementų tinklą.

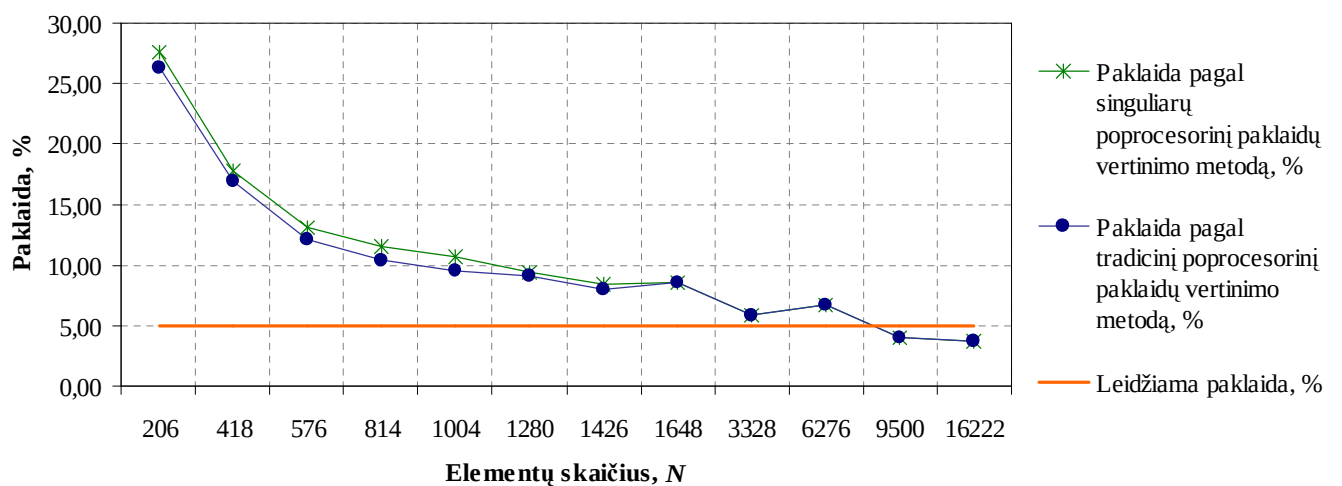
Pagerintas singuliarus poprocesorinis sprendinys skaičiuojamas tik spindulio $R = 0.1$, išvesto iš singuliaraus taško, zonoje. Likusioje srities dalyje paklaidoms vertinti imamas tradicinis poprocesorinis sprendinys, todėl sumuojant visos srities paklaidas, sprendinio ir paklaidos pagerinimas singuliaraus taško zonoje nėra toks ryškus. 4.22 paveiksle pateiktas papildomas paklaidų palyginimo grafikas, suskaičiuotas naudojant ne visos srities, o tik singuliaros zonos $R = 0,1$ paklaidų reikšmes. Šiuo atveju 206 elementų tinklo paklaidos reikšmė, gauta pagal tradicinį poprocesorinį paklaidų vertinimo metodą, yra 41,009 %, o paklaida, gauta pagal singuliarių poprocesorinį paklaidų vertinimo metodą, yra 44,517 % arba 3,508 % didesnė (4.4 lentelė). Vadinasi, pasiūlytas sprendinio skaičiavimo singuliarioje zonoje algoritmas tikrai leidžia pagerinti sprendinio reikšmę singuliaraus taško zonoje. Taikant pasiūlytąjį metodą tikslesnius rezultatus gauname jau esant mažesniai tinklo elementų skaičiui, todėl galime teigti, kad skaičiavimo sąnaudos taip pat yra mažesnės.



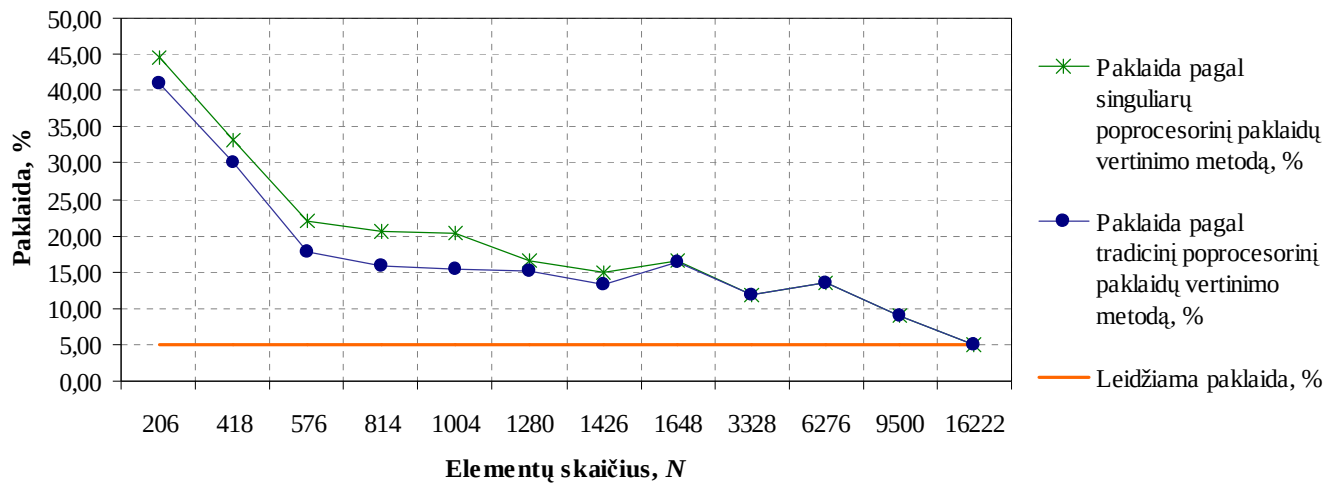
4.18 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: poslinkių U_x konvergavimo grafikas



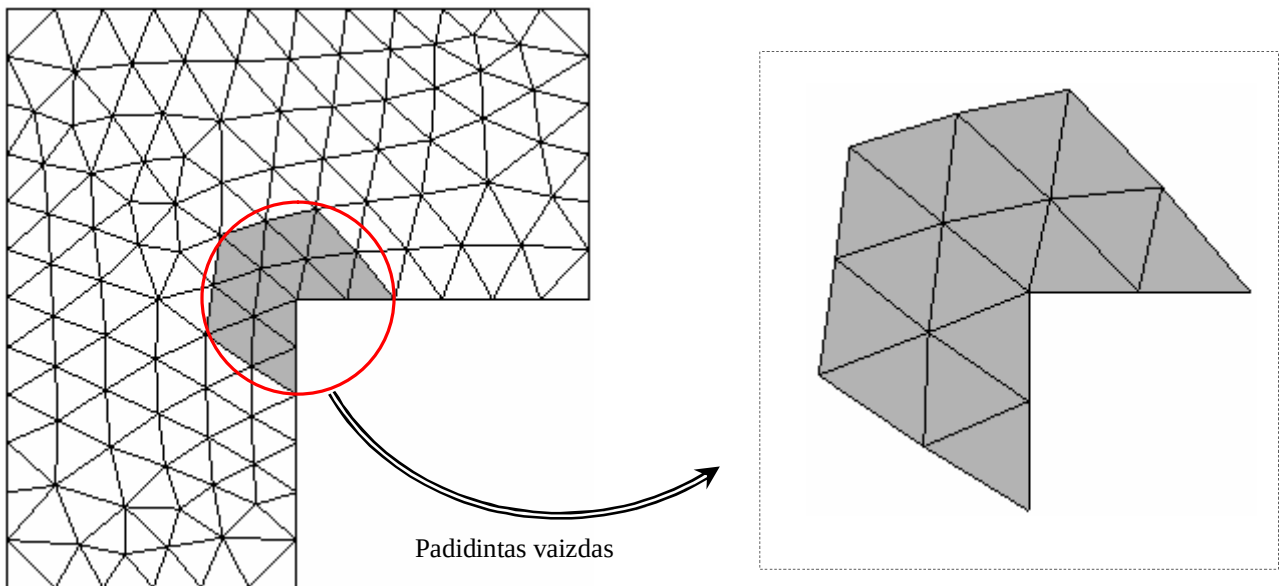
4.19 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: poslinkių U_y konvergavimo grafikas



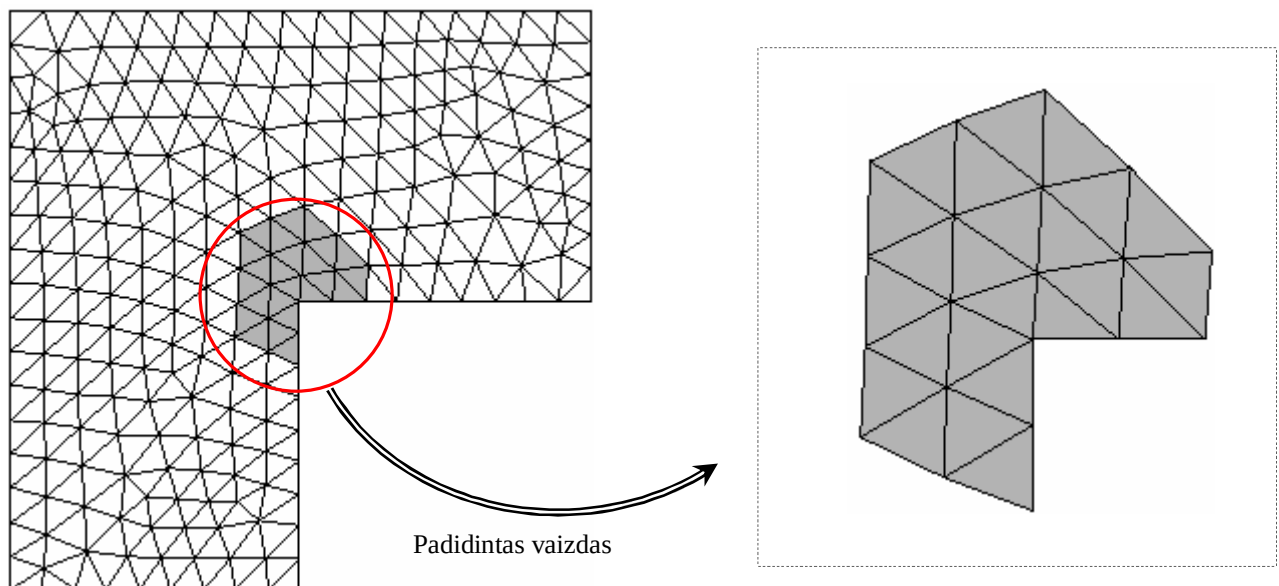
4.20 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: visos nagrinėjamos srities Ω paklaidų konvergavimo palyginimas



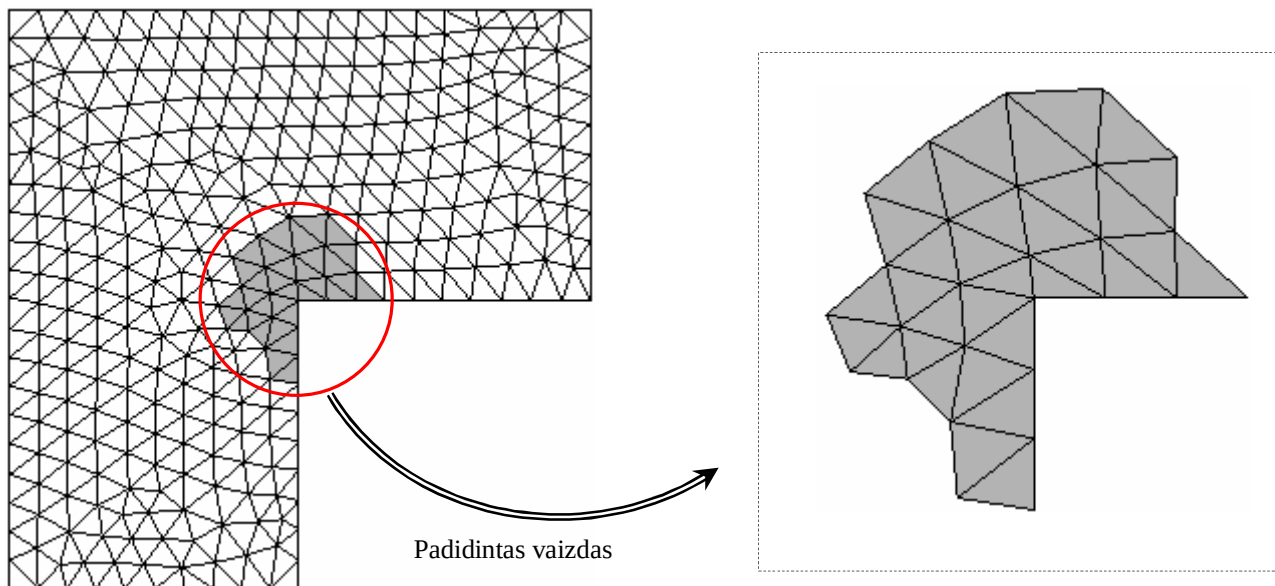
4.21 pav. Nereguliarūs tolygūs tinklai su kintama singuliaru sritimi: paklaidų konvergavimo palyginimas srityje, apribotoje spinduliu $R=0,1$, išvestu iš singuliarus srities taško



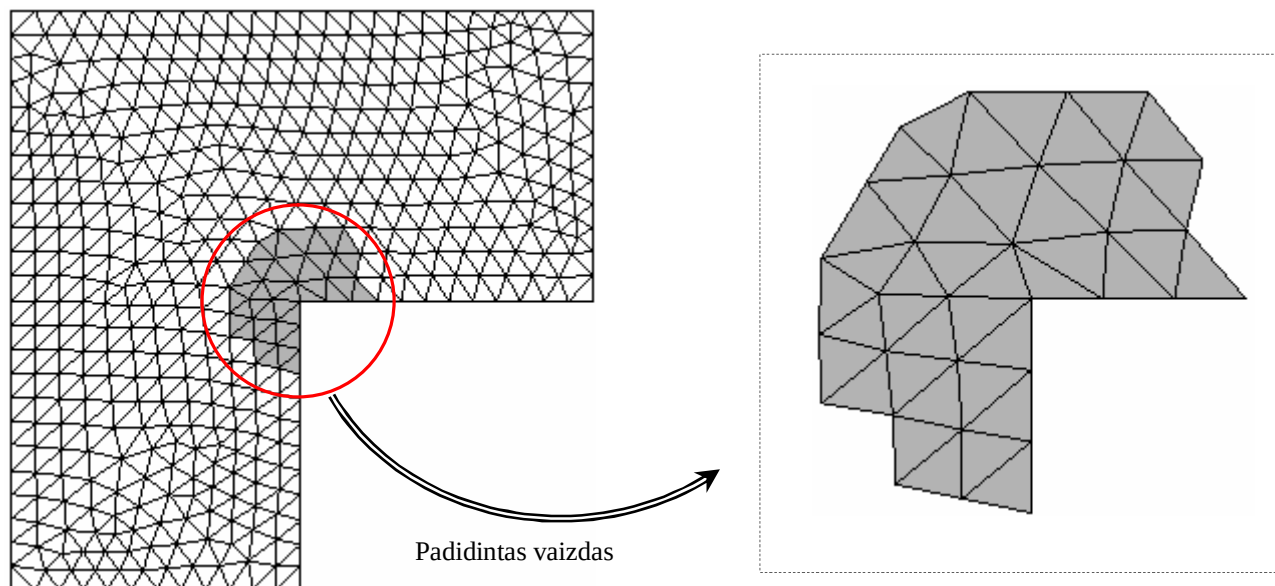
4.22 pav. Nereguliarūs tolygūs tinklai su kintama singuliaru sritimi: 216 elementų tinklas



4.23 pav. Nereguliarūs tolygūs tinklai su kintama singuliaru sritimi: 418 elementų tinklas



4.24 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 576 elementų tinklas



4.25 pav. Nereguliarūs tolygūs tinklai su kintama singuliaria sritimi: 814 elementų tinklas

Išnagrinėjus pasiūlytus pavyzdžius, matome, kad sprendinio reikšmė, gauta taikant pasiūlytą singuliarių poprocesorinį paklaidų vertinimo metodą, singuliariame taške visais atvejais buvo artimesnė tiksliam sprendiniui, nei sprendinio reikšmė, gauta taikant tradicinį poprocesorinį paklaidų vertinimo metodą. Pagerinta sprendinio reikšmė leidžia tiksliau įvertinti paklaidos reikšmę, o tikslesnė paklaida leidžia sugeneruoti geresnį baigtinių elementų tinklą kitai iteracijai, vadinasi, galutinį optimalų tinklą turėtume gauti su mažesnėmis skaičiavimo sąnaudomis.

Kitame skyrelyje įrodysime, kad taikant pasiūlytąjį metodą skaičiavimo sąnaudos tikrai sumažėja.

4.3 lentelė. Nereguliarių tolygių tinklų su kintama singuliaria sritimi poslinkių reikšmės

Poslinkių UX reikšmės, e-4								
Elementų skaičius	206	418	576	814	1004	1280	1426	1648
Popprocesorinis sprendinys be singuliarumo įvertinimo	2,284100	2,318165	2,331815	2,337685	2,346355	2,349022	2,349151	2,353390
Popprocesorinis sprendinys su singuliarumo įvertinimu	2,312419	2,337277	2,347109	2,355155	2,357850	2,358714	2,358906	2,359035
Elementų skaičius	3328	6276	9500	16222				
Popprocesorinis sprendinys be singuliarumo įvertinimo	2,356431	2,357014	2,357576	2,357708				
Popprocesorinis sprendinys su singuliarumo įvertinimu	2,358201	2,358020	2,357936	2,357825				
Konvergavęs sprendinys	2,362	2,362	2,362	2,362	2,362	2,362	2,362	2,362
Poslinkių UY reikšmės, e-4								
Elementų skaičius	206	418	576	814	1004	1280	1426	1648
Popprocesorinis sprendinys be singuliarumo įvertinimo	1,471427	1,475317	1,494346	1,498393	1,438833	1,458668	1,461805	1,471427
Popprocesorinis sprendinys su singuliarumo įvertinimu	1,479035	1,477931	1,494794	1,498880	1,462139	1,471959	1,475545	1,479035
Elementų skaičius	3328	6276	9500	16222				
Popprocesorinis sprendinys be singuliarumo įvertinimo	1,475317	1,494346	1,498393	1,504101				
Popprocesorinis sprendinys su singuliarumo įvertinimu	1,477930	1,494794	1,498880	1,504255				
Konvergavęs sprendinys	1,511	1,511	1,511	1,511	1,511	1,511	1,511	1,511

4.4 lentelė. Nereguliarių tolygių tinklų su kintama singuliaria sritimi paklaidų reikšmės

Sprendinio paklaidos norma visai nagrinėjamai sričiai, %								
Elementų skaičius	206	418	576	814	1004	1280	1426	1648
Paklaida be singuliarumo įvertinimo	26,349	16,891	12,143	10,447	9,489	9,078	7,982	8,481
Paklaida su singuliarumo įvertinimu	27,617	17,796	13,109	11,556	10,634	9,404	8,353	8,583
Elementų skaičius	3328	6276	9500	16222				
Paklaida be singuliarumo įvertinimo	5,878	6,644	3,976	3,738				
Paklaida su singuliarumo įvertinimu	5,868	6,641	3,974	3,742				
Leidžiama paklaida	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000
Sprendinio paklaidos norma singuliarioje srities zonoje, %								
Elementų skaičius	206	418	576	814	1004	1280	1426	1648
Paklaida be singuliarumo įvertinimo	41,009	30,157	17,871	15,857	15,455	15,144	13,322	16,241
Paklaida su singuliarumo įvertinimu	44,517	33,199	21,941	20,614	20,354	16,525	14,907	16,649
Elementų skaičius	3328	6276	9500	16222				
Paklaida be singuliarumo įvertinimo	11,813	13,424	8,945	4,956				
Paklaida su singuliarumo įvertinimu	11,872	13,435	8,935	4,962				
Leidžiama paklaida	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000

4.4 Paktinis prisitaikančiųjų baigtinių elementų strategijos, įvertinančios singuliaros zonos įtaką, taikymas sprendžiant plokštuminius tamprumo teorijos uždavinius

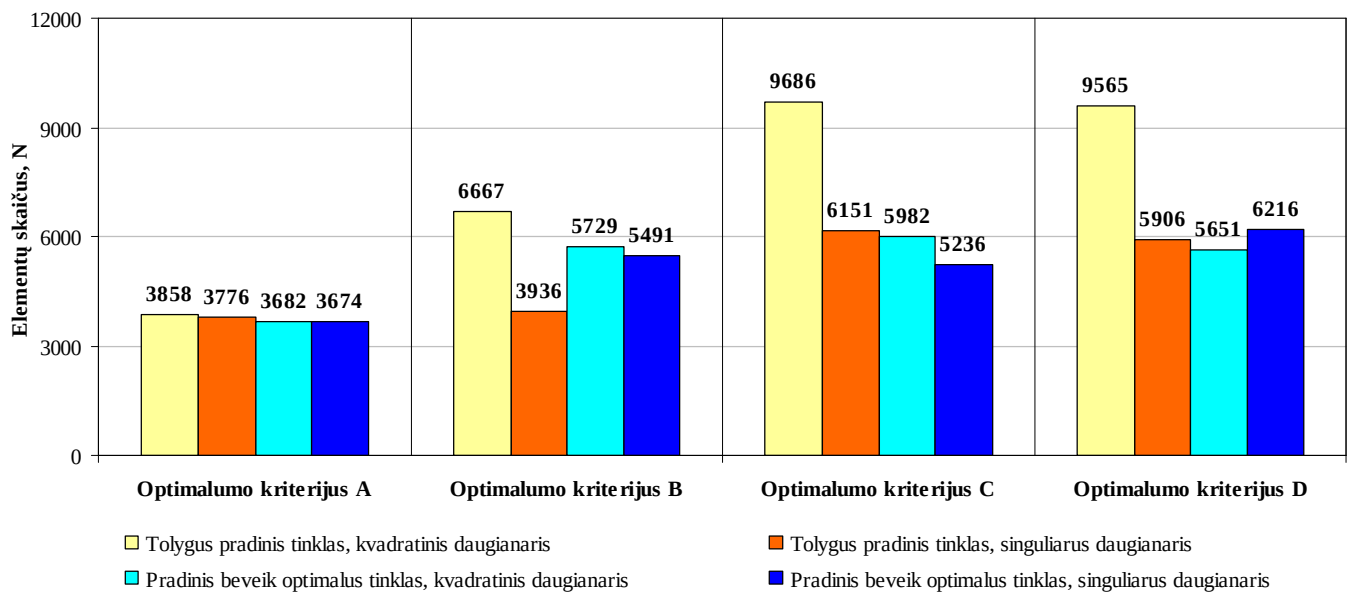
Grįžkime prie 3.4.3 nagrinėtos L-formos plokštės. Atlikę skaitinius eksperimentus, turėjome tokius rezultatus: taikydami globaliuosius tolygaus paklaidos pasiskirstymo (*Optimalumo Kriterijus A*) bei tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui optimalumo kriterijus (*Optimalumo Kriterijus B*), iteracijų, reikalingų galutiniam optimaliam tinklui gauti, skaičiaus buvo vienodas tiek naudojant tolygų pradinį tinklą, tiek pradinį beveik optimalų tinklą. Elementų skaičius, gautas naudojant pradinį beveik optimalų tinklą buvo šiek tiek mažesnis (176 elementais pagal tolygaus paklaidos pasiskirstymo optimalumo kriterijų ir 938 elementais pagal tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui optimalumo kriterijų), nei elementų skaičius, gautas pagal tolygų pradinį tinklą. Taigi dėl singuliaros taško įtakos vien tik pradinis beveik optimalus tinklas žymiai rezultatų nepagerino.

Prisitaikančiosioms baigtinių elementų strategijoms parenkant lokalius optimalumo kriterijus situacija buvo daug geresnė. Parinkus pradinį beveik optimalų tinklą ir lokalų Mizeso įtempimais paremtą optimalumo kriterijų (*Optimalumo Kriterijus C*), galutiniam tinklui gauti užteko trijų iteracijų (esant tolygiam pradiniam elementų tinklui reikėjo keturių iteracijų), elementų skaičius galutiniame tinkle, gautame pagal Mizeso įtempimais paremtą optimalumo kriterijų buvo 5982, t. y. 3704 elementais mažesnis, nei pagal tolygų pradinį tinklą. Analogiškai ir esant maksimaliam įtempimų paklaidos procentiniam santykiui (*Optimalumo Kriterijus D*) – parenkant pradinį beveik optimalų tinklą, galutiniam optimaliam tinklui gauti užteko dviejų iteracijų (pagal tolygų reikėjo trijų), o elementų skaičius tame galutiniame optimaliame tinkle buvo 5651, t. y. net 3941 elementais mažesnis, nei pagal tolygų pradinį elementų tinklą. Vadinasi, lokaliųjų optimalumo kriterijų taikymas sprendžiant sudėtingus uždavinius leidžia daug tiksliau ir greičiau gauti galutinį rezultatą.

Šiame skyrelyje atliksime tokius papildomus skaičiavimus – analogiškai kaip ir 3.4.3 skyriuje, imsime tolygų pradinį bei pradinį beveik optimalų tinklus ir su kiekvienu iš keturių 3.2 skyriuje aprašytų optimalumo kriterijų atliksime baigtinių elementų analizę. Tik šiuo atveju poprocesorinio paklaidos vertinimo metu vietoj įprasto (4.15) kvadratinio daugianario imsime singuliarų (4.18) daugianarį. Iš viso palyginimui turėsime po keturis skaičiavimo atvejus kiekvienam optimalumo kriterijui:

1. Tolygus pradinis tinklas su kvadratinio daugianariu (rezultatai atkelti iš 3.4.3 skyriaus),
2. Tolygus pradinis tinklas su singuliaru daugianariu,
3. Pradinis beveik optimalus tinklas su kvadratinio daugianariu (rezultatai atkelti iš 3.4.3 skyriaus),
4. Pradinis beveik optimalus tinklas su singuliaru daugianariu.

Imame jau nagrinėtą 206 elementų tolygų tinklą ir, taikydami singuliarų poprocesorinį paklaidų vertinimo metodą bei tolygaus paklaidos pasiskirstymo optimalumo kriterijų (A), gauname galutinį optimalų tinklą. Matome, kad elementų skaičius galutiniame tinkle yra tik šiek tiek mažesnis, nei



4.26 pav. Elementų skaičiaus pasiskirstymas galutiniuose optimaliuose tinkluose skaičiuojant pagal skirtingas strategijas

skaičiuojant su tolygiu pradiniu elementų tinklu ir taikant tradicinį poprocesorinį paklaidų vertinimo metodą, o iteracijų skaičius nesumažėjo visai (4.5 lentelė). Vadinasi, šiuo atveju didesnio tikslumo nepasiekėme. Toliau imame pradinį beveik optimalų tinklą ir pagal sprendinio pagerinimo singuliariose zonose algoritmą randame galutinį optimalų tinklą. Šį kartą tokiame tinklui rasti prireikė tik dviejų iteracijų, o elementų skaičius tame galutiniame tinkle yra mažiausias iš visų keturių nagrinėtų atvejų. Vadinasi, taikydami kartu pradinio beveik optimalaus tinklo generavimą bei sprendinio pagerinimą singuliariose zonose, jau galime sumažinti skaičiavimo sąnaudas.

Analogiškus skaičiavimus pakartosime tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui optimalumo kriterijui (B). Imame tolygų pradinį tinklą ir pagal sprendinio pagerinimo singuliariose zonose algoritmą atliekame prisitaikančiųjų baigtinių elementų analizę bei surandame galutinį optimalų tinklą. Palyginę rezultatus (4.5 lentelė), gautus naudojant kvadratinį ir singuliarų daugianarius, matome, kad nors abiem atvejais galutiniam tinklui rasti prireikė trijų iteracijų, tačiau pagal singuliaraus poprocesorinio paklaidos vertinimo algoritmą gautas galutinis tinklas turi 3936 elementus, o tai net 2731 elementu mažiau, nei pagal tradicinį algoritmą gautame galutiniame tinkle. Dabar imame pradinį beveik optimalų tinklą ir pagal sprendinio pagerinimo singuliariose zonose algoritmą randame galutinį optimalų tinklą, kuriam apskaičiuoti šį kartą prireikė tik dviejų iteracijų. Be to, elementų skaičius šiame tinkle yra mažiausias iš visų keturių optimalių tinklų. Taigi tolygaus paklaidos kvadrato pasiskirstymo optimalumo kriterijui singuliarumo įvertinimas geresnius rezultatus davė tiek esant tolygiam pradiniam tinklui (žymiai mažesnis elementų skaičius), tiek pradiniam beveik optimaliam tinklui (mažiausias iteracijų ir elementų skaičius).

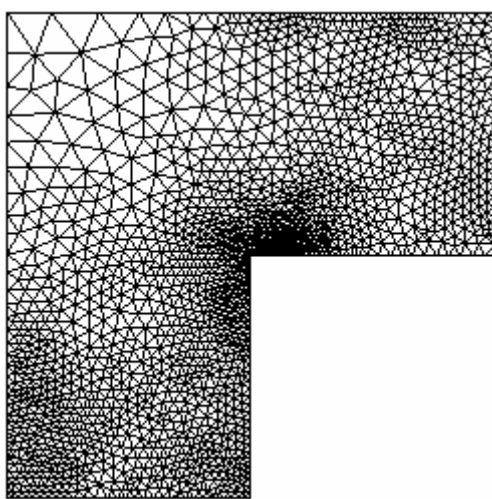
4.5 lentelė. L-formos plokštės skaičiavimo rezultatai, $h = 5\%$

Tolygaus paklaidos pasiskirstymo optimalumo kriterijus (A)	1-oji iteracija		2-oji iteracija		3-oji iteracija		4-oji iteracija	
	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
Tolygus pradinis tinklas, kvadratinis daugianaris	26,35	206	5,81	2831	4,74	3858	-	-
Tolygus pradinis tinklas, singuliarus daugianaris	27,62	206	5,56	3057	4,72	3776	-	-
Pradinis beveik optimalus tinklas, kvadratinis daugianaris	19,39	206	5,15	3267	4,88	3682	-	-
Pradinis beveik optimalus tinklas, singuliarus daugianaris	21,22	206	4,79	3674	-	-	-	-
Tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui optimalumo kriterijus (B)	1-oji iteracija		2-oji iteracija		3-oji iteracija		4-oji iteracija	
	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
Tolygus pradinis tinklas, kvadratinis daugianaris	26,35	206	8,16	2841	4,68	6667	-	-
Tolygus pradinis tinklas, singuliarus daugianaris	27,62	206	5,48	3069	4,90	3936	-	-
Pradinis beveik optimalus tinklas, kvadratinis daugianaris	19,39	206	5,06	4646	4,76	5729	-	-
Pradinis beveik optimalus tinklas, singuliarus daugianaris	21,22	206	4,99	5491	-	-	-	-
Mizeso įtempimai (C)	1-oji iteracija		2-oji iteracija		3-oji iteracija		4-oji iteracija	
	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
Tolygus pradinis tinklas, kvadratinis daugianaris	26,35	206	8,50	2493	5,64	3806	4,21	9686
Tolygus pradinis tinklas, singuliarus daugianaris	27,62	206	6,51	2782	4,63	6151	-	-
Pradinis beveik optimalus tinklas, kvadratinis daugianaris	19,39	206	5,98	3595	4,74	5982	-	-
Pradinis beveik optimalus tinklas, singuliarus daugianaris	21,22	206	6,00	4217	4,82	5236	-	-
Maksimalaus įtempimų paklaidos procentinio santykio optimalumo kriterijus (D)	1-oji iteracija		2-oji iteracija		3-oji iteracija		4-oji iteracija	
	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius	$h, \%$	Elementų skaičius
Tolygus pradinis tinklas, kvadratinis daugianaris	26,35	206	5,26	4390	4,17	9565	-	-
Tolygus pradinis tinklas, singuliarus daugianaris	27,62	206	4,84	5906	-	-	-	-
Pradinis beveik optimalus tinklas, kvadratinis daugianaris	19,39	206	4,50	5651	-	-	-	-
Pradinis beveik optimalus tinklas, singuliarus daugianaris	21,22	206	4,46	6216	-	-	-	-

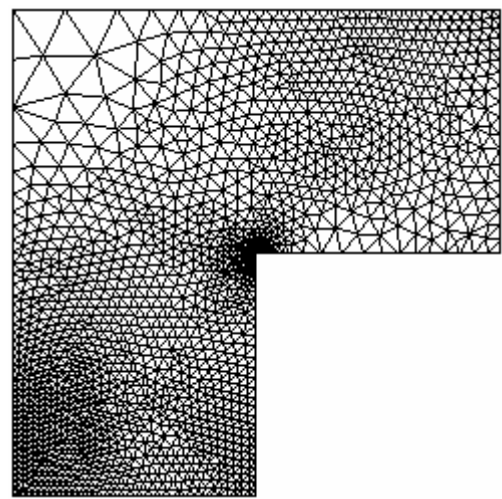
Nors lokalieji optimalumo kriterijai L-plokštės uždavinio atveju (3.4.3 skyrius) davė pakankamai gerus rezultatus, tačiau įdomumo dėlei ir šiems kriterijams atliksime analizę taikydami singuliarių poprocesorinį paklaidų vertinimo metodą. Pradėsime nuo lokalojo Mizeso įtempimais paremto optimalumo kriterijaus C. Vėl imame tą patį 206 elementų tinklą ir, įvertindami singuliarumą, skaičiuojame galutinius optimalius tinklus tolygiam pradiniam ir pradiniam beveik optimaliam tinklams. Gauname: esant tolygiam pradiniam tinklui su kvadratinio daugianariu turime keturias iteracijas ir 9686 elementus; vietoj kvadratinio daugianario parinkę singuliarų, gauname tris iteracijas ir 6151 elementus

(viena iteracija ir 3535 elementais mažiau, nei pirmuoju atveju); imdami pradinį beveik optimalų tinklą ir kvadratinį daugianarį, turime tris iteracijas ir 5982 elementus (viena iteracija mažiau, nei pirmuoju atveju, ir 169 elementais mažiau, nei antruoju atveju); paėmę pradinį beveik optimalų tinklą ir singuliarių daugianarį turime tris iteracijas ir 5236 elementus (viena iteracija mažiau, nei pirmuoju atveju, ir 746 elementais mažiau, nei trečiuoju atveju). Paskutiniuoju atveju gauname mažiausią elementų skaičių (4.5 lentelė). Vadinasi, kaip ir esant pirmiems dviems globaliesiems optimalumo kriterijams, lokaliajo Mizeso įtempimais paremto optimalumo kriterijaus atveju pradinio beveik optimalaus tinklo parinkimas ir sprendinio patikslinimas singuliariose zonose taip pat duoda mažiausias skaičiavimo sąnaudas.

Išanalizavę paskutinį lokalųjį maksimalaus įtempimų paklaidos procentinio santykio optimalumo kriterijų (D), gauname: tolygaus pradinio tinklo su kvadratinio daugianariu atveju turime tris iteracijas ir 9565 elementus; vietoj kvadratinio daugianario parinkę singuliarių, gauname dvi iteracijas ir 5906 elementus (viena iteracija ir 3659 elementais mažiau, nei pirmuoju atveju); imdami pradinį beveik optimalų tinklą ir kvadratinį daugianarį, turime dvi iteracijas ir 5651 elementus (viena iteracija mažiau, nei pirmuoju atveju ir 255 elementais mažiau, nei antruoju atveju); parinkę pradinį beveik optimalų tinklą ir singuliarių daugianarį turime dvi iteracijas ir 6216 elementus (viena iteracija mažiau, nei pirmuoju atveju, ir 565 elementais daugiau, nei trečiuoju atveju). Mažiausią iteracijų ir elementų skaičių šiuo atveju gauname naudodami pradinį beveik optimalų tinklą ir kvadratinį daugianarį (4.5 lentelė). Singuliaraus sprendinio naudojimas šio optimalumo kriterijaus atveju davė labai panašius rezultatus, todėl galima teigti, kad prisitaikančiųjų baigtinių elementų strategijų algoritmui vietoj globaliojo optimalumo kriterijaus parinkus lokalųjį, galutinį optimalų tinklą gausime jau su pakankamai mažomis skaičiavimo sąnaudomis, o pridėję singuliaraus sprendinio skaičiavimą tam tikrais atvejais galėsime tikėtis dar tikslesnių rezultatų.

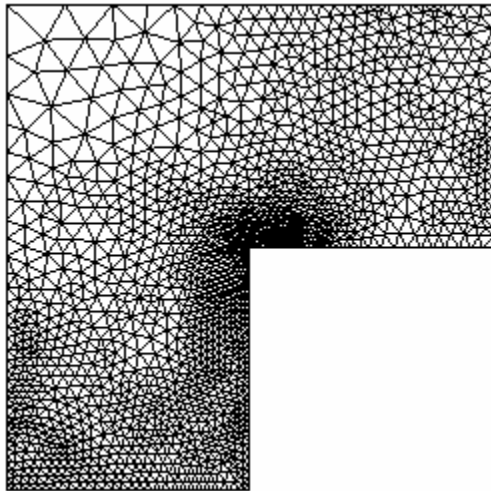


(a)

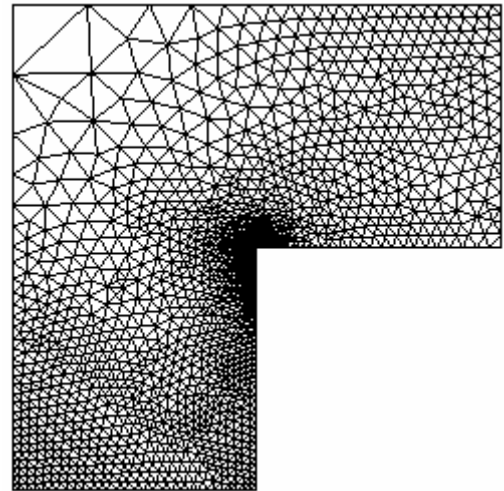


(b)

4.27 pav. Galutiniai tinklai, gauti pagal kriterijų A naudojant singuliarių daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui

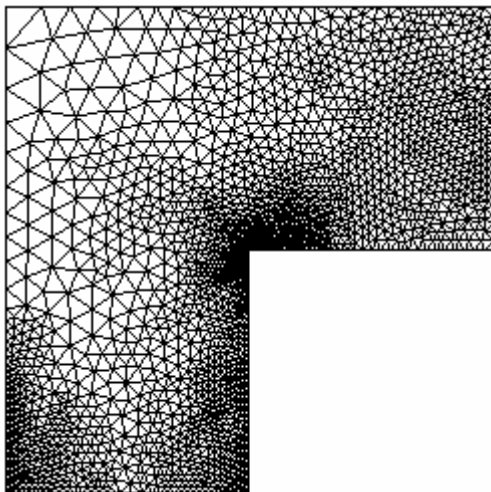


(a)

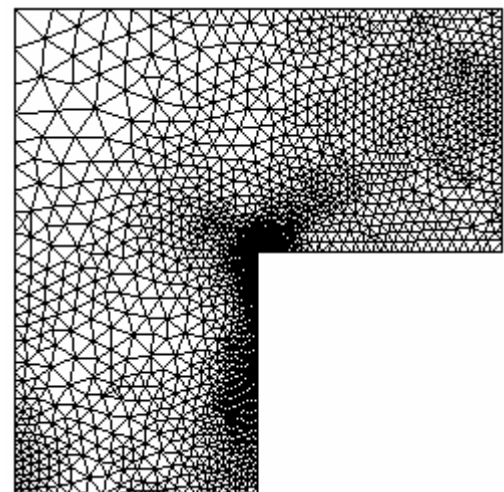


(b)

4.28 pav. Galutiniai tinklai, gauti pagal kriterijų B naudojant singuliarių daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui

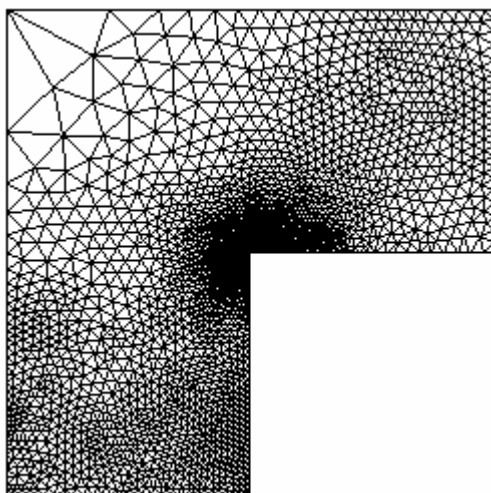


(a)

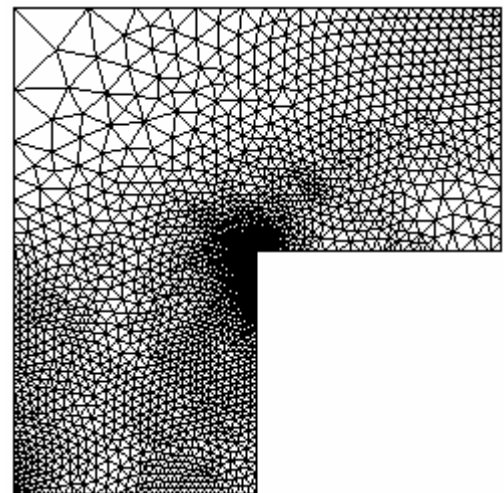


(b)

4.29 pav. Galutiniai tinklai, gauti pagal kriterijų C naudojant singuliarių daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui



(a)



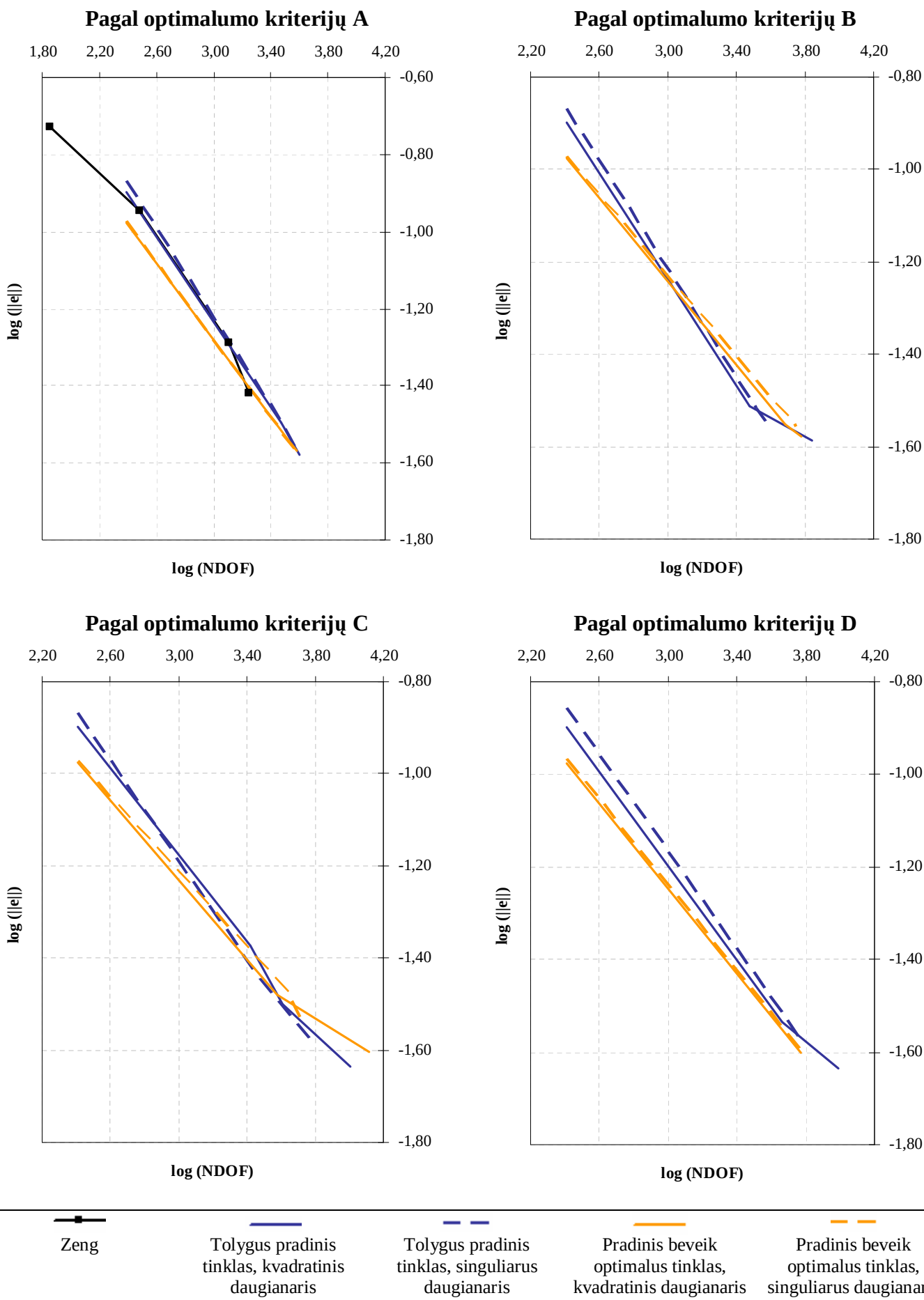
(b)

4.30 pav. Galutiniai tinklai, gauti pagal kriterijų D naudojant singuliarių daugianarį: (a) – esant tolygiam pradiniam tinklui, (b) – esant beveik optimaliam pradiniam tinklui

Lygindami galutinius optimalius tinklus, gautus pagal keturis nagrinėtus optimalumo kriterijus, matome, kad elementų dydis ir pasiskirstymas juose nėra vienodas. Esant lokalesiems optimalumo kriterijams didesnis kiekis mažiausio dydžio elementų sukoncentruojama ten, kur tikimasi gauti didesnius įtempimų gradientus (pvz., singuliariume taške), o didėjant atstumui nuo singuliaraus taško, elementų dydis auga ir jų tankis retėja labiau, nei globalių optimalumo kriterijų atveju, kai elementų tankis ir elementų dydis pasikirsto tinkle daug tolygiau. Kaip matėme iš skaitinių pavyzdžių, toks didesnio kiekio ir mažesnio dydžio elementų sukoncentravimas probleminėse srityse leido gauti rezultatą su mažesnėmis skaičiavimo sąnaudomis (mažesniu iteracijų skaičiumi). Tačiau taikant globalius optimalumo kriterijus galutiniai optimalūs tinklai turi mažesnę elementų skaičių. Pavyzdžiui, tolygaus pradinio tinklo su singuliarium daugianariu atveju globaliesiems optimalumo kriterijams A ir B galutiniuose tinkluose turime atitinkamai po 3776 ir 3936 elementus, o lokalesiems optimalumo kriterijams C ir D – po 6151 ir 5906 elementus, t. y. lokalieji optimalumo kriterijai duoda apie 30 % daugiau elementų, nei globalieji optimalumo kriterijai. Elementų skaičiaus priklausomybės nuo pasirinkto optimalumo kriterijaus grafinis palyginimas pateiktas 4.26 pav.

Siekiant geriau išnagrinėti pasiūlytų strategijų konvergavimo savybes, 4.31 pav. pateikiamos $\log\|e\|_{E(W)} - \log(NDOF)$ diagramos, čia $NDOF$ – laisvės laipsnių skaičius, $\|e\|_{E(W)}$ – visos srities diskretizacijos paklaidos energetinė norma. Gautų rezultatų palyginimui parenkame Zeng (1991) pateiktus skaičiavimų duomenis, gautus taikant prisitaikančiąsias baigtinių elementų strategijas su tiesiniais trikampaiais elementais.

Kiekvienam optimalumo kriterijui sukonstruojame po keturias konvergavimo kreives – pagal rezultatus, gautus taikant prisitaikančiąsias BE strategijas su tolygiu pradiniu tinklu ir kvadratinium daugianariu, su tolygiu pradiniu tinklu ir singuliarium daugianariu, su pradiniu beveik optimaliu tinklu ir kvadratinium daugianariu bei su pradiniu beveik optimaliu tinklu ir singuliarium daugianariu. Lygindami kreives, gautas prisitaikančiųjų BE pirmai iteracijai parinkus pradinį beveik optimalų tinklą ir optimalumo kriterijų A, matome, kad jos beveik sutampa su Zeng pateikta konvergavimo kreive tiek atsižvelgiant į singuliarumo įtaką, tiek ir neatsižvelgiant. Tuo tarpu kreivės, gautos pirmai prisitaikančiųjų BE strategijų iteracijai parinkus tolygų pradinį tinklą ir optimalumo kriterijų A, yra žemiau nuo palyginamosios kreivės. Tai reiškia, kad pradinio beveik optimalaus tinklo parinkimas leidžia tiksliau aproksimuoti paklaidas nei tolygaus pradinio tinklo parinkimas. Taip pat matome, kad poprocesorinio pagerinto singuliaraus sprendinio taikymas prisitaikančiose BE strategijose duoda statesnes kreives lyginant su paklaidos kreive, gauta neatsižvelgiant į singuliarumą. Tai reiškia, kad singuliarių zonų įtakos įtraukimas į prisitaikančiąsias strategijas leidžia padidinti konvergavimo greitį ir sumažinti iteracijų, reikalingų galutiniam optimaliam tinklui gauti, skaičių.



4.31 pav. Konvergavimo greičio palyginimas

4.5 Išvados

Šiame skyriuje papildomam poprocesorinio sprendinio įvertinimui singuliaraus taško srityje pateikiama nauja strategija, leidžianti sumažinti skaičiavimo sąnaudas ieškant galutinio optimalaus tinklo:

- lyginant su tradiciniu poprocesoriniu sprendiniu, pagerintas singuliarus poprocesorinis sprendinys reikšmę, labai artimą tiksliai sprendiniui, įgyja esant maždaug 3–6 kartus mažesniai elementų skaičiui tinkle;

- taikant pagerinto singuliaraus poprocesorinio sprendinio radimo algoritmą kiekvienoje prisitaikančiųjų baigtinių elementų strategijos iteracijoje gaunama tikslesnė sprendinio reikšmė, nei taikant tradicinį poprocesorinį sprendinio radimo metodą, o tai leidžia tiksliau aproksimuoti paklaidą ir galutinį optimalų tinklą gauti su mažesnėmis skaičiavimo sąnaudomis;

- jei baigtinių elementų analizė atliekama parinkus tolygaus paklaidos pasiskirstymo optimalumo kriterijų, singuliaraus poprocesorinio sprendinio radimo algoritmo taikymas didelės įtakos galutiniam elementų skaičiui tinkle neturi, tačiau parinkus tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui, Mizeso įtempimais paremto ar įtempimų paklaidos procentinio santykio optimalumo kriterijus, elementų skaičių galutiniame tinkle galima sumažinti apie 40 %.

- apjungę pradinio beveik optimalaus tinklo generavimo algoritmą, singuliaraus poprocesorinio sprendinio radimo metodiką bei parinkę skirtingus optimalumo kriterijus, gauname naują metodą, leidžiantį atlikti prisitaikančiųjų baigtinių elementų analizę su mažesniu iteracijų skaičiumi ir minimaliu elementų kiekiu galutiniame optimaliame tinkle.

5. DARBO REZULTATŲ APIBENDRINIMAS

Atlikus jau esamų baigtinių elementų metodų analizę, nustatyta, jog šios metodikos ne visais atvejais tinka plokštuminiams tamprumo teorijos uždaviniams spręsti, todėl šiame darbe pasiūlytos naujos strategijos, pagerinančios jau esamų strategijų efektyvumą bei tinkamumą inžinerinių uždavinių sprendimui:

1. Disertacijoje pateikta nauja strategija, leidžianti generuoti pradinį beveik optimalų baigtinių elementų tinklą bei reguliuoti elemento dydį įvairiose kritinėse konstrukcijos srityse. Pagrindinis dėmesys skiriamas priešprocesorinei procedūrai, kuri nustato elemento dydį dar prieš prasidedant baigtinių elementų analizei. Elemento dydžio funkcija, gauta taikant kontrolinės erdvės bei kontrolinės sferos apibrėžimus, visiškai kontroliuoja elementų dydžio pasiskirstymą sudėtingose konstrukcijose.
2. Išnagrinėjus skaitinius pavyzdžius, nustatyta, kad prisitaikančiųjų baigtinių elementų analizės algoritmo pirmai iteracijai parinkus pradinį beveik optimalų tinklą, sugeneruotą pagal pasiūlytą strategiją, galutinį optimalų tinklą galima gauti per dvi iteracijas ir su 20–40 % mažesniu elementų skaičiumi.
3. Pradinio beveik optimalaus tinklo parinkimas iteracijų ir elementų skaičių galutiniame optimaliame tinkle leido sumažinti tik tuose uždaviniuose, kurie neturėjo tipinių singuliarių taškų.
4. Išnagrinėjus globalius bei lokalius optimalumo kriterijus, galima teigti, kad mažiausią elementų skaičių gauname taikydami tolygaus paklaidos pasiskirstymo optimalumo kriterijų, o parinkus tolygaus paklaidos kvadrato pasiskirstymo ploto vienetui, Mizeso įtempimais paremto ar maksimalaus įtempimų paklaidos procentinio santykio optimalumo kriterijus, elementų skaičius gali išaugti 5 kartus.
5. Elementų dydžio ir tankio kitimas taikant tolygaus paklaidos pasiskirstymo optimalumo kriterijų yra ne toks žymus, kaip galutiniuose optimaliuose tinkluose, gautuose parinkus kitus tris optimalumo kriterijus, kurie didesnę kiekį smulkesnių elementų koncentruoja tose konstrukcijos vietose, kur tikimasi gauti didžiausius įtempimų gradientus.
6. Išnagrinėjus skaitinius pavyzdžius galima konstatuoti, kad parenkant lokaliuosius optimalumo kriterijus ir pradinį beveik optimalų tinklą, uždavinių, turinčių tipinių singuliarių taškų, sprendimui iteracijų skaičius sumažėjo iki 2, o elementų skaičius – apie 40 %, tuo tarpu taikant globalius optimalumo kriterijus iteracijų skaičius liko toks pat, o elementų skaičius sumažėjo tik 4–14 %.

7. Išnagrinėjus esamus poprocesorinio sprendinio paklaidos skaičiavimo singuliaraus taško aplinkoje metodus, nustatyta, kad jų pritaikymas h-prisitaikančiųjų baigtinių elementų strategijoms reikalauja didelių pakeitimų. Dėl šios priežasties pasiūlytas naujas lengvai realizuojamas poprocesorinio sprendinio radimo singuliaraus taško aplinkoje metodas, tinkantis būtent h-prisitaikančioms baigtinių elementų strategijoms.
8. Lyginant su tradiciniu poprocesoriniu sprendiniu, pagerintas singuliarus poprocesorinis sprendinys reikšmę, labai artimą tiksliam sprendiniui, įgyja esant maždaug 3–6 kartus mažesniai elementų skaičiui tinkle.
9. Baigtinių elementų analizę pradėjus nuo pradinio beveik optimalaus tinklo ir kiekvienoje prisitaikančiųjų baigtinių elementų strategijos iteracijoje suradus pagerintą singuliarų poprocesorinį sprendinį galutinį optimalų tinklą per dvi iteracijas galima gauti ne tik lokaliems optimalumo kriterijams, bet ir globaliems taip pat.
10. Taikant kartu klasikinį prisitaikančiųjų strategijų algoritmą, pradinio beveik optimalaus tinklo generavimo procedūrą, skirtingų optimalumo kriterijų parinkimą bei sprendinio radimo singuliaraus taško aplinkoje metodą, gautas naujas, labai efektyvus algoritmas, pagal kurį galutinio optimalaus tinklo generavimas įtakojamas dviem būdais: iteracijų, reikalingų optimaliam tinklui gauti, skaičius bei elementų skaičius tame galutiniame optimaliame tinkle sumažinamas parenkant pradinį beveik optimalų tinklą bei tiksliau apskaičiuojant sprendinį singuliariose zonose, o elementų dydis bei pasiskirstymas tame galutiniame optimaliame tinkle reguliuojamas parenkant vieną ar kitą optimalumo kriterijų.

6. CITUOTOS LITERATŪROS SĄRAŠAS

Kitų autorių:

- Aalto J. M. (1997) Built-in field equations for recovery procedures // *Comput. Struct.*, 64, p. 157–176.
- Aalto J. M. (1999) On patch recovery methods and corner singularities // *European Conference on Computational Mechanics*, Munchen, Germany, 25 p. in CD-ROM.
- Aalto J. M. and Kuula K. A. (1998). Mapped strips of elements for singularities and unbounded domains // *WCCM IV*, Buenos Aires, 20 p. in CD-ROM.
- Akin J. E. (1994) *Finite elements for analysis and design* // Academic Press, London, England, 560 p.
- Apel Th. and Milde F. (1996) Comparison of some mesh refinement strategies near edges // *Comm. Numer. Methods Engrg.*, 12, p. 373–381.
- Apel Th. and Nicaise S. (1998) The finite element method with anisotropic mesh grading for elliptic problems in domains with corners and edges // *Math. Methods Appl. Sci.*, 21, p. 519–549.
- Atkočiūnas J. and Merkevičiūtė D. (2004) Optimal shakedown design for bar systems: strength, stiffness and stability constraints // *Proceedings of the Seventh International Conference on Computational Structures Technology*. Lisbon, Portugal 7-9 September 2004 / Civil-Comp Press. Stirling: Civil-Comp Press, p. 361–362.
- Atkočiūnas J. ir Nagevičius J. (2004) *Tamprumo teorijos pagrindai* // Bendrasis vadovėlis. Vilnius, Technika, 528 p.
- Atkočiūnas J., Jarmolajeva E. ir Merkevičiūtė D. (2004) Optimal shakedown loading for circular plates. *Structural and Multidisciplinary Optimization* // *International Society for Structural and Multidisciplinary Optimization (ISSMO)*, 27(3), p. 178–188.
- Babuška I. and Aziz A. K. (1976) On the angle condition in finite element method // *SIAM Journal of Numerical Analysis*, 13, p. 214–226.
- Babuška I. and Suri M. (1990) The p-and h-p versions of the finite element method – an overview // *Comput. Methods Appl. Mech. Eng.*, 80, p. 5–26.
- Babuška I., Szabo B. A. and Katz I. N. (1981) The p-version of the Finite Element Method // *SIAM J. Numer. Anal.*, 18, p. 515–545.
- Babuška I., Zienkiewicz O. C., Gago J. and Oliveira E. R. (1986) Accuracy estimates and adaptive refinements in finite element computations // New York: Wiley, 296 p.
- Babuška I., Zienkiewicz O. C., Gago J. and Olivera E. R. de A. (1986) Accuracy Estimates and Adaptive Refinements in Finite Element Computations // John Wiley & Sons, 408 p.
- Bank R E. and J. Xu (1996) An algorithm for coarsening unstructured meshes // *Numerische Mathematik*, 73, p. 1–36.
- Barauskas R. (1998) *Baigtinių elementų metodo pagrindai* // Kaunas, Technologija, 375 p.
- Barauskas R., Kačianauskas R. ir Belevičius R. (2004) *Baigtinių elementų metodo pagrindai* // Technika, 612 p.
- Barauskas R., Kuprys M. ir Leonavičiūtė U. (2003). Object-oriented implementation of transient wave propagation finite element models // *Informacines technologijos ir valdymas / Kauno technologijos universitetas*, 4(29), p. 30–38.
- Barsoum R. S (1976) On the use of isoparametric finite elements in linear fracture mechanics // *Int. J. Num. Meth. Engng.*, 10, p. 25–37.

- Bathe K. J. (1978) Numerical methods in finite element analysis // Prentice-Hall, Englewood Cliffs, New Jersey, USA, 528 p.
- Bathe K. J. (1996) Finite element procedures // Prentice Hall, Englewood Cliffs, New Jersey, USA, 1028 p.
- Baušys R. and Wiberg N. E. (1999) Adaptive finite element strategy for acoustic problems // Journal of Sound and Vibration, 226, p. 905–922.
- Baušys R. ir Wiberg N. E. (1999) Adaptive finite element strategy for acoustic problems // Journal of Sound and Vibration, 226, p. 79–89.
- Baušys R. ir Wiberg N. E. (2000) Postprocessing techniques in finite element analysis // Mechanika, 2(22), p. 58–64.
- Belhachmi Z. Sac-Epée J. M. and Sokolowski J. (2005) Mixed Finite Element Methods for Smooth Domain Formulation of Crack Problems // SIAM Journal on Numerical Analysis, 43(3), p. 1295–1320.
- Bhatia R. P. and Lawrence K. L. (1990) Two-dimensional finite element mesh generation based on stripwise automatic triangulation // Computers & Structures, 36, p. 309–319.
- Bochev P. and Gunzburger M. D. (2006) Least-Squares Finite Element Methods for Optimality Systems Arising in Optimization and Control Problems // SIAM Journal on Numerical Analysis, 43(6), p. 2517–2543.
- Borooman B. and Zienkiewicz O. C. (1997) Recovery equilibrium in patches (REP) // Int. J. Numer. Methods Eng., 40, p. 137–164.
- Borouchaki H., Hecht F. and Frey P. J. (1998) Mesh gradation control // International Journal of Numerical Methods in Engineering, 43, p. 1143–1165.
- Braack M. and Ern A. (2003) A Posteriori Control of Modeling Errors and Discretization Errors // SIAM Multiscale Modeling & Simulation, 1(2), p. 221–238.
- Bramble T. H. and Shatz A. H. (1977) Higher order local accuracy by averaging in the finite element methods // Math. Comp., 31, p. 94–111.
- Bugeda G. (2000) Comparison between different adaptive remeshing strategies based on different global and local optimality criteria // European Congress on Computational Methods in Applied Sciences and Engineering, Barcelona, Spain, 15 p. in CD-ROM.
- Bugeda G. (2001) Definition of alternative adaptive remeshing strategies based on different global and local optimality criteria // European Conference on Computational Mechanics, Cracow, Poland, 15 p. in CD-ROM.
- Bugeda G. and Oliver J. (1991) Automatic adaptive remeshing for structural shape optimization // European Conf. on New Advances in Computat. Struct. Mech., Giens, France.
- Bugeda G. and Oliver J. (1993) A general methodology for structural shape optimization problems using automatic adaptive remeshing // Int. J. Num. Meth. Engng., 36, p. 3161–3185.
- Bugeda G. and Onate E. (1992) New adaptive techniques for structural problems // 1st European Conference on Numerical Methods in Engineering, Brussels, Belgium, p. 230–239.
- Bugeda G. and Onate E. (1994) Adaptive techniques for elliptic problems // CIVIL-COMP Ltd., Edinburg, Scotland, p. 77–85.
- Cai Z. and Starke G. (2004) Least-Squares Methods for Linear Elasticity // SIAM Journal on Numerical Analysis, 42(2), p. 826–842.

- Chatzipantelidis P. and Lazarov R. D. (2005) Error Estimates for a Finite Volume Element Method for Elliptic PDEs in Nonconvex Polygonal Domains // *SIAM Journal on Numerical Analysis*, 42(5), p. 1932–1958.
- Coorevits P. and Bellenger E. (2004) Alternative mesh optimality criteria for h-adaptive finite element method // *Finite Elements in Analysis and Design*, 40(9-10), p. 1195–1215.
- Dauge M. (1988) Elliptic boundary value problems on corner domains // Volume 1341 of *Lecture Notes in Mathematics*. Springer, Berlin, p. 259.
- Delaunay B. (1934) Sur la sphere vide // *Bulletin of the Academy of Sciences of the USSR VII, Class Sci. Mat. Nat.*, 6, p. 793–800.
- Demkowicz L., Devloo Ph. And Oden J. T. (1985) On an h-type mesh refinement strategy based on a minimization of interpolation error // *Comput. Methods Appl. Mech. Eng.*, 3, p. 67–89.
- Dempsey J. P. and Sinclair G. B (1981) On the singular behavior at the vertex of bi-material wedge // *J. Elasticity*, 11, p. 317–327.
- Desai C. S. and Abel F. J. (1972) *Introduction to the finite element method* // Van Nostrand Reinhold Company, New York, USA, 224 p.
- Dobrowolski M. and Villegas M. (2005) A Stabilized Mixed Finite Element Method for Elliptic Systems of First Order // *SIAM Journal on Numerical Analysis*, 43(3), p. 949–969.
- El-Hamalawi A. (2004) A 2D combined advancing front-Delaunay mesh generation scheme // *Finite Elements in Analysis and Design*, 40(9-10), p. 935–1281.
- Field D. A. (2000) Qualitative measures for initial meshes // *International Journal for Numerical Methods in Engineering*, 47, p. 887–906.
- Frey P. J. and George P. L. (2000) *Mesh generation; application to finite elements* // Hermes Science Publishing, Oxford, 814 p.
- Frykestig J. (1994) Advancing front mesh generation techniques with application to the finite element method // *Chalmers University of Technology. Göteborg*, 224 p.
- Frykestig J. (1994) Advancing-front mesh generation techniques with application to the finite element method // *Research report, Chalmers Univ. of Technology, Göteborg, Sweeden*, 224 p.
- Fuenmayor F. J. and Oliver J. L. (1996) Criteria to achieve nearly optimal meshes in h-adaptive finite element method // *Int. J. Num. Meth. Engng.*, 39, p. 4039–4061.
- George P. L. and Seveno E. (1994) The advancing front mesh generation method revisited // *Int. J. Numer. Methods in Eng*, 37, p. 3605–3619.
- Grisvard P. (1985) Elliptic problems in nonsmooth domains // *Pitman Publishing, England*, p. 424.
- Grisvard P. (1995) Singular behavior of elliptic problems in non Hilbertian Sobolev spaces // *J. Math. Pures Appl.* 74, p. 3–33.
- Gu H., Zong Z. and Hung K. C. (2004) A modified superconvergent patch recovery method and its application to large deformation problems // *Finite Elements in Analysis and Design*, 40(5-6), p. 665–687.
- Gupta A. K., Fang T. H. and Chen H. (1991) Computational efficiency on p- and h- versions elements // *Commun. Appl. Numer. Methods*, 7, p. 87–92.
- Hansbo P. (1989) Adaptivity and streamline diffusion procedures in the finite element method // *Ph.D. Thesis, Depth. Of Struct. Mech., Chalmers Univ. Of Tech., Goteborg, Sweeden*, p. 73.
- Hinton E. and Owen D. J. R. (1979) *An introduction to finite element computations* // Pineridge Press Limited, Swansea, UK, p. 385.

- Irons B. and Ahmad S. (1980) Techniques of finite elements // Ellis Horwood, Chichester, England, 253 p.
- Yerry M. A. and Shephard M. S. (1983) A modified-quadtrees approach to finite element mesh generation // IEEE Computer Graphics Appl., 3(1), p. 39–46.
- Ying X. (1986) A reliable root solver for automatic computation with application to stress analysis of composite plane wedge // Ph.D. Dissertation, Washington University, St. Louis, 82 p.
- Yosibash Z. and Szabo B. A. (1995) Numerical analysis of singularities in two-dimensions. Part 1: computation of eigenpairs // Int. J. Num. Meth. Engng., 38, p. 2055–2082.
- Yosibash Z. and Szabo B. A. (1995) The solution of axisymmetric problems near singular points and computation of stress intensity factors // Finite Elements Anal. Des. 119(1-2), p. 115–129.
- Jin H. and Tanner R. I. (1993) Generation of Unstructured Tetrahedral meshes by Advancing Front technique // Int. J. Numer. Methods Eng., 33, p. 1805–1823.
- Joe B. (1991) Construction of three-dimensional Delaunay triangulations using local transformations // Comput. Aided Geom. Design, 8, p. 123–142.
- Kačianauskas R., Stupak E. ir Kienzel R. (2002) Macroscopic Finite Element Simulation of Brittle-to-Ductile Failure of SENB Specimen // Mechanika, 1(33), p. 11–16.
- Kačianauskas R., Markauskas D. (2004) Moving Locally Predefined Remeshing for Deep Cone Penetration FE Analysis // Informatica, 15(4), p. 489–514.
- Kang E. ir Haghighi K. (1995) Intelligent finite element generation // Engineering with Computers, 11, p. 70–82.
- Kela A., Perucchio R. and Voelcher H. (1986) Toward automatics finite element analysis // Comp. Mech. Eng., p. 7–71.
- Knyazev A. V. and Osborn J. E. (2006) New A Priori FEM Error Estimates for Eigenvalues // SIAM Journal on Numerical Analysis, 43(6), p. 2647–2667.
- Kondratiev V. A. (1967) Boundary value problem for elliptic equations in domains with conical or angular points // Trans.Moscow Math. Soc. 16, p. 227–313.
- Kondratiev V. A. (1970) The smoothness of a solution of Dirichlet's problem for second order elliptic equations in a region with piecewise smooth boundary // Differ. Equation, 6, p. 1392–1401.
- Kufner A. and Sändig A. M. (1987) Some Applications of Weighted Sobolev Spaces // Teubner, Leipzig, 268 p.
- Kvamsdal T. and Okstad K. M. (1998) Error estimation based on superconvergent patch recovery using statically admissible stress fields // Int. J. Numer. Methods Eng., 42, p. 443–472.
- Lee C. K and Lo S. H.(1993) A new scheme for the generation of a graded quadrilateral mesh // Computers & Structures, 52(2), p. 847–857.
- Lee C. K. and Hoobs R. E.(1999). Automatic adaptive finite element mesh generation over arbitrary two-dimensional domain using advancing front technique // Computers & Structures, 71, p. 9–34.
- Leguillon D. and Shanches-Palencia E. (1987) Computation of singular solution in elliptic problems and elasticity // Wiley, New York, 200 p.
- Li L. Y. , Bettess P., Bull J. W., Bond T. and Applegarth I. (1995) Theoretical formulations for adaptive finite element computations // Comm. Numer. Methods Engrg., 11, p. 857-868.
- Li L. Y. and Bettess P. (1995) Notes on mesh optimality criteria in adaptive finite element computations // Communications in Numerical Methods in Engineering, 11, p. 911–915.

- Linardakis L. and Chrisochoides N. (2006) Delaunay Decoupling Method for Parallel Guaranteed Quality Planar Mesh Refinement // *SIAM Journal on Scientific Computing*, 27(4), p. 1394–1423.
- Lo S. H. (1985). A new mesh generation scheme for arbitrary planar domains // *Int. j. numer. Methods eng.*, 21, p. 1403–1426.
- Löhner R. (1988) Some useful Data Structures for the Generation of Unstructured Grids // *Commun. Numer. Methods eng.*, 4, p. 123–135.
- Löhner R. (1996) Extensions and improvements of the advancing front grid generation technique // *Communications in Numerical Methods in Engineering*, 12, p. 683–702.
- Löhner R. (1996) Progress in grid generation via the advancing front technique // *Engineering with Computers*, 12, p. 186–210.
- Löhner R. and P. Parikh (1988) Three-dimensional grid generation by the advancing front method // *Int.j. numer. Methods fluids*, 8, p. 1135–1149.
- Melenk J. M. (2005) hp-Interpolation of Nonsmooth Functions and an Application to hp-A posteriori Error Estimation // *SIAM Journal on Numerical Analysis*, 43(1), p. 127–155.
- Moller P. W. (1994) Procedures in adaptive finite element analysis // Ph.D. Thesis, Depth. Of Struct. Mech., Chalmers Univ. Of Tech., Goteborg, Sweeden, 76 p.
- Morin P., Nochetto R. H. and Siebert K. G. (2002) Convergence of adaptive finite element methods // *SIAM Review*, 44(4), p. 631–658.
- Oden J. T. and Brauchli H. (1971) On the calculation of consistent stress distributions in finite element approximations // *Int. J. Numer. Methods Eng.*, 3, p. 317–325.
- Oden J. T., Demkowicz L., Rachowicz W. and Westermann T. A. (1989) Toward a universal h-p adaptive finite element strategy, Part 2. A posteriori error estimates // *Comput. Methods Appl. Mech. Eng.*, 77, p. 113–180.
- Oden J. T., Demkowicz L., Strouboulis T. and Devloo P. (1986) Adaptive methods for problems in solid and fluid mechanics // In: I. Bahuška, O.C. Zienkiewicz, J. Gago and E.R. de A. Oliveira, eds. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*. New York: Wiley, p. 249–280.
- Okstad K. M. and Mathisen K. M. (1994). Towards automatic geometrically non-linear shell analysis. Part 1: implementation of an h-adaptive mesh refinement procedure // *Int. J. Numer. Methods Eng.*, 37, p. 2657–2678.
- Onate E. and Bugeda G. (1993) A study of mesh optimality criteria in adaptive finite element analysis // *Eng. Comp.*, 10, p. 307–321.
- Owen S. J. (1992) An implementation of natural neighbor interpolation in three dimensions // Master's Thesis, Brigham Young University, 72 p.
- Owen S. J. and Saigal S. (2000) Surface mesh sizing control // *International Journal for Numerical Methods in Engineering*, 47, p. 497–511.
- Papadakis P. (1988) Computational aspects of the determination of the stress intensity factors for two-dimensional elasticity // Ph.D. Dissertation, University of Maryland, 84 p.
- Peraire J., Peiro J. and Morgan K. (1992) Adaptive remeshing for three-dimensional compressible flow computations // *Jour. of Comput. Phys.*, 103, p. 269–285.
- Peraire J., Peiro J., Formaggia L., Morgan K. and Zienkiewicz O. C. (1988) Finite element Euler computations in three dimensions // *International Journal for Numerical Methods in Engineering*, 26, p. 2135–2159.

- Peraire J., Vahdati M., Morgan K. and Zienkiewicz O. C. (1987) Adaptive remeshing for compressible flow computations // *Jour. Of Comput. Phys.*, 72, p. 449–466.
- Picasso M. (2003) An Anisotropic Error Indicator Based on Zienkiewicz--Zhu ErrorEstimator: Application to Elliptic and Parabolic Problems // *SIAM Journal on Scientific Computing*, 24(4), p. 1328–1355.
- Preparata F. P. and Shamos M. I. (1985) Computational geometry, an introduction // Springer-Verlag. 598 p.
- Qian Z. Q. (2001) On the evaluation of wedge corner stress intensity factors of bi-material joints with surface tractions // *Computers & Structures*, 79, p. 53–64.
- Samet H. (1984) The Quadtree and Related Hierarchical Data structures // *ACM Computing Surveys*, 16(2), p. 187–260.
- Selna L. and Hakam A. (2000) Least square methods in structural analysis // 5th Int. Conf. on Computational Structures Technology, Leuven, Belgium, 6-8 September 2000, p. 43–57.
- Song Ch. and Wolf J. P. (2002) Semi-analytical representation of stress singularities as occurring in cracks in anisotropic multi-materials with the scaled boundary finite-element method // *Computers & Structures*, 80, p. 183–197.
- Steven O. J. and Saigal S. (2000) Surface mesh sizing control // *International Journal for Numerical Methods in Engineering*, 47, p. 497–511.
- Stevenson R. (2005) An Optimal Adaptive Finite Element Method // *SIAM Journal on Numerical Analysis*, 42(5), p. 2188–2217.
- Stupak E. (2004) Dvimačių struktūrų irimo modeliavimas naudojant prisitaikančiuosius baigtinių elementų tinklus // *Daktaro disertacija, Fundamentinių mokslų fakultetas, Vilniaus Gedimino Technikos universitetas, Vilnius, Lietuva*, 87 p.
- Stupak E. ir Baušys R. (2000) Nestruktūrinių BE tinklų generavimas sudėtingiems dvimačiams objektams // *Statyba*, VI (1), p. 17–24.
- Stupak E. ir Kačianauskas R. (2000) Adaptyviųjų baigtinių elementų tinklų taikymas įtempimų būviui nustatyti ties koncentratoriais // *Mechanika*, Nr. 4 (24), p. 31–35.
- Szabo B. A. (1990) The p-and h-p Versions of the finite element method in solid mechanics // *Comput. Methods Appl. Mech. Eng.*, 80, p. 185–195.
- Szabo B. A. and Babuška I. (1991) Finite element analysis // Willey, New York, 384 p.
- Szabo B. A. and Yosibash Z. (1996) Numerical analysis of singularities in two dimensions. Part 2: computation of generalized flux/stress intensity factors // *Int. J. Num. Meth. Engng.*, 39, p. 409–434.
- Topping B. H. V, Muylle J., Ivanyi P., Putanowicz R. and Cheng B. (2004) Finite element mesh generation // Saxe-Coburg Publications, 296 p.
- Turner M. J., Clough R. W., Martin H. and Topp L. J. (1956) Stiffness and Deflection Analysis of Complex Structures // *J. Aero. Sci.*, 13, p. 805–823.
- Wiberg N. E. and Abdulwahab F. (1993) A posteriori error estimation based on superconvergent derivatives and equilibrium // *Int. J. Numer. Methods Eng.*, 36, p. 2703–2724.
- Wiberg N. E., Abdulwahab F. and Žiukas S. (1994) Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions // *Int. J. Numer. Methods Eng.*, 37, p. 3417–3440.

- Zeng L. F. (1991) On adaptive finite element procedures for static and dynamic problems // Ph.D. Thesis, Depth. Of Struct. Mech., Chalmers Univ. Of Tech., Goteborg, Sweeden, 127 p.
- Zeng L. F. and Wiberg N. E. (1992) Adaptive h-p procedures for high accuracy finite element analysis of two dimensional linear elastic problems // Comput. Struct., 42, p. 869–886.
- Zhu J. Z. (1987) Error estimation, adaptivity and multigrid technique in finite element method // Ph.D. Thesis, Depth. Of Civil. Eng., Wales Univ., Swansea, UK, 87 p.
- Zhu J. Z. and Zienkiewicz O. C. (1988) Adaptive techniques in the finite element method // Comm. Appl. Numer. Methods, 4, p. 197–204.
- Zienkiewicz C. O. and Taylor R. L. (2000) Adaptive finite element refinement // The Finite Element Method, 1, p. 401–428.
- Zienkiewicz C. O. and Taylor R. L. (1967) The finite element method in structural and continuum mechanics // McGraw Hill, London, 1st edition, 619 p.
- Zienkiewicz O. C. and Zhu J. Z. (1992) The superconvergent patch recovery and a posteori error estimates. Part 1. The recovery technique // Int. J. Numer. Methods Eng., 33, p. 1331–1364.
- Zienkiewicz O. C. and Zhu J. Z. (1992) The superconvergent patch recovery and a posteori error estimates. Part 2. Error estimates and adaptivity // Int. J. Numer. Methods Eng., 33, p. 1365–1382.
- Zienkiewicz O. C. and Zhu J. Z. (1987) A simple error estimator and adaptive procedure for practical engineering analysis // Int. J. Num. Meth. Engng., 24, p. 337–357.
- Zienkiewicz O. C., Zhu J. Z. and Gong N. G. (1989) Effective and practical h-p version adaptive analysis procedures for the finite element method // Int. J. Numer. Methods Eng., 28, p. 879–891.
- Žarnovskij V. (2005) Įtempimų būvio erdviškumo įtakos plokštelių su įpjova įtempimų intensyvumo koeficientui skaitinis tyrimas // Daktaro disertacija, Fundamentinių mokslų fakultetas, Vilniaus Gedimino Technikos universitetas, Vilnius, Lietuva, 94 p..
- Žarnovskij V. ir Kačianauskas R. (2003) The finite element investigation of the thickness influence on 3D stress and strain fields in the SENB specimens // Mechanika, 4(42), p. 5–10.

Linios Vasiliauskienės publikacijų sąrašas:

- Vasiliauskienė L. ir Baušys R. (2002) Intelligent Initial Finite Element Mesh Generation for Solutions of 2D problems // Informatica, 13(2), p. 239–250 (ISI).
- Vasiliauskienė L. ir Baušys R. (2003). Adaptive Finite Element Analysis Based on Different Mesh Optimality Criteria // Konferencijos „Mechanika–2003“ pranešimų leidinys, p. 133–138.
- Vasiliauskienė L. ir Baušys R. (2003) Intelligent Adaptive Mesh refinement procedure based on global and local mesh optimality criteria // Mechanika, 3(41), p. 11–17.
- Vasiliauskienė L. ir Baušys R. (2006) Mesh Optimization Strategies for Solution of Engineering Problems // Solid State Phenomena, 113, p. 157–162 (ISI).