

Запорізький національний університет
Міністерства освіти і науки України

Харківський національний університет радіоелектроніки
Міністерства освіти і науки України

Кваліфікаційна наукова праця
на правах рукопису

ЧОПОРОВ СЕРГІЙ ВІКТОРОВИЧ

УДК: 004.94:519.876.2:621(043.5)

ДИСЕРТАЦІЯ
МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА АНАЛІЗ ФОРМ ОБ'ЄКТІВ
У САПР МАШИНОБУДУВАННЯ

05.13.12 – системи автоматизації проектувальних робіт

Подається на здобуття наукового ступеня доктора технічних наук

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Підпис С. В. Чопоров

Науковий консультант

Гоменюк Сергій Іванович, доктор технічних наук, професор

Цей примірник дисертаційної роботи ідентичний за змістом з іншими, поданими до спеціалізованої вченої ради Д 64.052.02

Вчений секретар спеціалізованої вченої ради Д 64.052.02

Підпис
Печатка

Л. В. Колесник

АНОТАЦІЯ

Чопоров Сергій Вікторович. Математичне моделювання та аналіз форм об'єктів у САПР машинобудування. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.12 «Системи автоматизації проектувальних робіт». – Запорізький національний університет, Міністерство освіти і науки України; Харківський національний університет радіоелектроніки, Міністерство освіти і науки України; Запоріжжя, 2019.

Дисертацію присвячено вирішенню актуальної науково-технічної проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного на дослідження їх стану. У загальному випадку для моделювання стану об'єктів машинобудування використовують системи диференціальних рівнянь з частинними похідними, для розв'язку яких необхідно використовувати чисельні методи. Тоді виникає необхідність переходу від неперервних моделей об'єктів до дискретних. У результаті вирішення зазначеної проблеми здійснюється шляхом:

- розробки математичних моделей форм об'єктів машинобудування у САПР з використанням функціонального підходу;
- побудови дискретних моделей форм об'єктів машинобудування, поданих функціонально;
- виконання чисельного аналізу стану об'єкту на базі дискретної моделі його форми;
- адаптація дискретної моделі форми об'єкту для забезпечення потрібної точності чисельного аналізу.

Метою дисертаційної роботи є розробка ефективних методів математичного моделювання форм об'єктів у САПР машинобудування.

Для досягнення поставленої мети у дисертаційній роботі необхідно розв'язати такі *основні задачі*:

- виконати огляд і аналіз сучасного стану проблеми;
- розробити гібридний підхід до подання форм об'єктів машинобудування з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання);
- розробити проєкційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально;
- розробити метод згладжування координат вузлів дискретних моделей меж об'єктів, форма яких задана функціонально;
- модифікувати метод локальних перебудов елементів дискретних моделей на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об'єктів;
- розробити метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально;
- розробити метод згладжування координат внутрішніх вузлів дискретних моделей форм об'єктів;
- розробити метод генерації адаптивних дискретних моделей форм об'єктів (adaptive meshing), заданих функціонально, для забезпечення скінченно-елементного аналізу поведінки об'єкта із заданою точністю;
- розробити методи генерації дискретних моделей форм об'єктів і скінченно-елементного аналізу з використанням технологій паралельних обчислень;
- розробити проблемно-орієнтовану мову для опису об'єктів машинобудування складної форми з використанням гібридного підходу, а також задач побудови дискретних моделей та скінченно-елементного аналізу;
- розробити САПР для апробації розроблених методів при дослідженні поведінки об'єктів машинобудування.

Об'єкт дослідження – об'єкти машинобудування.

Предмет дослідження – математичні моделі форм об'єктів машинобудування.

Дослідження базується на комплексному використанні методів аналітичної геометрії, математичного аналізу та математичного моделювання: для опису форм об'єктів використовується теорія R-функцій; для побудови дискретних моделей

форм об'єктів – методи аналітичної геометрії, двійкового пошуку; для мінімізації функціоналів при пошуку оптимальних координат вузлів – методи покоординатного спуску і спряжених градієнтів; для побудови адаптивних дискретних моделей форм об'єктів – метод скінченних елементів; для апробації розроблених методів у САПР – об'єктно-орієнтований підхід.

У результаті виконання дисертаційного дослідження розроблені математичні моделі, наукові методи та програмне забезпечення для математичного моделювання та аналізу форм об'єктів у САПР машинобудування. *Водночас отримані такі нові наукові результати:*

- вперше запропоновано гібридний підхід до подання форм об'єктів з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання), що дозволяє зменшити час побудови математичних моделей;

- вперше запропоновано проєкційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально, що дозволяє будувати дво- та тривимірні дискретні моделі;

- вперше запропоновано метод згладжування координат вузлів дискретних моделей меж об'єктів, форма яких задана функціонально, у результаті чого можна підвищити точність моделювання в околах зламів їх поверхонь;

- отримав подальшого розвитку метод локальних перебудов елементів дискретних моделей у частині його узагальнення на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об'єктів, що дозволяє підвищити точність скінченно-елементного аналізу;

- вперше запропоновано метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально, який дозволяє будувати дво- і тривимірні моделі як з використанням симплексів (трикутників або тетраєдрів), так і з використанням топологічних кубів (чотирикутників або шестигранників);

- вперше запропоновано метод згладжування координат внутрішніх вузлів дискретних моделей форм об'єктів, відмінність якого від інших методів полягає в усуненні вироджених елементів та підвищенні точності скінченно-елементного аналізу;

– вперше запропоновано метод генерації адаптивних дискретних моделей форм об'єктів (adaptive meshing), заданих функціонально, який дозволяє будувати дво- і тривимірні дискретні моделі для виконання скінченно-елементного аналізу із потрібною точністю;

– отримали подальшого розвитку методи генерації дискретних моделей форм об'єктів і скінченно-елементного аналізу у частині розробки паралельних алгоритмів, що дозволяє зменшити витрати часу на моделювання;

– вперше розроблена проблемно-орієнтована мова, яка дозволяє описувати об'єкти машинобудування складної форми з використанням гібридного підходу, а також задачі побудови дискретних моделей та скінченно-елементного аналізу.

Наукові результати, отримані у результаті виконання дисертаційного дослідження, у сукупності є *вирішенням науково-технічної проблеми* підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного на дослідження їх стану. Очікувані науково-технічні ефекти:

– підвищення якості візуального подання моделей об'єктів машинобудування;

– зниження витрат на натурні випробування об'єктів, що проектується, за рахунок підвищення точності моделювання при дослідженні їх стану з використанням чисельних методів на базі дискретних моделей.

Наукові результати, отримані в дисертаційній роботі, відповідають формулі спеціальності 05.13.12 та напрямкам досліджень № 2, 4, 6 і 7 згідно з паспортом спеціальності, що свідчить про створення математичного, лінгвістичного та програмного забезпечення систем автоматизації проектування машинобудування.

Розроблені у дисертаційній роботі методи можуть використовувати конструкторські бюро і науково-дослідні організації для розробки САПР. Розв'язання задач дослідження дозволило розробити САПР для чисельного аналізу стану об'єктів машинобудування на базі адаптивних дискретних моделей.

Результати дисертаційної роботи використані у навчальному процесі кафедри програмної інженерії Запорізького національного університету для розробки

лабораторних робіт у курсах «Паралельні та розподілені обчислення» і «Програмне забезпечення наукових досліджень».

Результати дисертаційної роботи використані при виконанні держбюджетних тем з номерами держреєстрації: 0112U003061, 0113U000804, 0115U000761, 0117U007204. Також результати роботи використані у дослідженнях за Грантом Президента України для молодих учених на 2013 рік (№ держреєстрації: 0113U006252). Результати роботи також використані у науково-виробничій діяльності державного підприємства «Конструкторське бюро «Південне» ім. М. К. Янгеля» при виконанні п'яти господарчих договорів.

Результати дисертаційної роботи опубліковано в 55 роботах. З них 3 статті опубліковано у періодичних виданнях, які реферуються та включено до наукометричної бази Scopus. Опубліковано 3 монографії. У наукових фахових виданнях, що включено до Переліку МОН України з технічних наук, опубліковано 19 робіт, з них 2 статті опубліковано у журналах, які також реферуються та включено до наукометричної бази Web of Science. У наукових фахових виданнях, що включено до Переліку МОН України з фізико-математичних наук, опубліковано 10 робіт, які додатково висвітлюють наукові результати. Отримано 6 свідоцтв про реєстрацію авторського права на комп'ютерні програми. У збірниках матеріалів конференцій та тез доповідей опубліковано 15 робіт, з них одна робота опублікована у збірнику, який включено до наукометричної бази Scopus.

Ключові слова: САПР, математична модель, форма об'єкта, дискретна модель, адаптивна дискретна модель, проблемно-орієнтована мова, скінченний елемент.

Список публікацій здобувача

1. Построение дискретных моделей геометрических объектов: монография / В. М. Колодяжный, Д. А. Лисин, А. А. Лисняк, С. В. Чопоров. Харьков: ХНАДУ, 2013. 268 с.

2. Функциональный поход к геометрическому моделированию технических систем: монография / С. В. Чопоров, С. Н. Гребенюк, С. И. Гоменюк [и др.] Запорожье: Запорожский национальный университет, 2016. 176 с.

3. Чопоров С. В. Дискретные модели технических объектов: монография. Херсон: Видавничий дім «Гельветика», 2018. 324 с.
4. Чопоров С. В. Построение неравномерных дискретных сеток для функциональных математических моделей на базе теории R-функций // Радиоэлектроника, информатика, управление. 2011. № 2. С. 70–75.
5. Гоменюк С. И., Чопоров С. В., Лисняк А. А. FORTU-FEM – автоматизированная система для исследования сложных механических процессов // Вестник Херсонского национального технического университета. 2012. № 2(45). С. 101–105.
6. Математическое моделирование некоторых крепежных соединений на базе теории R-функций / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, Е. В. Панасенко // Радиоэлектроника, информатика, управление. 2012. №2(27). С. 117–122.
7. Чопоров С. В., Лисняк А. А., Гоменюк С. И. Подход к определению областей сгущения сетки конечных элементов на базе теории R-функций // Методи розв’язування прикладних задач механіки деформівного твердого тіла. Дніпропетровськ: Ліра. 2012. Вип. 13. С. 413–419.
8. Чопоров С. В., Гоменюк С. И., Лисняк А. А. Сравнительный анализ треугольных и четырехугольных конечных элементов // Вестник Херсонского национального технического университета. 2013. № 2(47). С. 382–386.
9. Чопоров С. В. Использование технологий параллельных вычислений в методе конечных элементов // Радиоэлектроника, информатика, управление. 2013. № 2(29). С. 88–94.
10. Кудин А. В., Чопоров С. В. Компьютерное моделирование изгиба круглой трехслойной пластины с использованием аналитического и численного подходов // Радиоэлектроника, информатика, управление. 2014. № 1(30). С. 75–81.
11. Чопоров С. В. Проблемно-орієнтована мова для моделювання геометричних об’єктів при функціональному підході // Вестник Херсонского национального технического университета. 2014. № 3(50). С. 532–539.
12. Гоменюк С. И., Чопоров С. В., Аль-Омари М. А. В. Функциональное сглаживание сеток конечных элементов // Технология приборостроения. 2015. № 1. С. 10–14.

13. Чопоров С. В., Гоменюк С. И. Параллельный способ построения сеток треугольных элементов при функциональном представлении // Вестник Херсонского национального технического университета. 2015. № 3(54). С. 511–517.

14. Чопоров С. В. Метод фоновой сетки для триангуляции двумерных областей при функциональном подходе. Радиоэлектроника, информатика, управление. 2015. № 4(35). С. 31–38. DOI: 10.15588/1607-3274-2015-4-5 [Включено до наукометричної бази Web of Science].

15. Формализация моделей многослойных тонкостенных конструкций в САПР / С. В. Чопоров, С. И. Гоменюк, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омари, Х. Х. Алатемнех // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Серія: Нові рішення в сучасних технологіях. 2015. № 62(1171). С. 99–106.

16. Триангуляция поверхностей при гибридном параметрическом и неявном представлении / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, М. А. В. Аль-Омари, Х. Х. Алатамнех // Вестник Херсонского национального технического университета. 2016. № 3(58). С. 542–549.

17. Analytical and Numerical Modelling of the Axisymmetric Bending of Circular Sandwich Plates with the Nonlinear Elastic Core Material / A. Kudin, S. Choporov, Yu. Tamurov, M. A. V. Al Omari // Journal of Solid Mechanics. 2016. Vol. 8, Iss. 3. P. 590–601. [Включено до наукометричної бази Scopus].

18. Чопоров С. В., Аль-Атемнех Б. Г. М., Гоменюк С. И. Підхід до геометричного моделювання в паралельних обчислювальних системах зі спільною пам'яттю // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 27. С. 164–172.

19. Чопоров С. В. Гибридное представление сплошных тел с использованием неявных и параметрических функций // Радиоэлектроника, информатика, управление. 2017. № 3. С. 60–70. DOI: 10.15588/1607-3274-2017-3-7 [Включено до наукометричної бази Web of Science].

20. Чопоров С. В. Сглаживание сеток четырехугольных элементов с использованием локальной минимизации функционала // Вестник Херсонского национального технического университета. 2017. Т. 2, № 3(62). С. 234–239.

21. Kudin A. V., Choporov S. V., Gomenyuk S. I. Axisymmetric bending of circular and annular sandwich plates with nonlinear elastic core material // Mathematical models and computer simulations. 2017. Vol. 9, No 5. P. 601–612. DOI: doi.org/10.1134/S2070048217050076 [Включено до наукометричної бази Scopus].

22. Математическое обеспечение анализа прочности силовых элементов ракетно-космической техники / Д. В. Акимов, В. З. Грищак, И. Ф. Ларионов, С. И. Гоменюк, Д. В. Клименко, С. В. Чопоров, С. Н. Гребенюк // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 26. С. 5–17.

23. Чопоров С. В. Представление вычислительных сеток в системах инженерного анализа // Вісник Херсонського національного технічного університету. 2018. Т. 1, № 3(66). С. 372–379

24. Choporov S., Homenyuk S, Grebenyuk S. Optimized Smoothing of Discrete Models of the Implicitly Defined Geometrical Objects' Surfaces // Eastern-European Journal of Enterprise Technologies. 2018. Vol. 3/4 (93). P. 52–60. DOI: 10.15587/1729-4061.2018.130787 [Включено до наукометричної бази Scopus].

25. Мухин В. В., Чопоров С. В. Визуализация дифференциальных характеристик дискретных данных // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 2. С. 60–65.

26. Чопоров С. В. Алгебраические способы учета перемещения узла сетки конечных элементов // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 1. С. 117–121.

27. Кудин А. В., Тамуров Ю. Н., Чопоров С. В. Аналитический и численный анализ изгиба круглой трехслойной пластины под действием локальных нагрузок // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 67–81.

28. Модель напружено-деформованого стану зубчатого колеса / Х. Х. Алатемнех, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омарі, С. В. Чопоров // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 2. С. 173–186.

29. Снежкова Л. С., Чопоров С. В. Методы и подходы к моделированию геометрических объектов в контактных задачах // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 171–195.

30. Математическое моделирование и исследование прочности силовых элементов конструкций космических летательных аппаратов / Д. В. Акимов, В. З. Грищак, С. И. Гоменюк, С. Н. Гребенюк, А. А. Лисняк, С. В. Чопоров, И. Ф. Ларионов, Д. В. Клименко, В. Н. Сиренко // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 6–13.

31. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 239–248.

32. Методы построения дискретных моделей: структурированные и блочно-структурированные сетки / С. В. Чопоров, С. И. Гоменюк, Х. Х. Алатамнех, К. С. Оспищев // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 1. С. 272–284.

33. Методы построения дискретных моделей: неструктурированные сетки / С. В. Чопоров, А. А. Лисняк, Ю. А. Борисовская, О. С. Козлова, Л. С. Снежкова // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 2. С. 237–250.

34. Чопоров С. В. Автоматизация анализа температурной устойчивости пластин с использованием метода конечных элементов // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2017. № 2. С. 304–316.

35. Комп'ютерна програма «Система автоматизованого проектування FORTU-FEM»: Свідоцтво про реєстрацію авторського права на твір № 43043 Україна / С. І. Гоменюк, В. З. Грищак, С. М. Гребенюк, С. В. Чопоров, А. О. Лісняк; дата реєстрації 02.04.2012.

36. Комп'ютерна програма «Автоматизована система "Stability"»: Свідоцтво про реєстрацію авторського права на твір № 53201 Україна / С. І. Гоменюк,

С. М. Гребенюк, В. З. Грищак, Ю. М. Тамуров, С.В. Чопоров; дата реєстрації 20.01.2014.

37. Комп'ютерна програма «Система автоматизованого проектування «qMesher»: Свідоцтво про реєстрацію авторського права на твір № 53829 Україна / С.В. Чопоров, С.І. Гоменюк, С.М. Гребенюк, А.О. Лісняк; дата реєстрації 24.02.2014.

38. Комп'ютерна програма «Бібліотека консольної індикації «qzIO» («qzIO»): Свідоцтво про реєстрацію авторського права на твір № 70462 Україна / С. В. Чопоров, В. З. Грищак, С. М. Гребенюк, С. І. Полюга; дата реєстрації 15.02.2017.

39. Комп'ютерна програма «Бібліотека паралельної обробки розріджених матриць qzMatrix» («qzMatrix»): Свідоцтво про реєстрацію авторського права на твір № 70465 Україна / С. В. Чопоров, С. І. Гоменюк, А. О. Лісняк; дата реєстрації 15.02.2017.

40. Комп'ютерна програма «Паралельний інтерпретатор мови специфікацій математичних моделей геометричних об'єктів»: Свідоцтво про реєстрацію авторського права на твір № 82295 Україна / С. В. Чопоров, А. О. Лісняк, О. В. Кудін, М. С. Ігнатченко; дата реєстрації 17.10.2018.

41. Глушко К. А., Мухін В. В., Чопоров С. В. Автоматизована підсистема побудови трьох характеристик поверхні функції двох змінних // Актуальні проблеми математики та інформатики: збірка тез доповідей III Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 26–27 квітня 2012 р.). Запоріжжя: ЗНУ, 2012. – С. 69.

42. Лісняк А. А., Чопоров С. В. Способы определения областей сгущения вычислительной сетки для моделей на базе теории R-функций // Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 5–8 листопада 2012 р.). Харків: Ротапринт Інституту проблем машинобудування ім. А. М. Підгорного НАН України, 2012. – С. 38.

43. Кліценко Г. О., Чопоров С. В. Розробка способу врахування переміщення вузлів скінченно-елементної сітки // Актуальні проблеми математики та

інформатики : збірка тез доповідей IV Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 25–26 квітня 2013 р.). Запоріжжя: ЗНУ, 2013. 135 с.

44. Choporov S. Parallel Computing Technologies in the Finite Element Method // High Performance Computing: Proceedings of Third International Conference HPC-UA 2013 (Kyiv, Ukraine, 7–11 October, 2013). Kyiv: National Technical University of Ukraine "Kyiv Polytechnic Institute", 2013. P. 85–91.

45. Кліценко Г. О., Чопоров С. В. Розробка способу врахування тріщин конструкцій в сітці на базі чотирикутників // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. С. 147.

46. Снежкова Л. С., Чопоров С. В. Дискретні моделі в контактних задачах на базі теорії R-функцій // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. 147 с.

47. Choporov S. V., Lisnyak A. O., Akimov D. V. Discrete models generation for layered shells of a spacecraft // Aviation in the XXI-st Century. Safety in Aviation and Space Technologies: Proceedings The Sixth World Congress (Kyiv, Ukraine, 23–25 September, 2014). Ukraine, Kyiv: NAU, 2014. V. 1. P. 1.14.45–1.14.48.

48. Оспищев К. С., Чопоров С. В. Моделирование напряженно-деформированного состояния эластомерных конструкций на базе современных информационных технологий // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 89.

49. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 90–91.

50. Чопоров С. В. Гибридный подход к представлению геометрических объектов // Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 21–24 листопада 2016 р.). Харків: Інститут проблем машинобудування ім. А. Н. Підгорного НАН України, 2016. С. 36.

51. Модель напряженно-деформированного состояния тонкостенных конструкций ракетной техники с учетом температурных деформаций / В. З. Гришак, С. И. Гоменюк, С. Н. Гребенюк, С. В. Чопоров, Д. В. Акимов // АВІА-2017: матеріали XIII Міжнародної науково-технічної конференції (м. Київ, 19–21 квітня 2017 р.). Київ: Національний авіаційний університет, 2017. С. 7.41–7.45.

52. Чопоров С. В., Акимов Д. В. Использование элементов связи при функциональном подходе к заданию нагрузок в моделях оболочечных конструкций // Інформатика, управління, штучний інтелект: матеріали IV Міжнародної конференції студентів, магістрів та аспірантів (м. Харків, 21–23 листопада 2017 р). Харків: НТУ «ХП», 2017. С. 105.

53. Buckling Analysis of Circular Sandwich Plates with the Nonlinear Elastic Core Material / O. Kudin, S. Choporov, S. Gomenyuk, V. Gristchak // Mechanika 2018: Proceedings of 23rd international conference at Kaunas University of Technology (Kaunas, Lithuania, 18 May, 2018). Kaunas University of Technology, 2018. P. 92–100.

54. Чопоров С. В. Проекційний підхід до генерації поверхневих сіток при функціональному поданні тіл // Сучасні проблеми механіки та математики: збірник наукових праць (м. Львів, 22–25 травня 2018 р.) / ред. А. М. Самойленка та Р. М. Кушніра. 2018. Т. 1. С. 139–140. URL: www.iapmm.lviv.ua/mpmm2018 (дата звернення: 18.09.2018).

55. Design Patterns for Object-Oriented Scientific Software / S. Choporov, S. Gomenyuk, O. Kudin, A. Lisnyak // CEUR Workshop Proceedings. 2018. Vol. 2105. P. 441–444. [Включено до наукометричної бази Scopus].

ABSTRACT

Choporov Serhii Viktorovych. Mathematical Modeling and Analysis of Objects' Geometric Shapes for Computer-Aided Design in Mechanical Engineering. A qualifying scientific work on the rights of a manuscript.

A dissertation for the degree of a doctor of technical sciences in the specialty 05.13.12 «Computer-Aided Design System». – Zaporizhzhia National University, Ministry

of Education and Science of Ukraine; Kharkiv National University of Radioelectronics, Ministry of Education and Science of Ukraine; Zaporizhzhia, 2019.

The dissertation is devoted to solving the urgent scientific and technical problem of increasing efficiency of mechanical engineering objects' design by improving an accuracy of mathematical modelling and reducing time of the simulation. In mechanical engineering, a lot of physical models is based on partial differential equations. The finite element method is one the most popular numerical methods for solution of partial differential equations. This method requires the replacement of a continuous domain by a discrete model (meshing). Hence, to solve the problem following steps must be completed:

- description of a domain using some representation scheme;
- generation of a discrete model of the domain;
- performing a numerical simulation of an object's state using the discrete model;
- generation of an adaptive discrete model, which improves an analysis accuracy.

The objective of the dissertation is the development of efficient methods for solids' geometric shapes simulation in CAD of machinery.

To achieve the objective of the dissertation, the following tasks must be accomplished:

- to make a review of the problem development;
- to develop a hybrid function representation of geometric shapes;
- to develop a projection method of boundary meshing for function represented objects;
- to develop a method for boundary meshes smoothing of function represented domains;
- to improve the local refinement method for boundary meshes;
- to develop a method for meshing of two- and three-dimensional function represented objects into solid elements;
- to develop a method of inner node's smoothing;
- to develop a method for adaptive meshing using solid elements for function represented objects;

- to develop parallel meshing and finite element analysis techniques;
- to develop a domain-specific language for formalization of function represented geometric shapes and finite elements analysis problems;
- to implement developed scientific methods in CAD software.

The object of the research is mechanical engineering objects.

The subject of the research is mathematical models of geometric shapes of mechanical engineering objects.

This research uses methods of analytical geometry, mathematical analysis and mathematical modelling: the hybrid function representation of geometric shapes uses the theory of R-functions; the meshing methods employ algorithms of analytical geometry and the binary search method; the methods for nodes smoothing use the coordinate descent and conjugate gradient methods; the adaptive remeshing method employs the finite element analysis; the CAD software uses the object-oriented approach.

In the process of the research, CAD software for mechanical engineering objects' simulation has been developed. Finally, the following new scientific results have been obtained:

- the hybrid function representation of geometric shapes has been developed; this representation allows to combine implicitly and boundary defined shapes in one model of a domain;
- the projection method of boundary meshing for function represented objects has been developed;
- the method for boundary meshes smoothing has been developed;
- the local refinement method for boundary meshes has been improved; this method unified for both triangular and quadrilateral surface meshes;
- the method for meshing of two- and three-dimensional function represented objects into solid elements has been developed;
- the method of inner node's smoothing has been developed; this method employs minimization of an area functional (in 2D) or a volume functional (in 3D); this method allows to remove degenerated elements and to improve a mesh quality;

- the method for adaptive meshing using solid elements for function represented objects has been developed; this method uses the finite element method and error estimation to indicate elements for subdivision;
- the parallel meshing and finite element analysis techniques have been developed; these techniques allow to improve performance of the simulation;
- the domain-specific language for formalization of function represented geometric shapes and finite elements analysis problems has been developed.

Scientific results obtained in the dissertation is the solution of the scientific and technical problem of increasing efficiency of mechanical engineering objects' design by improving an accuracy of mathematical modelling and reducing time of the simulation. The following scientific and technical effects are possible:

- improving of the mechanical engineering objects visualization;
- cost reduction replacing physical experiments by numerical simulations based on discrete models with improved accuracy.

The methods obtained in the dissertation could be employed by design departments and scientific organizations for the CAD software development.

Results of the dissertation have been used in the educational process of the software engineering department of Zaporizhzhia National University in the following courses: «Parallel and Distributed Computing» and «Software for Scientific Researches».

Results of the dissertation have been used in the researches, which are funded by Ministry of Education and Science of Ukraine and by the Grant of the President of Ukraine for young scientists. Results of the dissertation also have been employed in the researches, which are funded Yuzhnoye State Design Office.

Results of the research have been printed int 55 articles. In the journals indexed by Scopus, 3 articles have been printed. In the proceedings of conferences indexed by Scopus, 1 article has been printed. 3 monographs have been printed. In the list of professional editions of Ukraine in technical sciences, 19 articles have been printed, 2 of them indexed by Web of Science. In the list of professional editions of Ukraine in physico-mathematical sciences sciences, 10 articles have been printed. 6 licenses on software have been registered. In proceedings of conferences, 15 articles have been printed.

Keywords: CAD, mathematical model, geometric shape, discrete model, adaptive discrete model, domain-specific language, finite element.

List of applicant's publications

1. Построение дискретных моделей геометрических объектов: монография / В. М. Колодяжный, Д. А. Лисин, А. А. Лисняк, С. В. Чопоров. Харьков: ХНАДУ, 2013. 268 с.
2. Функциональный поход к геометрическому моделированию технических систем: монография / С. В. Чопоров, С. Н. Гребенюк, С. И. Гоменюк [и др.] Запорожье: Запорожский национальный университет, 2016. 176 с.
3. Чопоров С. В. Дискретные модели технических объектов: монография. Херсон: Видавничий дім «Гельветика», 2018. 324 с.
4. Чопоров С. В. Построение неравномерных дискретных сеток для функциональных математических моделей на базе теории R-функций // Радиоэлектроника, информатика, управление. 2011. № 2. С. 70–75.
5. Гоменюк С. И., Чопоров С. В., Лисняк А. А. FORTU-FEM – автоматизированная система для исследования сложных механических процессов // Вестник Херсонского национального технического университета. 2012. № 2(45). С. 101–105.
6. Математическое моделирование некоторых крепежных соединений на базе теории R-функций / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, Е. В. Панасенко // Радиоэлектроника, информатика, управление. 2012. №2(27). С. 117–122.
7. Чопоров С. В., Лисняк А. А., Гоменюк С. И. Подход к определению областей сгущения сетки конечных элементов на базе теории R-функций // Методи розв'язування прикладних задач механіки деформівного твердого тіла. Дніпропетровськ: Ліра. 2012. Вип. 13. С. 413–419.
8. Чопоров С. В., Гоменюк С. И., Лисняк А. А. Сравнительный анализ треугольных и четырехугольных конечных элементов // Вестник Херсонского национального технического университета. 2013. № 2(47). С. 382–386.
9. Чопоров С. В. Использование технологий параллельных вычислений в методе конечных элементов // Радиоэлектроника, информатика, управление. 2013. № 2(29). С. 88–94.

10. Кудин А. В., Чопоров С. В. Компьютерное моделирование изгиба круглой трехслойной пластины с использованием аналитического и численного подходов // Радиоэлектроника, информатика, управление. 2014. № 1(30). С. 75–81.
11. Чопоров С. В. Проблемно-орієнтована мова для моделювання геометричних об'єктів при функціональному підході // Вестник Херсонского национального технического университета. 2014. № 3(50). С. 532–539.
12. Гоменюк С. И., Чопоров С. В., Аль-Омари М. А. В. Функциональное сглаживание сеток конечных элементов // Технология приборостроения. 2015. № 1. С. 10–14.
13. Чопоров С. В., Гоменюк С. И. Параллельный способ построения сеток треугольных элементов при функциональном представлении // Вестник Херсонского национального технического университета. 2015. № 3(54). С. 511–517.
14. Чопоров С. В. Метод фоновой сетки для триангуляции двумерных областей при функциональном подходе. Радиоэлектроника, информатика, управление. 2015. № 4(35). С. 31–38. DOI: 10.15588/1607-3274-2015-4-5 [The journal is indexed in Web of Science].
15. Формалізація моделей многослойных тонкостенных конструкций в САПР / С. В. Чопоров, С. И. Гоменюк, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омари, Х. Х. Алатемнех // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Серія: Нові рішення в сучасних технологіях. 2015. № 62(1171). С. 99–106.
16. Триангуляция поверхностей при гибридном параметрическом и неявном представлении / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, М. А. В. Аль-Омари, Х. Х. Алатамнех // Вестник Херсонского национального технического университета. 2016. № 3(58). С. 542–549.
17. Analytical and Numerical Modelling of the Axisymmetric Bending of Circular Sandwich Plates with the Nonlinear Elastic Core Material / A. Kudin, S. Choporov, Yu. Tamurov, M. A. V. Al Omari // Journal of Solid Mechanics. 2016. Vol. 8, Iss. 3. P. 590–601. [The journal is indexed in Scopus].
18. Чопоров С. В., Аль-Атемнех Б. Г. М., Гоменюк С. И. Підхід до геометричного моделювання в паралельних обчислювальних системах зі спільною

пам'яттю // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 27. С. 164–172.

19. Чопоров С. В. Гибридное представление сплошных тел с использованием неявных и параметрических функций // Радиоэлектроника, информатика, управление. 2017. № 3. С. 60–70. DOI: 10.15588/1607-3274-2017-3-7 [The journal is indexed in Web of Science].

20. Чопоров С. В. Сглаживание сеток четырехугольных элементов с использованием локальной минимизации функционала // Вестник Херсонского национального технического университета. 2017. Т. 2, № 3(62). С. 234–239.

21. Kudin A. V., Choporov S. V., Gomenyuk S. I. Axisymmetric bending of circular and annular sandwich plates with nonlinear elastic core material // Mathematical models and computer simulations. 2017. Vol. 9, No 5. P. 601–612. DOI: doi.org/10.1134/S2070048217050076 [The journal is indexed in Scopus].

22. Математическое обеспечение анализа прочности силовых элементов ракетно-космической техники / Д. В. Акимов, В. З. Грищак, И. Ф. Ларионов, С. И. Гоменюк, Д. В. Клименко, С. В. Чопоров, С. Н. Гребенюк // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 26. С. 5–17.

23. Чопоров С. В. Представление вычислительных сеток в системах инженерного анализа // Вісник Херсонського національного технічного університету. 2018. Т. 1, № 3(66). С. 372–379

24. Choporov S., Homenyuk S, Grebenyuk S. Optimized Smoothing of Discrete Models of the Implicitly Defined Geometrical Objects' Surfaces // Eastern-European Journal of Enterprise Technologies. 2018. Vol. 3/4 (93). P. 52–60. DOI: 10.15587/1729-4061.2018.130787 [The journal is indexed in Scopus].

25. Мухин В. В., Чопоров С. В. Визуализация дифференциальных характеристик дискретных данных // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 2. С. 60–65.

26. Чопоров С. В. Алгебраические способы учета перемещения узла сетки конечных элементов // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 1. С. 117–121.

27. Кудин А. В., Тамуров Ю. Н., Чопоров С. В. Аналитический и численный анализ изгиба круглой трехслойной пластины под действием локальных нагрузок // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 67–81.

28. Модель напружено-деформованого стану зубчатого колеса / Х. Х. Алатамнех, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омарі, С. В. Чопоров // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 2. С. 173–186.

29. Снежкова Л. С., Чопоров С. В. Методы и подходы к моделированию геометрических объектов в контактных задачах // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 171–195.

30. Математическое моделирование и исследование прочности силовых элементов конструкций космических летательных аппаратов / Д. В. Акимов, В. З. Грищак, С. И. Гоменюк, С. Н. Гребенюк, А. А. Лисняк, С. В. Чопоров, И. Ф. Ларионов, Д. В. Клименко, В. Н. Сиренко // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 6–13.

31. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 239–248.

32. Методы построения дискретных моделей: структурированные и блочно-структурированные сетки / С. В. Чопоров, С. И. Гоменюк, Х. Х. Алатамнех, К. С. Оспищев // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 1. С. 272–284.

33. Методы построения дискретных моделей: неструктурированные сетки / С. В. Чопоров, А. А. Лисняк, Ю. А. Борисовская, О. С. Козлова, Л. С. Снежкова // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 2. С. 237–250.

34. Чопоров С. В. Автоматизация анализа температурной устойчивости пластин с использованием метода конечных элементов // Вісник Запорізького

національного університету: Збірник наукових статей. Фізико-математичні науки. 2017. № 2. С. 304–316.

35. Комп'ютерна програма «Система автоматизованого проектування FORTU-FEM»: Свідоцтво про реєстрацію авторського права на твір № 43043 Україна / С. І. Гоменюк, В. З. Грищак, С. М. Гребенюк, С. В. Чопоров, А. О. Лісняк; дата реєстрації 02.04.2012.

36. Комп'ютерна програма «Автоматизована система "Stability"»: Свідоцтво про реєстрацію авторського права на твір № 53201 Україна / С. І. Гоменюк, С. М. Гребенюк, В. З. Грищак, Ю. М. Тамуров, С. В. Чопоров; дата реєстрації 20.01.2014.

37. Комп'ютерна програма «Система автоматизованого проектування «qMesher»»: Свідоцтво про реєстрацію авторського права на твір № 53829 Україна / С. В. Чопоров, С. І. Гоменюк, С. М. Гребенюк, А. О. Лісняк; дата реєстрації 24.02.2014.

38. Комп'ютерна програма «Бібліотека консольної індикації «qzIO» («qzIO»): Свідоцтво про реєстрацію авторського права на твір № 70462 Україна / С. В. Чопоров, В. З. Грищак, С. М. Гребенюк, С. І. Полюга; дата реєстрації 15.02.2017.

39. Комп'ютерна програма «Бібліотека паралельної обробки розріджених матриць qzMatrix» («qzMatrix»): Свідоцтво про реєстрацію авторського права на твір № 70465 Україна / С. В. Чопоров, С. І. Гоменюк, А. О. Лісняк; дата реєстрації 15.02.2017.

40. Комп'ютерна програма «Паралельний інтерпретатор мови специфікацій математичних моделей геометричних об'єктів»: Свідоцтво про реєстрацію авторського права на твір № 82295 Україна / С. В. Чопоров, А. О. Лісняк, О. В. Кудін, М. С. Ігнатченко; дата реєстрації 17.10.2018.

41. Глушко К. А., Мухін В. В., Чопоров С. В. Автоматизована підсистема побудови трьох характеристик поверхні функції двох змінних // Актуальні проблеми математики та інформатики: збірка тез доповідей III Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 26–27 квітня 2012 р.). Запоріжжя: ЗНУ, 2012. – С. 69.

42. Лисняк А. А., Чопоров С. В. Способы определения областей сгущения вычислительной сетки для моделей на базе теории R-функций // Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 5–8 листопада 2012 р.). Харків: Ротапринт Інституту проблем машинобудування ім. А. М. Підгорного НАН України, 2012. – С. 38.

43. Кліщенко Г. О., Чопоров С. В. Розробка способу врахування переміщення вузлів скінченно-елементної сітки // Актуальні проблеми математики та інформатики: збірка тез доповідей IV Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 25–26 квітня 2013 р.). Запоріжжя: ЗНУ, 2013. 135 с.

44. Choporov S. Parallel Computing Technologies in the Finite Element Method // High Performance Computing: Proceedings of Third International Conference HPC-UA 2013 (Kyiv, Ukraine, 7–11 October, 2013). Kyiv: National Technical University of Ukraine "Kyiv Polytechnic Institute", 2013. P. 85–91.

45. Кліщенко Г. О., Чопоров С. В. Розробка способу врахування тріщин конструкцій в сітці на базі чотирикутників // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. С. 147.

46. Снежкова Л. С., Чопоров С. В. Дискретні моделі в контактних задачах на базі теорії R-функцій // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. 147 с.

47. Choporov S. V., Lisnyak A. O., Akimov D. V. Discrete models generation for layered shells of a spacecraft // Aviation in the XXI-st Century. Safety in Aviation and Space Technologies: Proceedings The Sixth World Congress (Kyiv, Ukraine, 23–25 September, 2014). Ukraine, Kyiv: NAU, 2014. V. 1. P. 1.14.45–1.14.48.

48. Оспищев К. С., Чопоров С. В. Моделирование напряженно-деформированного состояния эластомерных конструкций на базе современных информационных технологий // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 89.

49. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 90–91.

50. Чопоров С. В. Гибридный подход к представлению геометрических объектов // Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 21–24 листопада 2016 р.). Харків: Інститут проблем машинобудування ім. А. Н. Підгорного НАН України, 2016. С. 36.

51. Модель напряженно-деформированного состояния тонкостенных конструкций ракетной техники с учетом температурных деформаций / В. З. Гришак, С. И. Гоменюк, С. Н. Гребенюк, С. В. Чопоров, Д. В. Акимов // АВИА-2017: матеріали XIII Міжнародної науково-технічної конференції (м. Київ, 19–21 квітня 2017 р.). Київ: Національний авіаційний університет, 2017. С. 7.41–7.45.

52. Чопоров С. В., Акимов Д. В. Использование элементов связи при функциональном подходе к заданию нагрузок в моделях оболочечных конструкций // Інформатика, управління, штучний інтелект: матеріали IV Міжнародної конференції студентів, магістрів та аспірантів (м. Харків, 21–23 листопада 2017 р.). Харків: НТУ «ХП», 2017. С. 105.

53. Buckling Analysis of Circular Sandwich Plates with the Nonlinear Elastic Core Material / O. Kudin, S. Choporov, S. Gomenyuk, V. Gristchak // Mechanika 2018: Proceedings of 23rd international conference at Kaunas University of Technology (Kaunas, Lithuania, 18 May, 2018). Kaunas University of Technology, 2018. P. 92–100.

54. Чопоров С. В. Проекційний підхід до генерації поверхневих сіток при функціональному поданні тіл // Сучасні проблеми механіки та математики: збірник наукових праць (м. Львів, 22–25 травня 2018 р.) / ред. А. М. Самойленка та Р. М. Кушніра. 2018. Т. 1. С. 139–140. URL: www.iapmm.lviv.ua/mpmm2018 (дата звернення: 18.09.2018).

55. Design Patterns for Object-Oriented Scientific Software / S. Choporov, S. Gomenyuk, O. Kudin, A. Lisnyak // CEUR Workshop Proceedings. 2018. Vol. 2105. P. 441–444. [The journal is indexed in Scopus].

ЗМІСТ

Вступ.....	28
1 Аналіз предметної області та постановка задач дослідження.....	39
1.1 Аналіз стану проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання...	39
1.2 Аналіз сучасних САПР машинобудування та будівництва.....	42
1.3 Аналіз методів побудови дискретних моделей форм об'єктів.....	44
1.3.1 Структуровані дискретні моделі форм об'єктів.....	45
1.3.2 Неструктуровані дискретні моделі форм об'єктів.....	53
1.3.3 Дискретні моделі об'єктів, форму яких можна подати оболонками.....	60
1.4 Аналіз методів оптимізації дискретних моделей форм об'єктів.....	63
1.4.1 Методи оптимізації внутрішніх вузлів та елементів.....	63
1.4.2 Методи оптимізації межових вузлів та елементів.....	65
1.5 Аналіз методів подання форм об'єктів.....	66
1.6 Постановка мети та задач дослідження.....	68
Висновки до розділу 1.....	70
2 Подання форм об'єктів у САПР машинобудування.....	72
2.1 Подання форм об'єктів на базі неявних функцій.....	72
2.2 Моделювання двовимірних форм об'єктів на базі неявних функцій.....	78
2.2.1 Примітиви.....	78
2.2.2 Приклади функціонального подання креслень двовимірних форм об'єктів.....	82
2.3 Моделювання тривимірних форм об'єктів на базі неявних функцій.....	86
2.3.1 Примітиви.....	87
2.3.2 Приклади функціонального подання креслень тривимірних форм об'єктів.....	90
2.4 Гібридний підхід до подання форм об'єктів машинобудування.....	98
2.4.1 Двовимірні форми.....	100

	25
2.4.2 Тривимірні форми.....	108
2.5 Подання оболонкових об'єктів і конструкцій.....	119
Висновки до розділу 2.....	122
3 Формалізація дискретних моделей форм об'єктів у САПР машинобудування.....	124
3.1 Формальне визначення дискретної моделі форми об'єкта машинобудування..	126
3.2 Операції вставки та видалення, пошуку сусідів.....	127
3.3 Структура даних.....	133
Висновки до розділу 3.....	135
4 Дискретні моделі меж об'єктів машинобудування.....	137
4.1 Проекційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально.....	140
4.1.1 Загальний алгоритм метода.....	140
4.1.2 Алгоритм пошуку проекцій вузлів на межу об'єкта.....	142
4.1.3 Приклади застосування.....	145
4.1.4 Двовимірний випадок.....	148
4.1.5 Аналіз результатів.....	149
4.2 Метод згладжування координат вузлів дискретних моделей меж об'єктів, форма яких задана функціонально.....	150
4.2.1 Функціонал відстані-довжини.....	151
4.2.2 Локальний алгоритм пошуку мінімуму функціоналу відстані-довжини....	155
4.2.3 Глобальний алгоритм пошуку мінімуму функціоналу відстані-довжини.	157
4.2.4 Приклади застосування методу.....	158
4.2.5 Двовимірний випадок.....	164
4.2.6 Аналіз результатів.....	165
4.3 Метод локальних перебудов елементів дискретних моделей меж об'єктів.....	165
Висновки до розділу 4.....	171
5 Дискретні моделі форм об'єктів на базі солідів.....	173
5.1 Дво- і тривимірні триангуляції.....	173
5.1.1 Алгоритм метода триангуляції.....	174
5.1.2 Алгоритм оптимізації чисельних характеристик дискретної моделі.....	185

5.2	Метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально.....	192
5.3	Згладжування Лапласа для внутрішніх вузлів.....	202
5.4	Метод згладжування координат внутрішніх вузлів дискретних моделей форм об'єктів.....	207
5.5	Використання методу для побудови дискретних моделей поверхонь об'єктів, поданих гібридно.....	217
	Висновки до розділу 5.....	220
6	Адаптивні дискретні моделі форм об'єктів машинобудування.....	221
6.1	Дослідження впливу форми скінченного елемента на точність рішення.....	221
6.1.1	Двовимірні задачі дослідження напружено-деформованого стану.....	222
6.1.2	Тривимірні задачі дослідження напружено-деформованого стану.....	228
6.2	Метод генерації адаптивних дискретних моделей форм об'єктів, заданих функціонально.....	232
6.3	Метод формування множини елементів для розбиття.....	241
	Висновки до розділу 6.....	252
7	Каркас програмного забезпечення.....	254
7.1	Шаблони проектування для об'єктно-орієнтованих САПР машинобудування.....	254
7.1.1	Декомпозиція та ідентифікація.....	256
7.1.2	Каталог шаблонів проектування.....	257
7.2	Методи генерації та обробки дискретних моделей форм об'єктів на базі технологій паралельних обчислень.....	263
7.2.1	Метод генерації дискретних моделей форм об'єктів з використанням паралельних обчислень.....	263
7.2.2	Скінченно-елементний аналіз з використанням технологій паралельних обчислень.....	266
7.3	Проблемно-орієнтована мова моделювання форм об'єктів.....	273
7.3.1	Базовий синтаксис.....	274
7.3.2	Допоміжні функції й операції.....	277

	27
7.3.3 Основні геометричні примітиви.....	278
7.3.4 Генерація дискретних моделей.....	281
7.3.5 Опис скінченно-елементних задач.....	286
Висновки до розділу 7.....	291
Висновки.....	293
Перелік джерел посилання.....	298
Додаток А. Довідка про використання результатів дисертаційної роботи державним підприємством «Конструкторське бюро «Південне».....	348
Додаток Б. Довідка про використання результатів дисертаційної роботи у навчальному процесі Запорізького національного університету.....	351

ВСТУП

Актуальність теми. Важливою частиною процесу проектування складних технічних об'єктів є дослідження їх експлуатаційних характеристик. Для цього використовують моделі різних видів: графічні, математичні, натурні тощо. Застосування математичного моделювання дозволяє зменшити витрати на натурні дослідження об'єктів, а також зробити прогнози відносно режимів експлуатації. Стан технічних об'єктів (наприклад, напружено-деформований) часто моделюється системами диференціальних рівнянь з частинними похідними, які не мають аналітичного розв'язку. У такому випадку використовують чисельні методи, які засновані на ідеї переходу від неперервної моделі об'єкта до дискретної. У результаті виникає необхідність автоматичної побудови дискретних моделей, які дозволяють розв'язати системи диференціальних рівнянь із заданою точністю.

Побудова дискретних моделей об'єктів машинобудування пов'язана з необхідністю опису їх форм. Для цього використовують різні подання форм об'єктів: граничне, конструктивна блокова геометрія, функціональне тощо. Функціональне подання, яке ґрунтується на використанні неявних функцій, є найбільш універсальним і теоретично дозволяє описати форму об'єкта довільної складності. Неявний опис форми об'єкта дозволяє класифікувати положення точки простору відносно його межі, але потребує розробки методів генерації дискретних моделей як об'єднання елементарних фігур (meshing).

У розвиток теорії та практики розробки математичного забезпечення систем автоматизації проектувальних робіт, а також математичного моделювання форм технічних об'єктів суттєвий вклад внесли такі вчені: Бреславський Д. В. [7], Гоменюк С. І. [15–20], Городецький А. С. [10, 23–25, 86], Киричевський В. В. [28, 31–35, 87, 90, 140, 165], Куценко Л. М. [49–51], Литвин О. М. [60–64], Максименко-Шейко К. В. [69–76, 403–404], Нефьодов Л. І. [3, 97–98], Підгорний О. Л. [104–107, 156], Плоский В. О. [108–111], Рвачов В. Л. [83, 114–123, 153], Романова Т. Є. [2, 136, 244, 389, 475], Соколовський Я. І. [134, 447–448], Стоян Ю. Г. [2, 136–139],

Толок В. О. [87, 149–150], Толок О. В. [143–148, 151, 403, 404], Черніков О. В. [36, 159–164], Шейко Т. І. [66–67, 69–70], Pasko A. [392–393], Requicha A. A. G. [409–413], Shapiro V. [432–435] та інші [1–5, 8, 14, 26–29, 65–68, 77–80, 95–96, 113, 124, 135].

У розвиток методів генерації дискретних моделей форм об'єктів значний вклад зробили такі вчені як Лісняк А. О. [55–58], Скворцов О. В. [125–127], Сковпень А. В. [128–129], Bern M. [208–211], Chew L. P. [234–236], Floater M. S. [268–270], Freitag L. A. [273–275], Joe B. [307–311], Miranda A. C. O. [358–361], Owen S. J. [382–386], Ruppert J. [417–418], Saxena A. [421–425], Schneiders R. [428–430], Shewchuk J. R. [438–441] та інші [185–207, 212–233, 237–266, 276–279, 282–306, 312–357].

Незважаючи на численні публікації, присвячені розв'язанню задач моделювання форм, виявлено протиріччя між необхідністю побудови дискретних моделей форм об'єктів, які дозволять із заданою точністю виконувати чисельний аналіз стану виробів у САПР машинобудування, та обмеженістю методів, що існують. Це визначає **актуальність науково-технічної проблеми** підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного для дослідження їх стану.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась згідно з планом науково-технічних робіт кафедри програмної інженерії Запорізького національного університету, а також у межах теми «Гібридні аналітико-чисельні методи розв'язку актуальних задач неоднорідного середовища» (№ держреєстрації: 0114U002656).

У межах держбюджетних науково-дослідних робіт:

– № 3/12 «Аналітико-чисельні методи розв'язку задач механіки неоднорідних конструкцій на базі сучасних комп'ютерних технологій та візуалізація процесів» (№ держреєстрації: 0112U003061); автор – виконавець, розробник методів генерації дискретних моделей та візуалізації результатів чисельного аналізу;

– № 9/13 «Розробка математичної моделі розподілу напружень у важконавантажених зубчастих колесах і створення прогресивної технології їх

виготовлення» (№ держреєстрації: 0113U000804); автор – виконавець, розробник математичної моделі зубчастого колеса і скінченно-елементної моделі розподілу напружень у ньому;

– № 3/15 «Математичне моделювання конструкцій неоднорідної структури на базі сучасних інформаційних технологій» (№ держреєстрації: 0115U000761); автор – виконавець, розробник методів генерації дискретних моделей конструкцій неоднорідної структури;

– № 5/17 «Розробка математичного забезпечення для інженерного аналізу об'єктів аерокосмічної техніки на базі хмарних технологій» (№ держреєстрації: 0117U007204). Автор – науковий керівник науково-дослідної роботи.

У межах госпдоговірних науково-дослідних робіт:

– договір № 1/15 від 05.02.2015 р. «Розроблення програмно-методичного забезпечення для визначення руйнівних навантажень оболонок тришарової стільникової конструкції із композиційного матеріалу в умовах комплексного експлуатаційного навантаження» (№ держреєстрації: 0115U004162), який укладено з державним підприємством «Конструкторське бюро «Південне» ім. М. К. Янгеля»; автор – виконавець, розробник математичних моделей головного обтікача й міжступенного відсіку, а також відповідних скінченно-елементних моделей;

– договір № 5/16 від 05.06.2016 р. «Розробки методики визначення руйнівних навантажень оболонок під дією внутрішнього тиску з урахуванням пластичних деформацій матеріалу і чисельного прогнозування зон руйнування оболонок під дією внутрішнього надлишкового тиску з урахуванням нагріву» (№ держреєстрації: 0116U006602), який укладено з державним підприємством «Конструкторське бюро «Південне» ім. М. К. Янгеля»; автор – виконавець, розробник дискретних моделей оболонкових конструкцій для чисельного прогнозування зон руйнування;

– договір № 8/17 від 16.05.2017 р. «Розробка методики визначення напружено-деформованого стану головного обтічника тришарової стільникової конструкції з раціональним вибором характеристик анізотропії матеріалу в залежності від характеру зовнішнього навантаження конструкцій та розробки розрахункової моделі для визначення критичних значень навантажень» (№ держреєстрації: 0117U001632),

який укладено з державним підприємством «Конструкторське бюро «Південне» ім. М. К. Янгеля»; автор – виконавець, розробник математичної моделі та програмного модуля для дослідження стану оболонок з урахуванням температурних деформацій;

– договір № 4/18 від 30.05.2018 р. «Розробка методики визначення зон руйнування і граничних тисків для вафельних оболонок паливного відсіку першого ступеню» (№ держреєстрації: 0118U001846), який укладено з державним підприємством «Конструкторське бюро «Південне» ім. М. К. Янгеля»; автор – виконавець, розробник дискретної моделі паливного відсіку першого ступеню;

– договір № 5/18 від 30.05.2018 р. «Відпрацювання і верифікації (за результатами статичних випробувань) методики визначення зон руйнування паливного відсіку другого ступеню при навантаженні внутрішнім надлишковим тиском» (№ держреєстрації: 0118U001845), який укладено з державним підприємством «Конструкторське бюро «Південне» ім. М. К. Янгеля»; автор – виконавець, розробник дискретної моделі паливного відсіку другого ступеню.

Також протягом виконання Гранта Президента України для молодих учених на 2013 рік у межах науково-дослідної роботи №Ф49/429-2013 «Математичне моделювання складних геометричних об'єктів з використанням технологій паралельних розрахунків» (№ держреєстрації: 0113U006252). Автор – науковий керівник науково-дослідної роботи.

Мета та задачі дослідження. Метою дисертаційної роботи є розробка ефективних методів математичного моделювання форм об'єктів у САПР машинобудування.

Для досягнення поставленої мети у дисертаційній роботі необхідно розв'язати такі основні задачі:

- виконати огляд і аналіз сучасного стану проблеми;
- розробити гібридний підхід до подання форм об'єктів машинобудування з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання);

- розробити проєкційний метод генерації дискретних моделей меж об’єктів, форма яких задана функціонально;
- розробити метод згладжування координат вузлів дискретних моделей меж об’єктів, форма яких задана функціонально;
- модифікувати метод локальних перебудов елементів дискретних моделей на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об’єктів;
- розробити метод генерації дискретних моделей на базі елементів типу солід для форм об’єктів, заданих функціонально;
- розробити метод згладжування координат внутрішніх вузлів дискретних моделей форм об’єктів;
- розробити метод генерації адаптивних дискретних моделей форм об’єктів (adaptive meshing), заданих функціонально, для забезпечення скінченно-елементного аналізу поведінки об’єкта із заданою точністю;
- розробити методи генерації дискретних моделей форм об’єктів і скінченно-елементного аналізу з використанням технологій паралельних обчислень;
- розробити проблемно-орієнтовану мову для опису об’єктів машинобудування складної форми з використанням гібридного підходу, а також задач побудови дискретних моделей та скінченно-елементного аналізу;
- розробити САПР для апробації розроблених методів при дослідженні поведінки об’єктів машинобудування.

Об’єкт дослідження – об’єкти машинобудування.

Предмет дослідження – математичні моделі форм об’єктів машинобудування.

Методи дослідження. Дослідження базується на комплексному використанні методів аналітичної геометрії, математичного аналізу та математичного моделювання: для опису форм об’єктів використовується теорія R-функцій; для побудови дискретних моделей форм об’єктів – методи аналітичної геометрії, двійкового пошуку; для мінімізації функціоналів при пошуку оптимальних координат вузлів – методи покоординатного спуску і спряжених градієнтів; для побудови адаптивних дискретних моделей форм об’єктів – метод скінченних

елементів; для апробації розроблених методів у САПР – об’єктно-орієнтований підхід.

Наукова новизна отриманих результатів. У результаті виконання дисертаційного дослідження розроблено математичні моделі, наукові методи та програмне забезпечення для математичного моделювання та аналізу форм об’єктів у САПР машинобудування. Водночас отримано такі нові наукові результати:

- вперше запропоновано гібридний підхід до подання форм об’єктів з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання), що дозволяє зменшити час побудови математичних моделей;

- вперше запропоновано проекційний метод генерації дискретних моделей меж об’єктів, форма яких задана функціонально, що дозволяє будувати дво- та тривимірні дискретні моделі;

- вперше запропоновано метод згладжування координат вузлів дискретних моделей меж об’єктів, форма яких задана функціонально, у результаті чого можна підвищити точність моделювання в околах зламів їх поверхонь;

- отримав подальшого розвитку метод локальних перебудов елементів дискретних моделей у частині його узагальнення на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об’єктів, що дозволяє підвищити точність скінченно-елементного аналізу;

- вперше запропоновано метод генерації дискретних моделей на базі елементів типу солід для форм об’єктів, заданих функціонально, який дозволяє будувати дво- і тривимірні моделі як з використанням симплексів (трикутників або тетраедрів), так і з використанням топологічних кубів (чотирикутників або шестигранників);

- вперше запропоновано метод згладжування координат внутрішніх вузлів дискретних моделей форм об’єктів, відмінність якого від інших методів полягає в усуненні вироджених елементів та підвищенні точності скінченно-елементного аналізу;

- вперше запропоновано метод генерації адаптивних дискретних моделей форм об’єктів (adaptive meshing), заданих функціонально, який дозволяє будувати

дво- і тривимірні дискретні моделі для виконання скінченно-елементного аналізу із потрібною точністю;

- отримали подальшого розвитку методи генерації дискретних моделей форм об'єктів і скінченно-елементного аналізу у частині розробки паралельних алгоритмів, що дозволяє зменшити витрати часу на моделювання;

- вперше розроблена проблемно-орієнтована мова, яка дозволяє описувати об'єкти машинобудування складної форми з використанням гібридного підходу, а також задачі побудови дискретних моделей та скінченно-елементного аналізу.

Наукові результати, отримані у результаті виконання дисертаційного дослідження, у сукупності є вирішенням науково-технічної проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного на дослідження їх стану. Очікувані науково-технічні ефекти:

- підвищення якості візуального подання моделей об'єктів машинобудування;
- зниження витрат на натурні випробування об'єктів, що проектуються, за рахунок підвищення точності моделювання при дослідженні їх стану з використанням чисельних методів на базі дискретних моделей.

Наукові результати, отримані в дисертаційній роботі, відповідають формулі спеціальності 05.13.12 та напрямкам досліджень № 2, 4, 6 і 7 згідно з паспортом спеціальності, що свідчить про створення математичного, лінгвістичного та програмного забезпечення систем автоматизації проектування машинобудування.

Практичне значення отриманих результатів. Розроблені у дисертаційній роботі методи можуть використовувати конструкторські бюро і науково-дослідні організації для розробки САПР. Розв'язання задач дослідження дозволило розробити САПР для математичного моделювання та чисельного аналізу стану об'єктів машинобудування на базі адаптивних дискретних моделей.

Результати дисертаційної роботи використані у науково-виробничій діяльності державного підприємства «Конструкторське бюро «Південне» ім. М. К. Янгеля». Результати роботи також використані у навчальному процесі кафедри програмної інженерії Запорізького національного університету для розробки лабораторних робіт

у курсах «Паралельні та розподілені обчислення» і «Програмне забезпечення наукових досліджень».

Особистий вклад здобувача. За результатами дослідження опубліковано 55 робіт: серед них – 3 монографії [112, 158, 182]; 21 стаття у наукових періодичних виданнях (з них 3 статті – у періодичних виданнях, які включено до наукометричної бази Scopus [193, 238, 324]; 2 статті – у періодичних виданнях, які включено до наукометричної бази Web of Science [170, 176]; 19 робіт – у наукових фахових виданнях України, що включено до Переліку МОН України з технічних наук [21–22, 48, 82, 84, 152, 157, 169–170, 172–178, 180, 181, 238]); 10 статей – у наукових фахових виданнях України, що включено до Переліку МОН України з фізико-математичних наук, які додатково висвітлюють наукові результати [47, 81, 88, 89, 92, 94, 131, 133, 166, 168]; 6 свідоцтв про реєстрацію авторського права на комп'ютерні програми [40–45]; 15 робіт – у збірниках матеріалів конференцій та тез доповідей [13, 37, 38, 59, 91, 103, 130, 132, 167, 171, 179, 219, 239, 240, 252] (з них 1 – у збірнику, який включено до наукометричної бази Scopus [252]).

Усі основні наукові результати, які виносяться на захист, отримані автором самостійно. Матеріали, які складають основу дисертаційної роботи, опубліковані в роботах [166, 168, 170–171, 174, 176–182, 240], виконаних одноосібно, складають особистий внесок здобувача. У роботах, які опубліковано зі співавторами, здобувачу належать такі результати: [21] – розробка методу згладжування дискретних моделей з використанням мінімізації функціоналу; [22] – розробка моделей об'єктів машинобудування; [40–44] – розробка методів генерації дискретних моделей та програмування алгоритмів на їх базі; [45] – розробка специфікацій мови опису форм об'єктів; [47–48, 193, 219, 324] – розробка скінченно-елементних моделей для чисельного аналізу напружено-деформованого стану та стійкості тришарових пластин; [81] – розробка дискретних моделей для аналізу міцності конструкцій ракетної техніки; [82, 92, 94, 133] – розробка математичних моделей форм об'єктів; [84] – аналіз сучасних САПР для дослідження стану ракетної техніки; [89, 88] – аналіз методів генерації дискретних моделей форм; [112] – аналіз методів генерації дискретних моделей геометричних об'єктів і розробка методів генерації дискретних

моделей на базі чотирикутників та шестигранників; [131] – аналіз методів математичного моделювання форм об’єктів; [152] – розробка методу тріангуляції при гібридному поданні форм об’єктів; [157] – розробка проблемно-орієнтованої мови для опису задач аналізу оболонкових конструкцій у САПР машинобудування; [158] – розробка моделей об’єктів машинобудування та методу побудови адаптивної тріангуляції; [169] – розробка алгоритму для обчислення моделей форм об’єктів у паралельних системах; [172] – розробка скінченно-елементних моделей для оцінки точності аналізу на базі елементів різної форми; [173] – розробка методу тріангуляції на базі технологій паралельних обчислень; [175] – розробка методу для знаходження областей, у яких необхідно збільшити кількість елементів дискретної моделі; [238] – метод пошуку координат вузлів дискретних моделей меж об’єктів; [252] – розробка об’єктно-орієнтованих шаблонів проектування САПР машинобудування на базі методу скінченних елементів.

Апробація результатів дисертаційної роботи. Основні результати дисертаційної роботи доповідалися і обговорювалися на міжнародних конференціях, у тому числі:

– IV Міжнародній науково-технічній конференції «Актуальні проблеми прикладної механіки і міцності конструкцій» (м. Запоріжжя, 2012 р.);

– V Міжнародній науково-технічній конференції «Комп’ютерні системи та мережні технології» (м. Київ, 2012 р.);

– XIII Міжнародній конференції з математичного моделювання МКММ 2012 (м. Херсон, 2012 р.);

– XIV Міжнародній конференції з математичного моделювання МКММ 2013 (м. Херсон, 2013 г.);

– The 3rd International Conference on High Performance Computing HPC-UA 2013 (м. Київ, 2013 р.);

– XV Міжнародній конференції з математичного моделювання МКММ 2014 (м. Херсон, 2014 р.);

– VI Всесвітньому конгресі «Авіація у XXI столітті» – «Безпека в авіації та космічні технології» (м. Київ, 2014 р.);

- V Міжнародній науково-технічній конференції «Актуальні проблеми прикладної механіки та міцності конструкцій» (м. Запоріжжя, 2015 р.);
- XVI Міжнародній конференції з математичного моделювання МКММ 2015 (м. Херсон, 2015 р.);
- XV Міжнародній конференції «Проблеми інформатики та моделювання (ПІМ-2015)» (м. Одеса, 2015 р.);
- XVII Міжнародній конференції з математичного моделювання МКММ 2016 (смт. Лазурне, Херсонська обл., 2016 р.);
- XIII Міжнародній науково-технічній конференції «ABIA-2017» (м. Київ, 2017 р.);
- VI Міжнародній науковій конференції «Космічні технології: сучасне та майбутнє» (м. Дніпро, 2017 р.);
- VI Міжнародній науково-технічній конференції «Актуальні проблеми прикладної механіки та міцності конструкцій» (м. Запоріжжя, 2017 р.);
- VI Міжнародній науково-технічній конференції «Інформаційні системи та технології ICT-2017» (Харків-Коблево, 2017 р.);
- XVIII Міжнародній конференції з математичного моделювання МКММ 2017 (смт. Лазурне, Херсонська обл., 2017 р.);
- VI Міжнародному радіоелектронному форумі «Прикладна радіоелектроніка. Стан та перспективи розвитку» (м. Харків, 2017р.);
- IV Міжнародній науково-технічній конференції «Інформатика, управління та штучний інтелект» (м. Харків, 2017 р.);
- Міжнародній науковій конференції «Сучасні проблеми механіки та математики» (м. Львів, 2018 р.);
- The 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer ICTERI 2018 (м. Київ, 2018 р.);
- The 23rd International Conference «Mechanika-2018» (Druskininkai, Lithuania, 2018 р.);
- XIX Міжнародній конференції з математичного моделювання (смт. Лазурне, Херсонська обл., 2018 р.).

Результати дисертаційної роботи доповідалися і обговорювалися на наукових семінарах кафедри програмної інженерії Запорізького національного університету.

У повному обсязі робота доповідалася на VII Міжнародній науково-технічній конференції «Інформаційні системи та технології ICT-2018» (Харків-Коблево, 2018 р.), а також на міжвузівському науковому семінарі «Актуальні проблеми прикладної математики і механіки» (м. Запоріжжя, 2018 р.).

Структура та обсяг дисертації. Дисертаційна робота складається зі вступу, семи розділів, висновків, переліку використаних джерел з 505 найменувань (50 с.), двох додатків (5 с.). Загальний обсяг дисертаційної роботи складає 352 сторінки, у тому числі 297 сторінок основного тексту, який проілюстрований 135 рисунками (34 с.) та 14 таблицями (3 с.).

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання

Актуальні тенденції у розвитку таких галузей як будівництво, авіа-, машино-, судно- і ракетобудування направлені на підвищення якості об'єктів, що випускаються, а також зменшення їх собівартості. Для цього використовують методи проектування, включно експериментальне та математичне моделювання, конструювання тощо.

Часто на практиці об'єкти машинобудування мають складну форму. Об'єкти бувають як суцільними тілами, так і тонкостінними оболонками із композиційних матеріалів. Їх натурні випробування потребують підготовки умов, що відповідають режимам експлуатації об'єкта, і, як наслідок, є вельми вартісними. У таких випадках для підвищення ефективності процесу проектування об'єктів машинобудування використовують математичне моделювання на базі комп'ютерної техніки. Водночас факторами, що впливають на ефективність процесу проектування об'єктів машинобудування, стають точність математичних моделей та час, необхідний на комп'ютерне моделювання. У результаті, останнім часом значно збільшилась кількість робіт, які присвячено розробці математичних моделей для проектування в авіакосмічній галузі [29–30, 68, 95–96, 113, 124], машинобудуванні [1, 4, 54, 85, 102] тощо [5, 71, 75, 135, 141, 183].

Отримані при проектуванні на стадії науково-дослідних робіт математичні моделі об'єктів часто базуються на складних диференціальних та/або інтегральних рівняннях, для розв'язку яких потрібно використовувати чисельні методи. В основу багатьох поширених чисельних методів (наприклад, методів граничних елементів, скінченних елементів, скінченних різниць тощо) покладено ідею переходу від

неперервної моделі до її дискретного аналогу. В одному з найбільш поширених чисельних методів, методі скінченних елементів (як і у деяких інших), для визначення стану об'єкта його форма подається простими фігурами (наприклад, трикутниками або тетраедрами), для котрих обчислюється апроксимація розв'язку вихідного рівняння. Відповідно, важливим кроком такого аналізу є побудова дискретної моделі форми об'єкта, яка дозволить отримати розв'язок із потрібною точністю.

При використанні дискретних моделей форм об'єктів спрощено процес дослідження об'єкта, що проектується, можна подати такою послідовністю кроків: 1) побудова математичної моделі форми об'єкта; 2) автоматична генерація еквівалентної дискретної моделі; 3) комп'ютерний аналіз необхідних характеристик. Отримані в результаті комп'ютерного аналізу параметри поведінки досліджуваного об'єкта можна використовувати для прийняття проектних рішень або оптимізації характеристик проектуваного об'єкта (наприклад, для мінімізації маси, оптимізації параметрів армування обшивок тощо). У останньому випадку концептуальна схема автоматизації етапу науково-дослідних робіт може бути подана алгоритмом (рис. 1.1), у якому на першому кроці формалізується модель. Математична модель виступає як вхідний параметр препроцесора, який генерує дискретну модель на другому кроці. Отримана дискретна модель є вхідним параметром для процесора чисельного методу. У результаті чисельного аналізу на третьому кроці отримують необхідні параметри стану об'єкта, що проектується. Параметри стану є вхідними для модуля оптимізації, який коригує роботу препроцесора (наприклад, варіюючи товщиною об'єкта) поки не будуть отримані оптимальні значення параметрів стану об'єкта або не буде забезпечена потрібна точність моделювання.

Необхідно відзначити, що точність математичного моделювання залежить від адекватності та типу використовуваних дискретних моделей форм об'єктів. Сьогодні у практиці математичного моделювання використовують одновимірні, двовимірні та тривимірні дискретні моделі форм, засновані на трикутниках, тетраедрах, чотирикутниках, шестигранниках тощо. Нижче проведений огляд основних підходів та методів побудови і оптимізації дискретних моделей форм об'єктів.

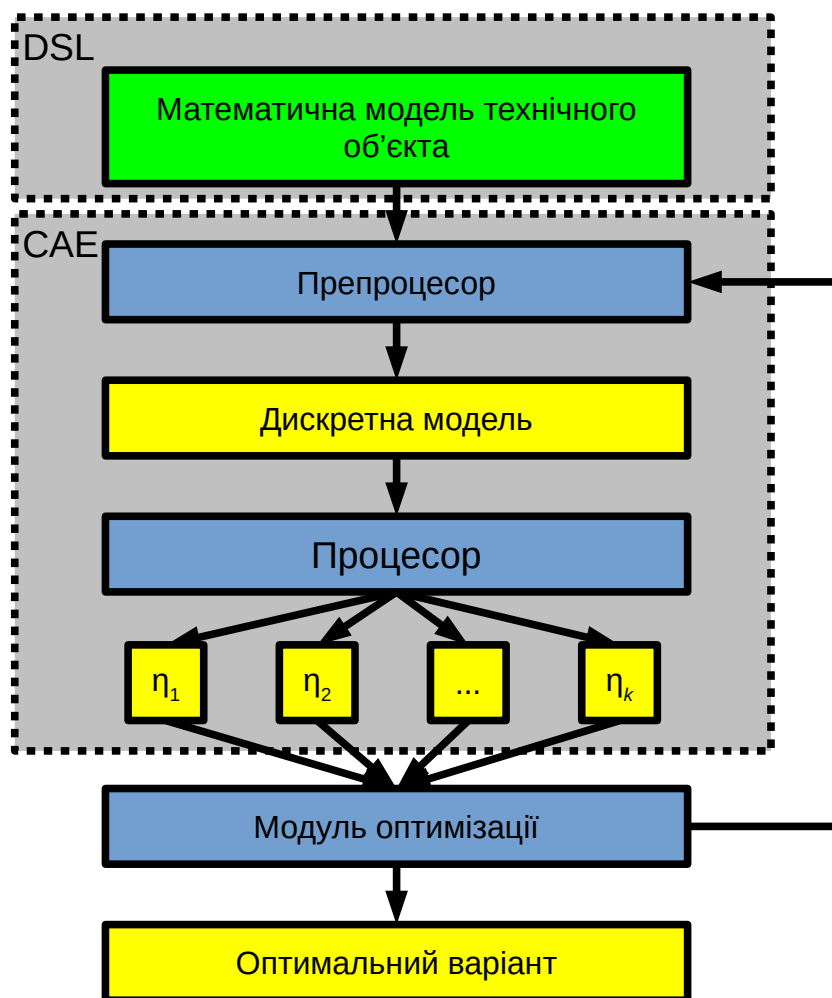


Рисунок 1.1 – Концептуальна схема автоматизації стадії науково-дослідних робіт при проектуванні об'єктів машинобудування

Водночас необхідно забезпечити введення модельної інформації проектувальником зрозумілим для нього способом. Актуальним напрямом пошуку розв'язку задачі формалізації вихідних даних в САПР є розробка лінгвістичного забезпечення – проблемно-орієнтованих мов. Вхідна мова повинна включати в себе дві складові: мову опису об'єктів і мову опису завдань [99].

Мова опису об'єктів, у контексті даної проблеми, повинна надавати засоби опису форм об'єктів та їх фізико-механічних властивостей. Мова опису завдань повинна включати засоби опису певних проектних операцій або процедур (наприклад, розрахунку з використанням методу скінченних елементів, оптимізації тощо).

1.2 Аналіз сучасних САПР машинобудування та будівництва

На сьогодні розроблено значну кількість як вітчизняних так і зарубіжних САПР, які надають засоби для чисельного аналізу об'єктів машинобудування та будівництва. Серед цих систем найбільш популярними є ANSYS, NASTRAN, PATRAN, ABAQUS, LS-DYNA, CATIA, PRO / ENGINEER, COSMOS, LIRA тощо.

Широко поширеною є система ANSYS [347], яка дозволяє розв'язувати значний клас задач інженерного аналізу. Серед них задачі дослідження напружено-деформованого стану при статичному і динамічному навантаженнях, механіки руйнування [260], переносу тепла [319], течії газів і рідин [398] тощо. Інструментарій дозволяє використовувати лінійні і нелінійні скінченні елементи, бібліотеку матеріалів і вибирати процесор методу скінченних елементів (MCE). Зокрема, як процесор MCE в системі ANSYS доступний LS-DYNA [195–196], який дозволяє розв'язувати задачі динамічної взаємодії твердих тіл, задачі удару, захисту лопаток, штампування та виливки.

Система MSC NASTRAN дозволяє скінченно-елементний аналіз напружено-деформованого стану, запасу міцності, власних частот і форм коливань, стійкості, усталені та несталі динамічні процеси, задач теплопередачі, акустичних явищ [364] тощо. Цю систему використовують для скінченно-елементного аналізу композитів [476], ламінатів [477], аерокосмічних матеріалів [202] тощо. Ця САПР дозволяє використовувати високопродуктивні обчислення [365].

САПР Patran [394] є інтегрувальною для систем інженерного аналізу, моделювання та проектування. Дана САПР забезпечує можливість інтегрування систем геометричного та скінченно-елементного моделювання, обробки результатів обчислень. [288], дослідження міцності у лінійній [247] та нелінійній [336] постановках, задач динаміки [256] тощо.

Скінченно-елементний програмний комплекс ABAQUS можна розглядати як САПР функціонального проектування (CAE). Ця система є вельми гнучкою, вона підтримує розширення на мові програмування Python [399]. Її використовують для

розв'язання задач контактної взаємодії [355], механіки руйнування [332], композитних матеріалів [476], переносу тепла [330] тощо.

До типу CAE також відноситься система COSMOS [314, 352], в основу якої покладено МСЕ. Ця САПР дозволяє розв'язувати статичні та динамічні задачі механіки твердого тіла, досліджувати процеси переносу тепла, динаміки газів і рідин тощо.

Вітчизняні програмні комплекси (ПК) ЛІРА [52, 53] та МОНОМАХ [93], які розроблені під керівництвом професора А. С. Городецького, є відомими САПР будівництва та машинобудування. ПК ЛІРА призначений для проектування та розрахунку (у тому числі дослідження міцності та стійкості) конструкцій будівництва та машинобудування. ПК МОНОМАХ – для автоматизованого проектування багатоповерхових і каркасних залізобетонних конструкцій. В основу підсистем інженерного аналізу обох ПК покладено МСЕ (метод переміщень).

Інший вітчизняний ПК МІРЕЛА+ [28, 31–35, 87, 90] розроблений під керівництвом професора Киричевського В. В. Цей ПК написано на мові FORTRAN. Він дозволяє автоматизувати скінченно-елементний аналіз конструкцій з еластомерних матеріалів.

САПР FORTU-FEM [15–20, 27] базується на використанні об'єктно-орієнтованої мови FROTU-3 для формалізації скінченно-елементних моделей технічних об'єктів і процесів. Водночас користувач може самостійно визначити функціонал, машинізація якого буде відповідати стану системи, що моделюється.

Комп'ютерна система «POLYE-RL» [117–118] призначена для розв'язку крайових задач математичної фізики з використанням теорії R-функцій, у тому рахунку на об'єктах фрактальної геометрії.

Потрібно зазначити, що розроблено також інші САПР, засновані на МСЕ. Серед них можна виокремити, наприклад, ADINA, HyperWorks, COMSOL Multiphysics, а також системи з відкритим вихідним кодом: MFree2D, FreeFem++, Elmer тощо [6, 39].

Популярність МСЕ у системах CAE доводить актуальність вирішення проблеми підвищення ефективності проектування об'єктів машинобудування проблеми шляхом збільшення точності скінченно-елементного аналізу.

1.3 Аналіз методів побудови дискретних моделей форм об'єктів

Швидкий розвиток комп'ютерних систем і САПР дозволили значно підвищити швидкість чисельного розв'язку задач, пов'язаних з проектуванням. Незважаючи на розвиток безсіткових методів [216, 339–340, 405] в автоматизованому проектуванні для чисельного аналізу найбільш поширеним є МСЕ [503–505]. Його використовують для розв'язання задач різних класів, які включають статику і динаміку. Ключовою для МСЕ є вимога побудови дискретної моделі форми об'єкта – декомпозиції форми об'єкта на множину елементарних фігур (скінченних елементів).

Під дискретною моделлю форми об'єкта (або сіткою, ці терміни використовують синонімічно) розуміють пару множин: множина вершин і множина зв'язків між вершинами (множина елементів). За суттю, побудова дискретної моделі форми об'єкта – це заміна вихідної неперервної області кінцевим об'єднанням найпростіших фігур: трикутників, чотирикутників, тетраєдрів, шестигранників тощо. У проектуванні це визначення можна доповнити обчислювальним процесом, що визначено на дискретній моделі форми об'єкта, за результатами роботи якого вносяться оптимізації у цю модель.

За розмірністю простору дискретні моделі форм об'єктів ділять на одновимірні, двовимірні і тривимірні. Дискретні моделі бувають структурованими (рис. 1.2, а) та неструктурованими (рис. 1.2, б, в). У структурованих дискретних моделях ребра елементів відповідають координатним напрямкам деякої (у загальному випадку криволінійної) системи координат. Навпаки, у неструктурованих дискретних моделях вибір ребер не має прив'язки до координатних напрямків. У структурованих дискретних моделях найбільшого поширення набули чотирикутники і шестигранники, а в неструктурованих часто використовують трикутники і тетраєдри (які спільно називають «симплексами»).

В цілому, найбільш популярними плоскими дискретними елементами є трикутники і чотирикутники. Також у деяких випадках (наприклад, в задачах оптимізації топології об'єктів, дослідження глобальних змін клімату тощо)

використовують плоскі шестикутники [325–326, 327, 346, 421–426, 462–464]. Аналогічно, у тривимірному просторі найбільш популярними є тетраедри і шестигранники (як природні аналоги, відповідно, трикутника і чотирикутника). Використовують піраміди і трикутні призми при генерації дискретних моделей рухом, а у задачах, наприклад, дослідження сферичних об'єктів розглядають дискретні моделі на базі ікосаедрів [295–296, 468, 455, 481].

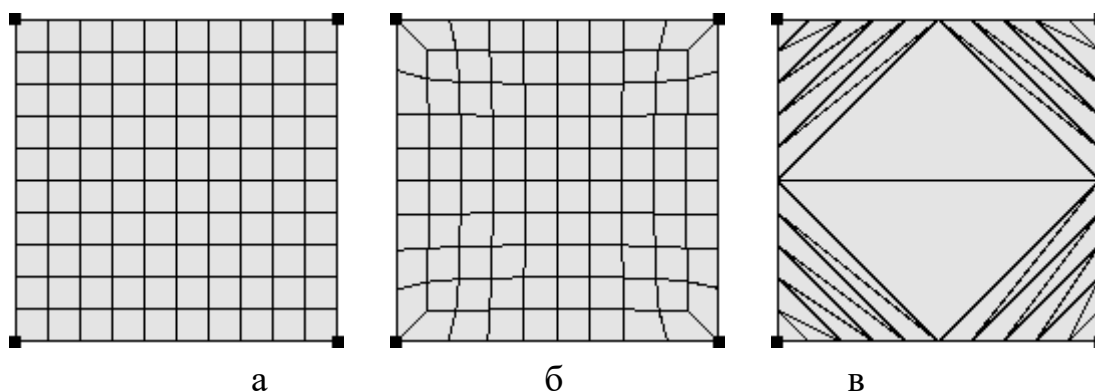


Рисунок 1.2 – Типи дискретних моделей:

а – структурована

б – неструктурована (чотирикутники)

в – неструктурована (трикутники)

1.3.1 Структуровані дискретні моделі форм об'єктів

Основною ознакою структурованих і блочно структурованих дискретних моделей форм об'єктів є наявність регулярних зв'язків між вузлами. Найбільш часто такі моделі використовують чотирикутні елементи на площині або шестигранні в тривимірному просторі. Водночас для зберігання структурованих дискретних моделей можна використовувати найпростіші структури даних – матриці, в яких сусідні вузли визначають сусідні елементи. Такі моделі виникли як основа методу скінченних різниць. Вони використовують ідею трансформації сітки, визначеної на одиничному квадраті (кубі в тривимірному випадку), у форму об'єкта за допомогою

відображення $F: [0, 1]^n \rightarrow \Omega$ (n – розмірність простору). Відображення F можна визначити явно (наприклад, ізопериметричне, конформне тощо) або неявно (наприклад, як розв’язок еліптичних рівнянь з частинними похідними). Актуальні наукові дослідження в області побудови таких дискретних моделей зосереджено на пошуку відображення F .

При побудові структурованих дискретних моделей прийнято розглядати дві координатні області: фізичну і розрахункову. Фізична область – це область Ω , для якої будують дискретну модель. Розрахунковою областю виступає одиничний квадрат (куб). Можна вважати, що у розрахунковій області визначена прямокутна система координат, а у фізичній – криволінійна. Наприклад, якщо фізична область є прямокутником, то побудова розрахункової сітки є тривіальною:

$$\begin{aligned}x_i &= x_0 + \frac{w}{n}(i-1), \quad i = \overline{1, n}, \\y_j &= y_0 + \frac{h}{m}(j-1), \quad j = \overline{1, m},\end{aligned}$$

де $(x_0; y_0)$ – координати нижнього лівого кута прямокутника;

w і h – відповідно, ширина та висота прямокутника;

n – кількість вузлів уздовж осі абсцис;

m – кількість вузлів уздовж осі ординат.

Для криволінійних фізичних областей необхідно знаходити відповідні перетворення координат. Наприклад, якщо вважати, що фізична область обмежена двома радіусами $r_0 \leq r \leq r_1$ і сторонами кута $0 \leq \alpha \leq \theta$, то перетворення

$$\begin{aligned}x(\xi, \eta) &= [r_0 + (r_1 - r_0)\eta] \cos(\theta \xi), \\y(\xi, \eta) &= [r_0 + (r_1 - r_0)\eta] \sin(\theta \xi)\end{aligned}$$

встановить відповідність з розрахунковою областю в системі координат (ξ, η) (наприклад, якщо $r_0=1$, $r_1=4$ і $\theta=\pi/4$, можна отримати фізичну область у вигляді, поданому на рис. 1.3).

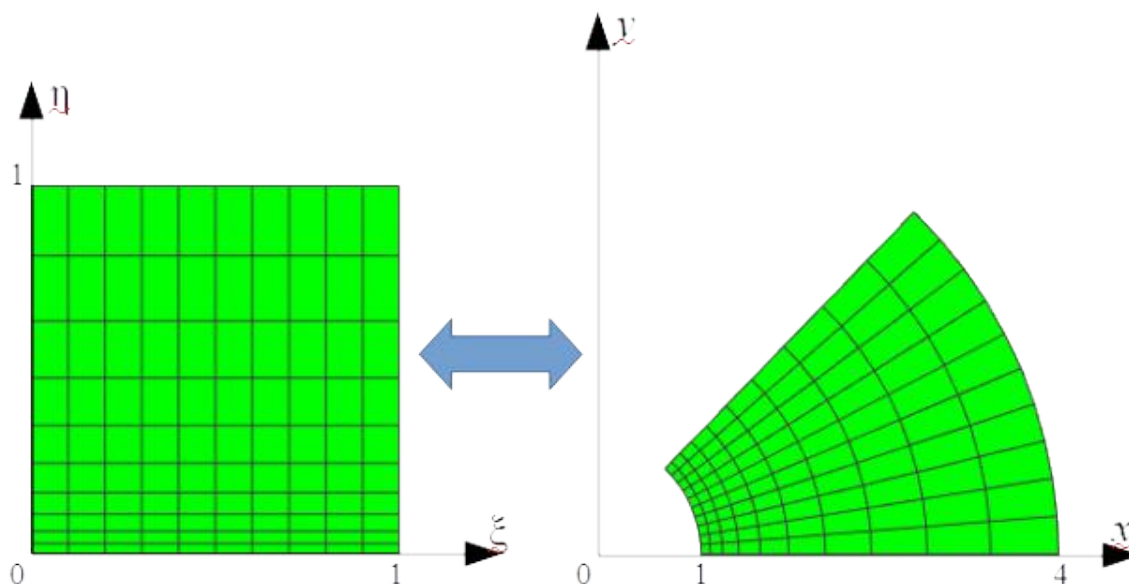


Рисунок 1.3 – Розрахункова та фізична область для кільцевого об'єкта

У випадках, коли подібний аналітичний зв'язок між розрахунковою та фізичною областями встановити важко, використовують методи інтерполяції. Наприклад, якщо область обмежена двома параметричними кривими, то для встановлення відповідності між фізичною та розрахунковою областями, можна використовувати односпрямовану інтерполяцію:

$$\begin{aligned} x(\xi, \eta) &= (1 - \xi)x_l(\eta) + \xi x_r(\eta), \\ y(\xi, \eta) &= (1 - \xi)y_l(\eta) + \xi y_r(\eta), \end{aligned} \quad (1.1)$$

де $\begin{cases} x_l(\eta) \\ y_l(\eta) \end{cases}$ і $\begin{cases} x_r(\eta) \\ y_r(\eta) \end{cases}$ – параметричне подання, відповідно, «лівої» і «правої» кривих

(для випадку, коли інтерполяція має горизонтальний характер, рис. 1.4, а).

Якщо інтерполяція має вертикальний характер (рис. 1.4, б), то формули будуть аналогічними до (1.1).

Якщо область обмежена чотирма кривими, то можна визначити відображення, яке ставить у відповідність квадрат $((0; 0), (0; 1), (1; 0), (1; 1))$, визначений у розрахунковій системі координат, фізичній області $ABCD$ (рис. 1.5). Таке відображення можна отримати послідовним застосуванням відображень (1.1) у кожному координатному напрямі з відніманням результатів білінійної інтерполяції:

$$\begin{aligned}
 x(\xi, \eta) &= (1-\xi)x_l(\eta) + \xi x_r(\eta) + (1-\eta)x_b(\xi) + \eta x_t(\xi) - \\
 &\quad - (1-\xi)(1-\eta)x_b(0) - (1-\xi)\eta x_t(0) - \xi(1-\eta)x_b(1) - \xi\eta x_t(1), \\
 y(\xi, \eta) &= (1-\xi)y_l(\eta) + \xi y_r(\eta) + (1-\eta)y_b(\xi) + \eta y_t(\xi) - \\
 &\quad - (1-\xi)(1-\eta)y_b(0) - (1-\xi)\eta y_t(0) - \xi(1-\eta)y_b(1) - \xi\eta y_t(1).
 \end{aligned} \tag{1.2}$$

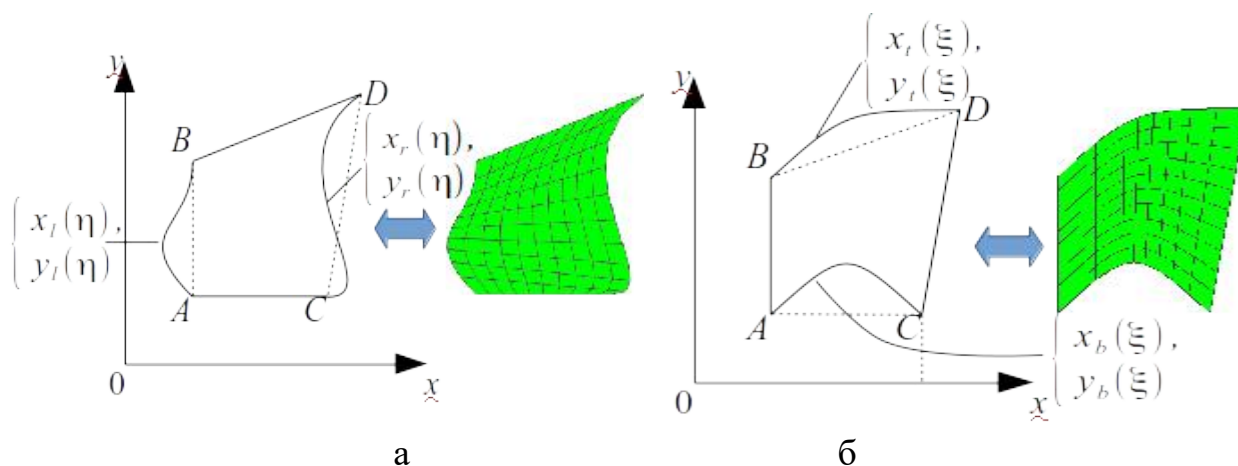


Рисунок 1.4 – Односпрямована інтерполяція:

а – горизонтальна

б – вертикальна

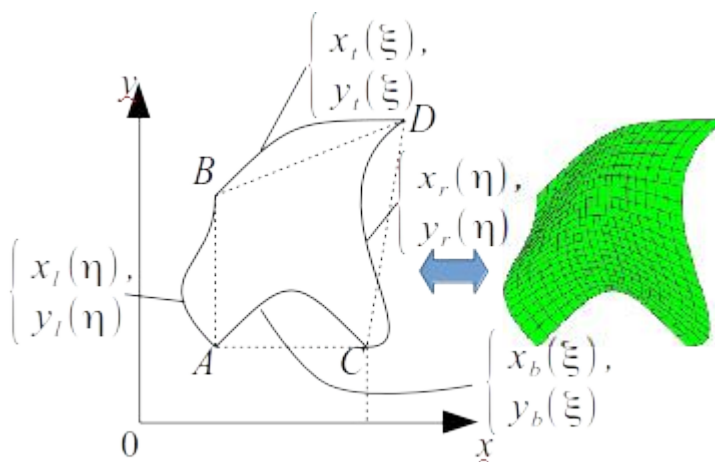


Рисунок 1.5 – Білінійна трансфінитна інтерполяція

Формули, аналогічні (1.1) і (1.2), можна отримати для інтерполяції тривимірних фізичних областей як на основі трілінійної інтерполяції, так і на основі інтерполяції Лагранжа або Ерміта [472]. Методи, що ґрунтуються подібних перетворень, називають алгебраїчними. Один з перших таких методів, що використовує узагальнені криволінійні координати, був запропонований в роботі

[286]. Аналогічні формули для побудови тривимірних структурованих дискретних моделей на базі трансфінитної інтерполяції розглянуті в роботах [249, 289–290].

Для контролю за розподілом внутрішніх вузлів, при побудові дискретних моделей на основі алгебраїчних методів, використовують різні функції розтягування [261] (наприклад, експоненціальні, гіперболічні тощо [472]), а також підходи, що ґрунтуються на методі багатьох поверхонь [154, 472], в яких вводяться додаткові поверхні (у двовимірному випадку – криві) для побудови інтерполяційних схем більш високого порядку.

У роботах [268–270, 356, 482] запропоновані алгебраїчні методи, які ґрунтуються на використанні барицентричних координат, для побудови та оптимізації дискретних моделей. Такі схеми переважно використовують для розбиття на скінченні елементи областей, визначених багатокутниками (наприклад, рис. 1.6) або багатогранниками. Для дискретизації областей, визначених неопуклими багатокутниками, розробляють апарат узагальнених барицентричних координат [356].

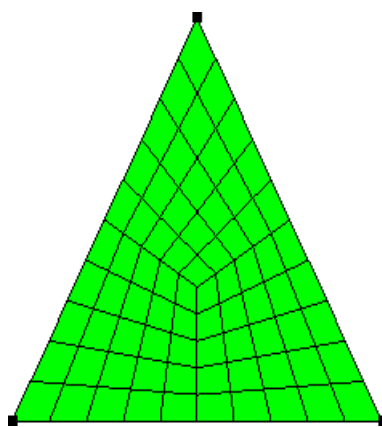


Рисунок 1.6 – Приклад розбиття трикутної області на базі барицентричних координат

Основними недоліками алгебраїчних методів є (у загальному випадку) відсутність ортогональності ребер і складність контролю за розміром сусідніх елементів. Ці факти негативно впливають на точність розв’язків, отриманих з використанням таких дискретних моделей.

Альтернативою алгебраїчних методів для побудови структурованих дискретних моделей є підходи, які ґрунтуються на розв’язанні крайової задачі з

частинними похідними. Такі методи називають диференціальними. Їх у залежності від типу базової системи диференціальних рівнянь можна поділити на еліптичні, параболічні та гіперболічні.

Один з найбільш загальних методів побудови дискретних моделей з використанням диференціальних рівнянь з частинними похідними ґрунтуються на розв'язанні рівняння Пуассона [77–80, 154, 186, 187, 197–199, 335, 472, 493–495, 500], що мають такий вигляд:

$$\begin{aligned}\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} &= P(\xi, \eta), \\ \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} &= Q(\xi, \eta),\end{aligned}\tag{1.3}$$

де P і Q – функції, які використовують для контролю за щільністю вузлів дискретної моделі.

Розв'язок рівняння (1.3) знаходять у розрахунковій області системи координат (ξ, η) , тому їх можна перетворити до вигляду [154]

$$\begin{aligned}g_{22} \frac{\partial^2 x}{\partial \xi^2} - 2g_{12} \frac{\partial^2 x}{\partial \xi \partial \eta} + g_{11} \frac{\partial^2 x}{\partial \eta^2} + g \left(P \frac{\partial x}{\partial \xi} + Q \frac{\partial x}{\partial \eta} \right) &= 0, \\ g_{22} \frac{\partial^2 y}{\partial \xi^2} - 2g_{12} \frac{\partial^2 y}{\partial \xi \partial \eta} + g_{11} \frac{\partial^2 y}{\partial \eta^2} + g \left(P \frac{\partial y}{\partial \xi} + Q \frac{\partial y}{\partial \eta} \right) &= 0,\end{aligned}\tag{1.4}$$

де

$$\begin{aligned}g_{11} &= \left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2, \\ g_{12} &= \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta}, \\ g_{22} &= \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2, \\ g &= \left(\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta} \right)^2.\end{aligned}$$

Рівняння (1.4) розв'язують чисельно, зазвичай, з використанням скінченно-різницевої схеми з крайовими умовами, що відповідають значенням координат вузлів на межі фізичної області. Дискретні моделі, отримані на основі таких систем диференціальних рівнянь, називають еліптичними. Також можна будувати структуровані моделі шляхом розв'язання гіперболічних або параболічних систем [472]. Аналогічні рівняння можна записати для тривимірних тіл (початковими даними будуть положення вузлів на граничних поверхнях) [77–80, 472]. Водночас для нерівномірного початкового розподілу вузлів на межі фізичної області часто використовують алгебраїчні методи для дискретизації кривих [335] і/або двовимірні диференціальні рівняння для дискретизації поверхонь [472].

Однією з найбільш істотних переваг використання рівнянь з частинними похідними є можливість забезпечення ортогональності ліній сітки до межі області [186, 187, 472, 493–495, 500], співпадіння з лініями течії еквівалентної задачі потенційного обтікання [154], а також гладкості внутрішньої частини сітки при наявності розривів меж. У результаті, методи генерації дискретних моделей, які ґрунтуються на рішенні диференціальних рівнянь, набули поширення у задачах дослідження динаміки газів і рідин [8, 77–80, 154, 185].

Описані вище методи застосовні для кубоподібних тіл (у топологічному сенсі). У разі, якщо тіло має форму, яку неможливо відобразити на куб, то використовують методи блокової декомпозиції (рис. 1.7). Основна ідея таких методів – поділ вихідної геометрії об'єкта на кубоподібні області, до кожної з яких застосовуються методи генерації структурованих дискретних моделей.

Один із перших методів блокової декомпозиції ґрунтуються на використанні октодерев і шаблонів для адаптації блоків [490]. Область рекурсивно розбивають на кубоподібні блоки, які в літературі часто називають октантами. Ітерацію по глибині рекурсії виконують, як правило, або в околі особливої точки, або межі з високою кривиною (наприклад, рис. 1.8). На заключному етапі за допомогою спеціальних шаблонів відновлюють коректність топології. Потім будують структуровані сітки для кожного з блоків. Розвиток цієї ідеї висвітлено у [415, 338, 402], узагальнення на тривимірний випадок – у [304, 321, 361, 491, 492].

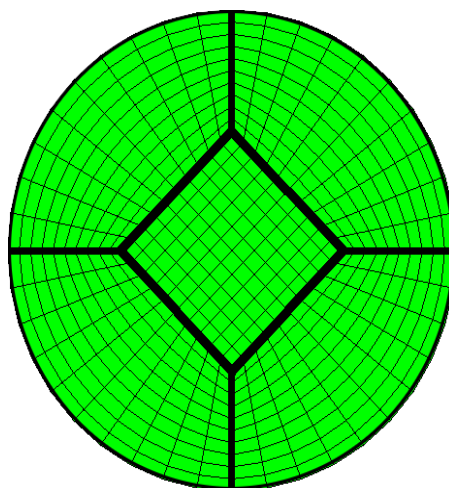


Рисунок 1.7 – Блок-структурована дискретна модель

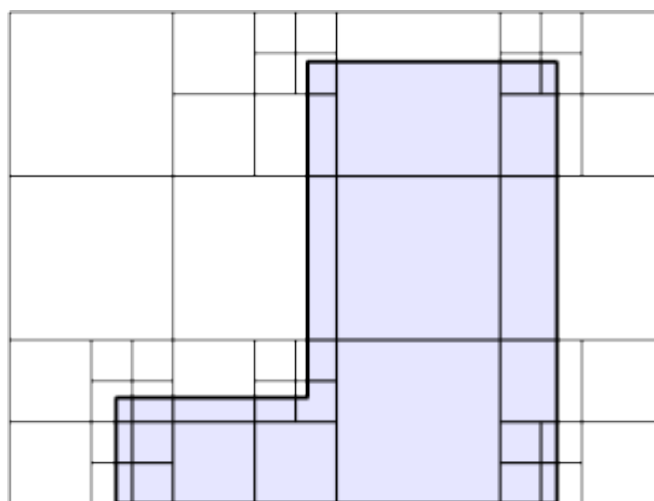


Рисунок 1.8 – Загальна ідея рекурсивного розбиття на блоки

Іншим поширеним способом декомпозиції на блоки є використання методу серединних осей [287, 400, 465]. Серединні осі пов'язані з поняттям «скелет багатокутника» – геометричного місця точок кривої, що утворена центрами вписаних кіл максимального радіуса, який дотикається до меж об'єкта. Вписане коло повинне дотикатись, щонайменше, двох точок межі об'єкта. Інакше, повинно існувати інше вписане коло, що повністю задовольняє цим вимогам. Подання області у вигляді серединних осей є вихідним, розбиття на блоки проводиться в області зламу серединної осі. Цей підхід має узагальнення для тривимірного випадку під назвою серединної поверхні. Варто зазначити, що побудова серединних осей або поверхонь є самостійною досить складною задачею, методи вирішення якої часто використовують триангуляцію Делоне або діаграми Воронова.

Розробляють методи декомпозиції області на блоки на основі нечіткої логіки [458, 459–460], методи на основі загрублення (coarsening) вихідної сітки [371] тощо [306, 497]. Алгоритми на основі нечіткої логіки є вельми перспективними, однак їх практичне застосування ускладнюється труднощами їх реалізації. Методи на основі загрублення як вихідні дані використовують початкову досить густу дискретну модель, в якій блоки формуються шляхом об'єднання сусідніх елементів, що дозволяє їх використовувати для перестроювання моделі за результатами розрахунку.

Однією з переваг блокової декомпозиції є можливість використання технологій паралельних обчислень при обробці блоків (особливості реалізації таких методів розглянуті в роботах [185, 198, 334, 452–453, 367]).

Таким чином, можна виділити ряд переваг і недоліків використання структурованих дискретних моделей форм об'єктів. До переваг можна віднести можливість використання більш високого порядку апроксимації (наприклад, у задачах газодинаміки і дослідження переміщення рідин і газів), можливість використання меншої кількості обчислювальних ресурсів (не потрібно зберігати інформації про сусідні вершини, ребра, грані тощо). Проте, адаптація структурованих або блок-структурованих моделей до складних тіл вельми трудомістка і водночас можлива поява вироджених елементів. Декомпозицію на блоки можна застосовувати при використанні методів твердотілої геометрії для подання форми об'єктів (блоки та їх границі подані явно). Однак, в разі тіл складної форми або при функціональному поданні методи декомпозиції на блоки практично не застосовні (через слабку формалізації блоків або відсутності явного подання границь).

1.3.2 Неструктуровані дискретні моделі форм об'єктів

Методи побудови неструктурованих дискретних моделей можна розділити на дві підгрупи: 1) методи побудови дискретних моделей на базі трикутників і тетраедрів (симплекси); 2) методи побудови дискретних моделей на базі

чотирикутників і шестигранників (топологічні куби). Симплекси набули широкого поширення завдяки їх топологічній гнучкості (для них визначені прості, добре формалізовані процедури вставки або видалення вузла, що дозволяє порівняно легко згущувати або загрублювати моделі).

Використання чотирикутників і шестигранників дозволяє більш точно моделювати об'єкт, що проектується, при використанні меншої кількості вузлів. Водночас генерація дискретних моделей на базі чотирикутних або шестигранних елементів є більш складною у порівнянні з трикутниками або тетраедрами. Це можна проілюструвати на такому прикладі. Нехай дана структурована сітка, в яку необхідно додати вузол, використовуючи тільки локальні перебудовування. Хоч це і можливо в плоскому випадку (рис. 1.9), але у тривимірному просторі не є можливим [472]. Отже, техніка, яку активно використовують для симплексів у методах на основі критерію Делоне [26], практично не застосовна для чотирикутників і шестигранників.

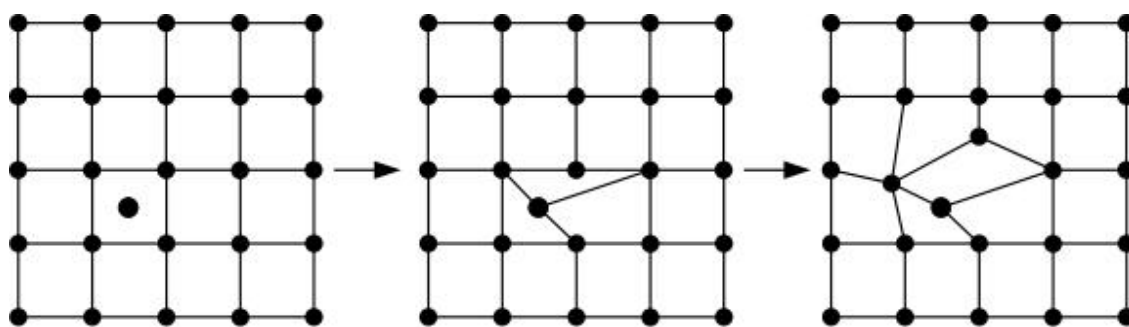


Рисунок 1.9 – Додавання вузла до сітки чотирикутних елементів

1. Методи генерації дискретних моделей на базі трикутників і тетраедрів

Однією з найбільш поширених на практиці є проблема побудови триангуляції вихідної області, придатної для використання в методах скінченних або граничних елементів. Проблема полягає в знаходженні множини трикутників, що покривають вихідну область, форма і розміри яких задовольняють наступним вимогам: не повинно бути трикутників з дуже маленькими і дуже великими кутами, трикутники не повинні бути меншими, ніж це необхідно, і не повинні перевищувати задані розміри. Методи на основі критерію Делоне [26] математично гарантують, що такі

вимоги можуть бути задоволені. Такі ж вимоги, зазвичай, висуваються й до дискретних моделей на базі тетраедрів, для яких математичним обґрунтуванням слугує узагальнення критерію Делоне на випадок n -мірного простору [483–484].

Сьогодні досить добре розвинені методи побудови триангуляції Делоне [26] для вихідної множини точок на площині, детальний аналіз яких можна знайти в роботах О. В. Скворцова [125–127], а також Р. Су і Р. Л. С. Драйсдейла [456]. Розроблені методи від найпростіших – інкрементних алгоритмів, які додають нові вузли в модель один за іншим, до найбільш швидких, які засновані на принципі «розділяй і володарюй» [456]. Вони відносно легко узагальнюються на тривимірний випадок [308–311] (достатньо змінити процедуру перевірки умови Делоне [438]). Ці методи отримали своє застосування в задачах комп'ютерної графіки та геоінформаційних системах. Проте, в задачах проектування потрібна апроксимація геометричних меж об'єкта ребрами і гранями елементів, що не враховується в цих методах.

Також розроблені способи та методи побудови триангуляції Делоне для областей, поданих багатокутниками або планарними графами [208–210, 234–235, 418, 473]. У цій групі методів найбільшого поширення набули модифікації методів Рапперта (J. Ruppert) [418] та Чю (L. Chew) [234–235], які є способами підвищення якості сітки, згенерованої для вихідної області на базі критерію Делоне. Їх узагальненням для тривимірного простору є методи побудови покриття тетраедрами області, обмеженої плоскими гранями [225, 232, 243, 337, 341, 357, 373, 439–441, 474]. В основі таких методів, зазвичай, знаходиться ітераційне додавання нового вузла для локального поліпшення геометричних характеристик дискретної моделі. Новий вузол може додаватися в дискретну модель для усунення елемента з невідповідною якістю в центр описаного кола (рис. 1.10) для трикутника [278] (описаної сфери для тетраедра), або в його цент мас [376]. У деяких роботах [254–255, 258] використовують розбиття Воронова як проміжне. Такі техніки додавання нового вузла не гарантують, що всі ребра та грані вихідного багатогранника будуть апроксимовані, відповідно, ребрами або гранями елементів. Для вирішення цієї проблеми використовують техніки додавання нових вузлів в середини відсутніх ребер або граней [294, 376–377, 418], які іноді комбінують з процедурою дзеркального

відображення (flip) [282, 377]. Проте, в задачах проектування як плоских, так і тривимірних технічних об'єктів складної форми подання їх меж у вигляді планарних графів або плоских граней є досить складною та нетривіальною задачею, що ускладнює практичне використання таких методів.

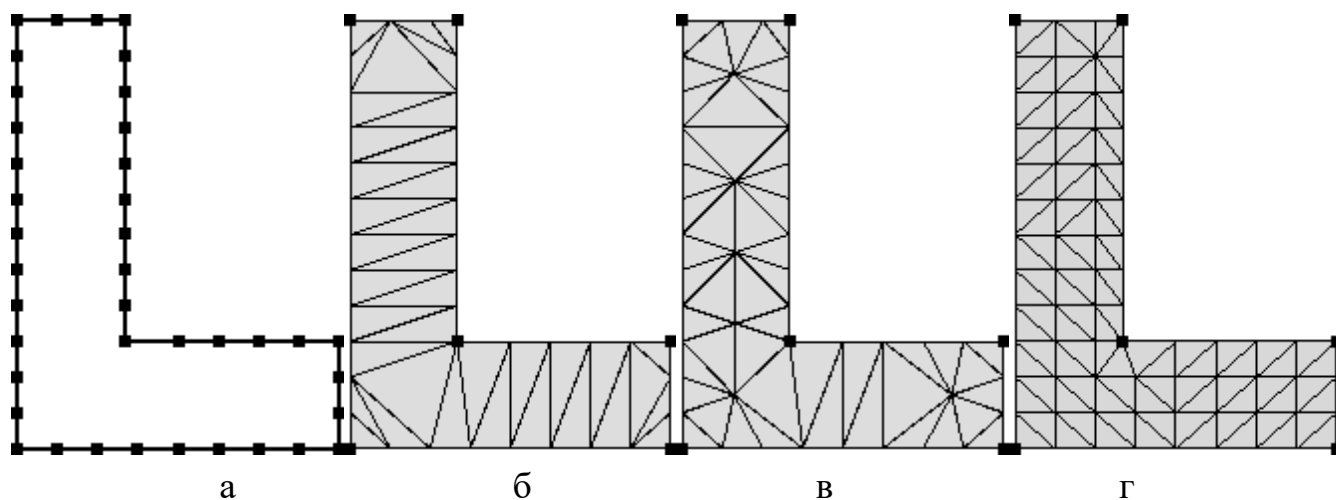


Рисунок 1.10 – Триангуляція L-області:

а – вихідний планарний граф

б – триангуляція Делоне

в – оптимізація мінімального кута й площі додаванням вузлів у центр описаного кола

г – дискретна модель отримана з використанням техніки граничної корекції

Методи рухомого фронту [206, 276–277, 323, 344, 351, 354, 370, 446] ґрунтуються на ідеї пошарового заповнення області елементами. Вихідною інформацією для них є дискретна модель межі області, яка утворює початковий фронт. Рухом фронту формують шари елементів до тих пір, поки вихідна область не буде заповнена повністю. Фронтальні методи запропоновані для генерації моделей як плоских (на базі трикутників) [206, 351, 354, 370, 446], так і тривимірних (на базі тетраедрів) [276–277, 344] об'єктів. У деяких роботах комбінують використання методу на основі критерію Делоне з рухомих фронтом [262, 323, 354]. Фронтальні методи досить універсальні, проте, необхідно відзначити їх високу ресурсомісткість і відносно низьку швидкість роботи. Їх недоліком є проблема самоперетину фронтів.

Також отримання якісного початкового фронту (особливо в тривимірному випадку) є самостійною складною задачею.

Інша група методів генерації дискретних моделей базується на використанні деякої початкової сітки з адаптацією її межових вузлів до меж області [12, 203, 209, 232, 362–363, 374, 437]. Вихідна сітка, зазвичай, є відносно простою (наприклад, структурована сітка). За такої умови накладається єдина вимога: розмір комірок повинен бути меншим, ніж мінімальна геометрична особливість області. На етапі адаптації тут поширене використання граничної корекції вузлів [12] (приклад застосування цього підходу наведено на рис. 1.10, г). Вперше ідея використання вихідної сітки на базі квадратних елементів для рівномірної тріангуляції двовимірної області, обмеженою планарним графом, була висвітлена у роботі [203] (доведено, що мінімальний кут у трикутниках буде не меншим 13° , якщо у вихідному графі немає кутів, гостріших 13°). Для побудови нерівномірних дискретних моделей використовуються адаптивні вихідні сітки на базі методу Quadtree [420, 490–491, 402]. У [209] показано, що при використанні адаптивної вихідної сітки кути в елементах (виключаючи гострі кути між ребрами на межі) будуть більшими за $18,4^\circ$. Цей результат був поліпшений в роботі [374], у якій доведено, якщо в межі області, поданої планарним графом, відсутні гострі кути, то у результаті тріангуляції всі кути елементів будуть знаходитися в межах від 30° до 90° . Даний підхід відносно просто узагальнюється на випадок тривимірного простору [362–363, 491]. Варто відзначити, що зазначені вище роботи використовують дискретну модель межі об'єкта як вхідні дані. Задача її отримання є складною, тут існує можливість отримання вироджених елементів біля межі області.

2. Методи на генерації дискретних моделей на базі чотирикутників і шестигранників

Прямі фронтальні методи генерації неструктурованих сіток чотирикутних елементів [200, 201, 212, 408, 442, 496] будують модель високої якості, адаптовану до границь з малою кількістю нерегулярних елементів. Їх основна ідея полягає у використанні початкової дискретизації межі та подальшому заповненні вихідної

області скінченними елементами шар за шаром. Прямі фронтальні методи також узагальнені на випадок тривимірного простору для отримання неструктурованих сіток шестигранних елементів. В такому випадку область вичерпують пласт за пластом, формуючи таким чином і дискретизацію фронту у вигляді чотирикутних граней, і дискретизацію внутрішньої частини області [213, 450–451]. Основними недоліками прямих фронтальних методів, що призводять до складності їх практичної реалізації та великих обчислювальних витрат, є: 1) трудомісткість пошуку вироджених елементів при перетині двох фронтів; 2) складність запобігання виникненню пустот (ділянок області, незаповнених елементами) [472]. Одночасно, отримання якісної початкової дискретизації меж об'єкта є досить складною (особливо в тривимірному випадку) самостійною задачею.

Непрямі фронтальні методи використовують попередньо побудовану сітку трикутників або тетраедрів. Їх основна ідея – це пошарова трансформація трикутників у чотирикутники [128–129, 141, 382–383] або тетраедрів у шестигранники [384–385]. На відміну від прямих фронтальних методів для них відсутня проблема перетину фронтів, вони захищені від появи порожнеч усередині контуру, але, найчастіше, вони будують сітку топологічно менш якісну, ніж кращі прямі фронтальні методи [128–129]. Серед непрямих фронтальних методів можна виокремити метод Q-Morph, запропонований Стівеном Оуеном (S. Owen) [382–383], та його модифікацію, розроблену А. В. Сковпенем [128–129], яка будує сітку, близьку до оптимальної, з використанням трансформацій трикутників. В результаті узагальнення методу Q-Morph на тривимірний випадок запропонований метод H-Morph [384–385]. Проте, недоліком даного підходу є необхідність наявності якісної попередньої дискретизації на базі трикутників або тетраедрів, отримання якої є окремою трудомісткою задачею.

Техніка просторового вигину континууму (the Spatial Twist Continuum), описана в роботах [272, 368, 470], дозволяє автоматично або напівавтоматично генерувати сітки чотирикутних і шестигранних елементів. В основу техніки генерації дискретних моделей закладено STC-подання, яке є двоїстим для чотирикутників або шестигранників [368, 470], що робить його придатним для

аналізу якості сіток. STC-подання, по суті, є аналогом діаграм Воронова. Воно будується для дискретної моделі межі об'єкта і потім конвертується в дискретну модель на базі чотирикутників або шестигранників. Основним недоліком алгоритмів, що використовують STC-подання, є висока обчислювальна складність: у гіршому випадку вони стають алгоритмічно повними [472]. У деяких роботах [271, 466–467] пропонується використовувати метод малих бічних зсувів (Whisker Weaving), який будує STC-подання на базі рухомого фронту. Недоліки такого підходу аналогічні недолікам прямих фронтальних методів. Загалом, необхідно відзначити, що результатом побудови двоїстого подання в тривимірному випадку не завжди є валідна сітка шестигранних елементів. Також досить важко реалізувати стійку версію алгоритму для методу на основі STC-подання.

Методи суперпозиції ґрунтуються на використанні деякої базової стратегії для побудови дискретних моделей на базі чотирикутників або шестигранників. Алгоритми даного класу, зазвичай, як вихідні дані використовують деяку вихідну сітку, яка більш-менш просто може бути побудована і покриває область навколо геометричного об'єкта. Потім вихідну дискретну модель адаптують до його меж. Необхідно відзначити, що ключовим кроком даної групи методів є складність генерації сітки високої якості біля межі об'єкта (межа має бути апроксимована ребрами елементів, які повинні бути невинродженими), а їх практична реалізація досить проблематична [472]. Існує два основні підходи до адаптації вихідної сітки до межі об'єкта: сітковий [428–430] і проєкційний [305, 457]. Перший підхід ґрунтується на ідеї пошуку точок, розташованих на межі області, відповідних граничним точкам початкової сітки, і подальшому формуванні шару прикордонних елементів за допомогою техніки ізоморфізму [428]. Другий підхід ґрунтується на переміщенні прилеглих вузлів вихідної сітки на межу об'єкта. Також методи на основі суперпозиції можна використовувати для адаптації нерівномірною початкової сітки. Для цього, як і в методах фонові сітки для симплексів, використовують октодерева [304, 338, 350, 429–430]. Спільною особливістю методів на основі суперпозиції є генерація елементів приблизно однакового розміру [472]. Водночас в результаті задоволення вимоги, щодо співвідношення розмірів елементу вихідної

сітки та найменшої геометричної особливості, отримана дискретна модель може бути надмірно густою.

1.3.3 Дискретні моделі об'єктів, форму яких можна подати оболонками

Для моделювання багатьох об'єктів машинобудування (наприклад, елементів ракетноносіїв, літаків, морських суден, автомобілів тощо) використовують оболонки. Оболонка – форма тіла, у якого один з розмірів (товщина) значно менше інших. Геометрично оболонку, зазвичай, подають серединною поверхнею. Серединні поверхні, зазвичай, задають за допомогою параметричних або неявних функцій.

Дискретні моделі оболонкових об'єктів машинобудування, зазвичай, використовують трикутні або чотирикутні оболонкові елементи. Такий підхід має кілька переваг відносно використання тривимірних елементів (солідів: тетраедрів або шестигранників): по-перше, економія обчислювальних ресурсів за рахунок використання меншої кількості елементів, по-друге, при використанні тонких тривимірних елементів виникають великі коефіцієнти в матриці жорсткості [504] в позиціях напрямів, відповідних товщині оболонки, що негативно впливає на точність моделювання.

Важливим є той факт, що у ряді методів побудови дискретних моделей на базі тетраедрів [276–277, 344] або шестигранників [213, 450–451] використовують моделі межі на базі, відповідно, трикутників або чотирикутників як вихідні.

При параметричному поданні поширена ідея побудови множин вузлів та елементів в двовимірному параметричному просторі з подальшим відображенням на поверхню. Проте, якісна дискретна модель, отримана в двовимірному параметричному просторі, не обов'язково буде відповідати якісній моделі, що складається з елементів на поверхні.

1. Методи побудови дискретних моделей поверхонь з використанням трикутних елементів

В роботах [498–499] розглянуті особливості використання відображень для побудови дискретних моделей з використанням трикутників для повністю складених з лінійних і квадратичних форм або B-сплайнів параметричних поверхонь. Для покращення якості дискретних моделей запропоновані техніки перестановки вузлів (swar) і згладжування Лапласа.

Також розроблені методи, які ґрунтуються на використанні рухомого фронту [328, 358–360]. Як і в двовимірному випадку, вихідними даними є дискретна модель меж (ребер) поверхні, внутрішня частина якої заповнюється шар за шаром. У роботах [358–360] запропоновані метрики для управління процесом побудови фронту та схема використання фонові сітки на базі методу Quadtree для поліпшення якості дискретної моделі. Схожа з рухомим фронтом ідея заповнення поверхні «біжучими» (marching) трикутниками з попереднім перетворенням подання поверхні з параметричного в неявне розглянута в [293]. Недоліки методів на основі рухомого фронту для поверхонь такі самі як у відповідних методів для суцільних областей.

Ряд робіт ґрунтується на ідеї адаптації методу Рапперта [418] для тріангуляції поверхні на базі критерію Делоне [26]. Ключовим тут є спосіб перевірки виконання умови потрапляння в описане коло для тривимірного трикутника та відображення його центру в параметричний простір. Одна з перших таких робіт [236] заснована на безпосередній перевірці виконання попадання в описану окружність в 3D. Така перевірка вельми екстенсивна з обчислювальної точки зору, тому часто використовують різні схеми апроксимації. Наприклад, у [228] запропонована схема використання основних кривин для апроксимації описаного кола поверхневого трикутника двовимірним еліпсом в параметричному просторі; у [281] запропонований підхід, заснований на використанні сфер.

Альтернативним параметричному поданням поверхонь, зазвичай, є неявне. Методи побудови тріангуляції поверхонь, визначених неявними функціями, можна умовно розділити на три групи: коміркові [345], біжучих трикутників [188] і методи на основі тесселяції [250].

2. Методи побудови дискретних моделей поверхонь з використанням чотирикутних елементів

Існують два основні підходи до генерації дискретних моделей на базі чотирикутників для оболонкових конструкцій: прямий і непрямий. Як і в двовимірному випадку, прямі схеми відразу заповнюють область чотирикутниками, а непрямі використовують попередньо побудовану дискретну модель на базі трикутників.

Історично першими прямими до дискретизації поверхні чотирикутниками є техніки на базі відображень [237, 246, 486, 502]. Елементи генерують для фізичної області шляхом відображення рівномірної або близькою до рівномірної сітки, побудованої в параметричній площині, на криволінійну поверхню. Такі техніки дозволяють отримати структуровані дискретні моделі високої якості. Однак, застосовні вони для відносно простих областей. Складні багатозв'язні області повинні бути розбиті на блоки [342]. Проте, можливості автоматичного застосування таких технік залишається вельми обмеженими, а обчислювальні ресурси, необхідні для побудови якісної дискретної моделі складної області, потрібні значні.

Альтернативна техніка генерації дискретних моделей криволінійних поверхонь з використанням чотирикутників заснована на методі рекурсивної декомпозиції [461]. Також розроблені прямі фронтальні методи (наприклад, [212]), які будують модель поверхні шар за шаром.

У непрямих методах чотирикутники формують, конвертуючи елементи попередньо побудованої моделі на базі трикутників за допомогою операцій злиття або розщеплення. Найпростіша схема для побудови множини чотирикутників (без гарантій повної відсутності трикутників у результаті), шляхом видалення діагоналей між суміжними трикутниками запропонована в [343]. Вона була розвинута в [333] до гарантованої відсутності трикутників у результаті для довільної області. Аналогічний підхід до конвертації трикутників у чотирикутники використаний у [312]. Техніка одночасної генерації трикутників та їх злиття у чотирикутники запропонована у [200]. З огляду на те, що непрямі техніки використовують вихідну сітку трикутників, порівняно нескладно забезпечити нерівномірну щільність вузлів

моделі (наприклад, [328–329]). Проте, результат роботи непрямих методів – менш якісні дискретні моделі (відносно кращих прямих).

1.4 Аналіз методів оптимізації дискретних моделей форм об'єктів

Точність результатів, яку дозволяють отримати САПР машинобудування, у тому числі, залежить від якості дискретних моделей форм об'єктів. При розробці методів поліпшення якості дискретних моделей найбільш часто використовуються дві характеристики: якість форми елемента (близькість до правильної форми, тобто, відсутність дуже гострих кутів, співвідношення довжин сторін близьке до одиниці тощо) та/або близькість сітки до рівномірної.

1.4.1 Методи оптимізації внутрішніх вузлів та елементів

Найбільшого поширення набули підходи, які ґрунтуються на згладжуванні Лапласа [266], що базується на ідеї переміщення оброблюваного вузла у центр мас багатокутника, утвореного сусідніми вузлами (рис. 1.11). Водночас актуальні напрямки досліджень у даній групі методів базуються на введенні системи вагових коефіцієнтів або контрольних функцій, заснованих, наприклад, на значеннях кутів елемента [501], спеціальних інтегральних функцій [226] тощо [212, 292, 200].

Альтернативою згладжування Лапласа є підходи, засновані на введенні цільової функції, оптимізація якої забезпечить найкраще розташування вузлів сітки. Наприклад, в якості цільової функції можуть виступати значення мінімального або максимального кута елемента [221, 274–275], співвідношення радіуса описаного кола до радіусу вписаного (для трикутників та тетраедрів) [391], площі елементів, довжини сторін або скалярні добутки векторів, визначених на сторонах елементів [205, 222, 317–318, 471]. У цій групі можна виділити методи, засновані на вирішенні

локальної (подібно згладжування Лапласа переміщається тільки один вузол) [192, 275] і глобальної (мінімізації підлягає функція координат всіх внутрішніх вузлів) [205, 222, 317–318, 471] оптимізаційних задач.

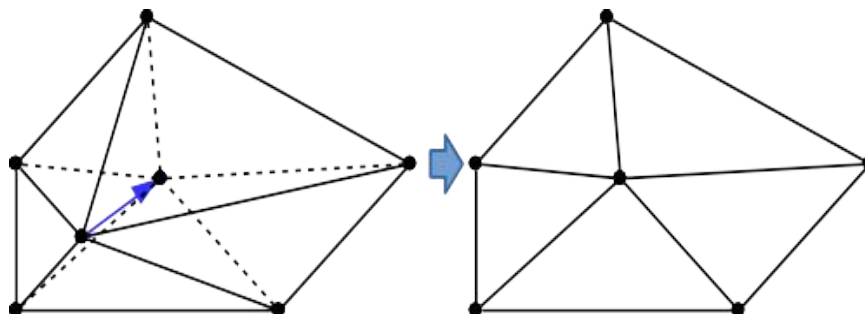


Рисунок 1.11 – Згладжування Лапласа

Перевага методів, які ґрунтуються на оптимізації, полягає в гарантуванні поліпшення якості сітки. Проте, їх реалізація вимагає розв’язання задачі оптимізації з числом змінних вдвічі або втричі більшим, ніж кількість вузлів сітки. Час, витрачений на виконання такої процедури, може у 30–40 разів перевищувати час, необхідний на виконання згладжування Лапласа [273].

Окремо розробляють гібридні методи [221, 230, 274], які ґрунтуються на комбінуванні оптимізаційних методів і згладжування Лапласа.

Описані вище методи, які ґрунтуються на оптимізації геометричних або топологічних характеристик дискретної моделі. Однак, в деяких задачах, наприклад, задачах побудови дискретних моделей для дослідження напружено-деформованого стану пластин, які перебувають під дією навантаження деякої форми, виникає необхідність управління процесом поліпшення сітки. Для підвищення точності інженерного аналізу необхідно оптимізувати також і геометричні характеристики елементів моделей, і розподіл їх щільності у залежності від фізичної постановки. Так, наприклад, в праці [316] розглянута схема оптимальної побудови дискретної моделі для чисельного розв’язання рівняння Пуассона-Больцмана. В [194] розглянута схема оптимізації дискретної моделі на базі чотирикутників з керуванням спеціальною функцією.

1.4.2 Методи оптимізації межових вузлів та елементів

Методи оптимізації положень межових вузлів та елементів мають одночасно найбільші складність та практичний інтерес. Перші праці у цьому напрямку пов'язані з методом крокуючих кубиків (marching cubes) [345], в якому для швидкої візуалізації поверхонь тривимірних геометричних об'єктів, заданих неявними функціями, будуються трикутні елементи за шаблонами значень у вузлах рівномірної сітки.

Для пошуку особливостей поверхонь тривимірних геометричних об'єктів (наприклад, значних змін кривини) у праці [265] запропонована модифікація методу крокуючих кубів, яка ґрунтується на аналізі градієнтів функції відстані та вставці нових вузлів. У працях [379–380] запропоновані підходи до поліпшення результатів роботи методу крокуючих кубиків з використанням дуальних сіток [379] або метрик [380], що враховують відхилення вузлів від поверхні та нормалей трикутників від градієнтів неявної функції. У праці [298] для оптимізації початкової тріангуляції запропоновано мінімізувати енергетичний функціонал при пошуку положень вузлів та зменшенні їх кількості. У праці [302] спільно з енергетичним функціоналом запропоновано використовувати неявну інтерполяцію поверхні.

Точніше подання особливостей можна отримати за допомогою адаптивних дискретних моделей. Наприклад, у праці [487] запропоновано використовувати ієрархічні структури (octrees) для прискорення роботи методів, заснованих на покроковому скануванні областей, а у працях [358, 436, 444] для збільшення кількості вузлів в областях з найбільшою кривиною поверхні. Ієрархічні структури також можуть використовуватися спільно з аналізом перетинів нормалей з поверхнею [313]. Використання ієрархічних структур для оптимізації дискретних моделей при адаптивному скінченно-елементному аналізі показано в роботі [360]. У праці [218] розглянуті особливості реалізації ієрархічних структур для оптимізації моделей з використанням графічного процесора.

Альтернативою сканування області елементарними об'ємами (кубами або тетраедрами) є сканування поверхні плоскими елементами. Наприклад, в методі крокуючих трикутників [297] поверхню заповнюють трикутниками, почавши рух з довільної її точки. У працях [188, 223] наведені адаптивні схеми для сканування неявних поверхонь трикутниками. Схожі ідеї реалізовані у фронтальному методі [293].

У працях [234–236, 253] запропоновано використовувати критерій Делоне з обмеженнями при генерації або оптимізації координат трикутників на поверхні. У роботі [263] схема з критерієм Делоне застосована для адаптивної тріангуляції поверхонь з урахуванням матеріалів (складених об'єктів).

У праці [300] запропонований підхід до пошуку особливостей поверхонь, який ґрунтується на сегментації та кластеризації при оптимізації мереж трикутних елементів. Для розв'язання схожої задачі в праці [366] використовують семантичну оптимізацію.

1.5 Аналіз методів подання форм об'єктів

Реалізація підходів і технік генерації дискретних моделей форм об'єктів залежить від схеми подання останніх. Під схемою подання множини форм об'єктів $\mathbf{O}$ множиною $\mathbf{L}$ розуміють відношення r , яке встановлює відповідність між $\mathbf{O}$ і $\mathbf{L}$ [184]. Якщо $\mathbf{O}$ – множина всіх можливих форм об'єктів, а $\mathbf{L}$ – множина формальних описів моделей геометричних об'єктів, то схема подання r встановлює відношення між формальним описом і формою об'єкта. Схему подання можна застосовувати у САПР машинобудування, якщо відношення r , що реалізує схему подання, є бієкцією.

У дослідженнях з комп'ютерного моделювання форм об'єктів, виконаних в роботах А. G. Requicha [410–413], виділені вісім основних схем подання.

1. Інженерні креслення – вельми поширений інструмент «спілкування» між інженерами шляхом подання об'єкта у вигляді набору двовимірних проекцій. Основним їх недоліком є можливість неоднозначного трактування деяких елементів, що призводить до складності їх формалізації.

2. Каркасне подання [353, 479] використовувалося в ранніх САПР машинобудування для візуалізації об'єктів їх ребрами (каркасом). Воно допускає неоднозначність інтерпретації, отже, його застосування обмежене.

3. Подання екземплярами примітивів засноване на формуванні колекції параметризованих моделей форм об'єктів. Кожен примітив власною назвою та списком параметрів подає сімейство форм. Основний недолік такого підходу – неможливість формування нових примітивів з існуючих.

4. Воксельне подання базується на перерахуванні кубічних комірок, які повністю належать об'єкту. Це подання застосовується архітектурі, де об'єкти часто мають блокову структуру [349]. Проте, загалом, воно подає форму з деякою точністю, яка збільшується зі зменшенням розміру комірки, але водночас зростає вимога до обчислювальних ресурсів.

5. Дискретні моделі подають форми об'єктів множинами простих фігур, які не перетинаються і можуть мати спільні вершини, ребра та грані. Поверхня об'єкта апроксимується гранями елементів. Основні області застосування цього подання – візуалізація і чисельний аналіз стану об'єктів.

6. Конструктивна суцільна (блокова) геометрія [229, 291, 407, 409–413, 416] (Constructive Solid Geometry, CSG) об'єднує сімейство схем подання твердих тіл як результат булевих операцій для комбінування примітивів або суцільних блоків. Семантично модель тіла розглядається як дерево теоретико-множинних операцій (об'єднання, перетин, різниця) і примітивів, поданих суцільними підмножинами евклідового простору. Примітивами, зазвичай, є найпростіші геометричні фігури (куб, циліндр, призма, піраміда, сфера, конус), що ускладнює їх застосування для моделювання нестандартних тіл.

7. Схеми подання розгорткою засновані на простому спостереженні, що рух плоского множини через простір може утворити «об'єм», що подається рухомим об'єктом і траєкторією його руху. Найчастіше так отримують тіла обертання або призми з різними перетинами. Такі техніки вельми наочні, однак їх застосування обмежене.

8. Граничне подання (Brep) [322, 386, 401, 454, 478] розглядає форму об'єкта як композицію його меж. Межі подають прямолінійними (наприклад, трикутниками) або криволінійними ділянками (наприклад, поверхнями Безьє), які називають гранями. Воно дуже поширене в комп'ютерній графіці, а також використовується для чисельного дослідження оболонкових конструкцій. Проте, ручна побудова таких моделей є досить трудомісткою.

Аналіз наведених вище «класичних» подань показав, що їх практичне застосування обмежене або вимагає значних зусиль для побудови моделей складних об'єктів складної форми. З точки зору універсальності одним з найбільш перспективних виглядає функціональне подання, яке ґрунтується на використанні мови неявних математичних функцій [114–123, 69–76, 215, 392–393, 434]. Проте, його використання інженерами вельми важке з точки зору побудови та верифікації моделі.

Одним з поширених способів опису моделей об'єктів зі складною формою в САПР машинобудування є використання проблемно-орієнтованих мов. Наприклад, HyperFun [345], PLaSM [387–388], BDRY [489], QMSH [283] тощо [9, 15–20]. Такі мови, зазвичай, дозволяють будувати моделі конструюючи їх з примітивів або шляхом опису поверхонь об'єкта за допомогою полігональних граней.

1.6 Постановка мети та задач дослідження

Аналіз літературних джерел, присвяченим різним аспектам моделювання форм об'єктів у САПР машинобудування, показав, що на сьогодні накопичено значний досвід у вирішенні проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного на дослідження їх стану. Водночас основні результати отримані в процесі розв'язання окремих задач проблеми: побудови неперервної математичної моделі форми об'єкта; генерації дискретної моделі, що адекватно подає форму; оптимізації положень вузлів і топології елементів дискретної моделі; скінченно-елементного аналізу на базі дискретної моделі.

Побудова математичних моделей для дослідження стану об'єктів машинобудування пов'язана з необхідністю опису їх форм. Для цього використовують різні подання форм об'єктів: граничне, конструктивна блокова геометрія, функціональне тощо. Аналіз показав, що функціональне подання, що засноване на використанні неявних функцій, є найбільш універсальним і теоретично дозволяє описати форму об'єкта довільної складності. Неявний опис форми об'єкта дозволяє класифікувати положення точки простору відносно його меж, але потребує розробки методів генерації дискретних моделей як множин простих фігур. Водночас розроблені методи генерації дискретних моделей для форм [55–58], заданих функціонально, не дозволяють отримувати моделі для чисельного моделювання стану об'єкта із заданою точністю.

Отже, незважаючи на чисельні публікації, присвячені розв'язанню задач моделювання й аналізу форм, виявлено протиріччя між необхідністю побудови дискретних моделей форм об'єктів, які дозволять із заданою точністю виконувати чисельний аналіз стану об'єктів у САПР машинобудування, та обмеженістю методів, що існують.

Враховуючи вище зазначене, метою дисертаційної роботи є розробка методів математичного моделювання та аналізу форм об'єктів у САПР машинобудування

Для досягнення поставленої мети у дисертаційній роботі необхідно розв'язати такі основні задачі:

- виконати огляд і аналіз сучасного стану проблеми;
- розробити гібридний підхід до подання форм об'єктів машинобудування з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання);
- розробити проекційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально;
- розробити метод згладжування координат вузлів дискретних моделей меж об'єктів;
- модифікувати метод локальних перебудов елементів дискретних моделей на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об'єктів;

- розробити метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально;
- розробити метод згладжування координат внутрішніх вузлів дискретних моделей форм об'єктів;
- розробити метод генерації адаптивних дискретних моделей форм об'єктів (adaptive meshing), заданих функціонально, для забезпечення скінченно-елементного аналізу поведінки об'єкта із заданою точністю;
- розробити методи генерації дискретних моделей форм об'єктів і скінченно-елементного аналізу з використанням технологій паралельних обчислень;
- розробити проблемно-орієнтовану мову для опису об'єктів машинобудування складної форми з використанням гібридного підходу, а також задач побудови дискретних моделей та скінченно-елементного аналізу;
- розробити САПР для апробації розроблених методів при дослідженні поведінки об'єктів машинобудування.

Висновки до розділу 1

У сучасних будівництві, авіа-, машино-, судно- і ракетобудуванні існує потреба у зменшенні собівартості продукції. Для цього при проектуванні використовують методи математичного моделювання, які дозволяють замінити вартісний натурний експеримент дослідженням математичної моделі об'єкта. Водночас поширене використання систем диференціальних рівнянь в частинних похідних, які не мають аналітичного розв'язку. У результаті, на практиці застосовують чисельні методи, засновані на ідеї переходу від неперервної моделі форми об'єкта до дискретної.

Першим етапом чисельного дослідження стану об'єкта є опис його форми. Далі виконується перехід до дискретної моделі, яка повинна не тільки адекватно відображувати форму об'єкта, а й дозволяти розв'язання вихідної системи диференціальних рівнянь із заданою точністю.

Більшість публікацій, присвячених побудові дискретних моделей, враховують лише геометричні характеристики об'єктів та не дозволяють будувати адаптивні дискретні моделі для форм об'єктів при функціональному підході, який виявлено як найбільш універсальний.

У результаті аналізу проблеми виявлено протиріччя між необхідністю побудови дискретних моделей форм об'єктів, які дозволять із заданою точністю виконувати чисельний аналіз стану об'єктів у САПР машинобудування, та обмеженістю існуючих методів.

На основі виконаного аналізу предметної області сформульовані мета і задачі дослідження.

Основні результати першого розділу опубліковано у роботах [22, 89, 88, 182].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилення [1, 4, 5, 6, 9, 12, 15–20, 26–35, 39, 52–58, 68–80, 85, 87, 90, 93, 95–96, 99, 102, 114–129, 135, 141, 154, 183–188, 194–199, 200–203, 205, 206, 208–210, 212, 213, 215, 216, 218, 221–223, 225, 228–230, 232, 234–237, 243, 246, 247, 249, 250, 253–256, 258, 260, 262, 286, 261, 263, 265, 268–271, 274–278, 281–283, 287–298, 300, 302, 304–306, 308–311, 313, 314, 316–319, 321–323, 325–328, 330, 332–347, 349–368, 370, 371, 373, 376, 377, 379–380, 382–388, 392–394, 398–402, 405, 407–413, 415, 416, 418, 420–426, 428–430, 434, 436–442, 444, 446, 450–453, 455–467, 468, 470–474, 476–479, 481–484, 487, 489–500, 502–505].

2 ПОДАННЯ ФОРМ ОБ'ЄКТІВ У САПР МАШИНОБУДУВАННЯ

Першим етапом при чисельному дослідженні стану об'єкта машинобудування є опис його форми. Типова задача цього етапу є такою. Дано: креслення або опис об'єкта. Необхідно побудувати математичну модель форми об'єкта. При розв'язанні такої задачі, можна виокремити два основних класи форм об'єктів: суцільні області (або тіла) та оболонки. Суцільні тіла – це форми, які можна розглядати як замкнені та обмежені підмножини евклідова простору. Оболонки – суцільні тіла, у яких один з розмірів (товщина) значно менший за інші. У результаті, при розробці подання цих форм об'єктів у САПР необхідно враховувати їх особливості.

2.1 Подання форм об'єктів на базі неявних функцій

З математичної точки зору тривимірне тіло – це підмножина тривимірного простору $\mathbf{R}^3$. В контексті автоматизації такі підмножини повинні бути замкнутими, обмеженими та регулярними в $\mathbf{R}^3$. Їх у роботі [410] називають g -множинами. Множини називають регулярними, якщо результат об'єднання, перетину або різниці будь-яких двох регулярних множин є регулярною множиною (порожня множина також вважається регулярною).

Для подання призматичних тіл зі сталою товщиною (наприклад, при дослідженні плоского деформованого стану) використовують двовимірні області, які, за аналогією з тривимірним випадком, моделюються замкнутими, обмеженими і регулярними підмножинами в $\mathbf{R}^2$.

При автоматизації g -множини можна подати явно (перерахуванням всіх точок, що їй належать) або неявно (формуванням умови для перевірки приналежності точки до неї). Явно описати g -множину можна лише в найпростіших випадках, тому неявний опис більш поширений.

Неявно r -множини можна визначити різними способами: 1) перерахуванням умов належності точки множині; 2) формуванням системи нерівностей або рівнянь, коренями яких збігаються з множиною; 3) формуванням єдиного рівняння або нерівності, корені якого збігаються з множиною. Наприклад, двовимірну квадратну область (рис. 2.1) при першому способі можна описати як область, розташовану між прямими $x=-1$, $y=1$, $x=1$ і $y=-1$. Її також можна подати системою нерівностей

$$\begin{cases} 1+x \geq 0, \\ 1-x \geq 0, \\ 1+y \geq 0, \\ 1-y \geq 0, \end{cases}$$

або однією нерівністю $2-x^2-y^2-\sqrt{(1-x^2)^2+(1-y^2)^2} \geq 0$ (рис. 2.2, а). Проте, при перших двох способах для опису складних областей доведеться формувати композиції описів або системи систем, що є серйозним конструктивним недоліком, що виявляється в багатьох ситуаціях, де потрібно враховувати геометричну інформацію на аналітичному рівні [46, с. 65]. Отже, опис множини єдиним рівнянням або нерівністю найбільш перспективний.

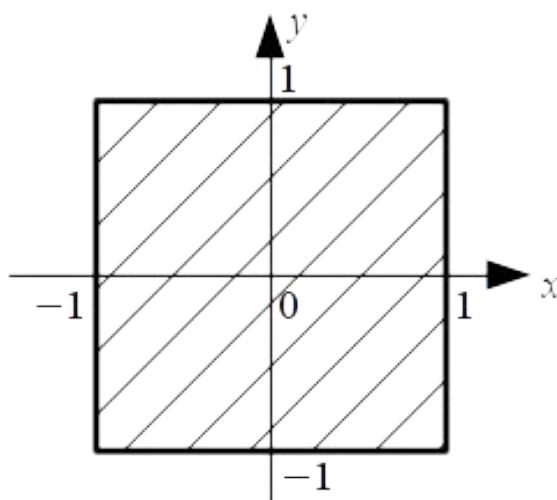


Рисунок 2.1 – Двовимірна квадратна область

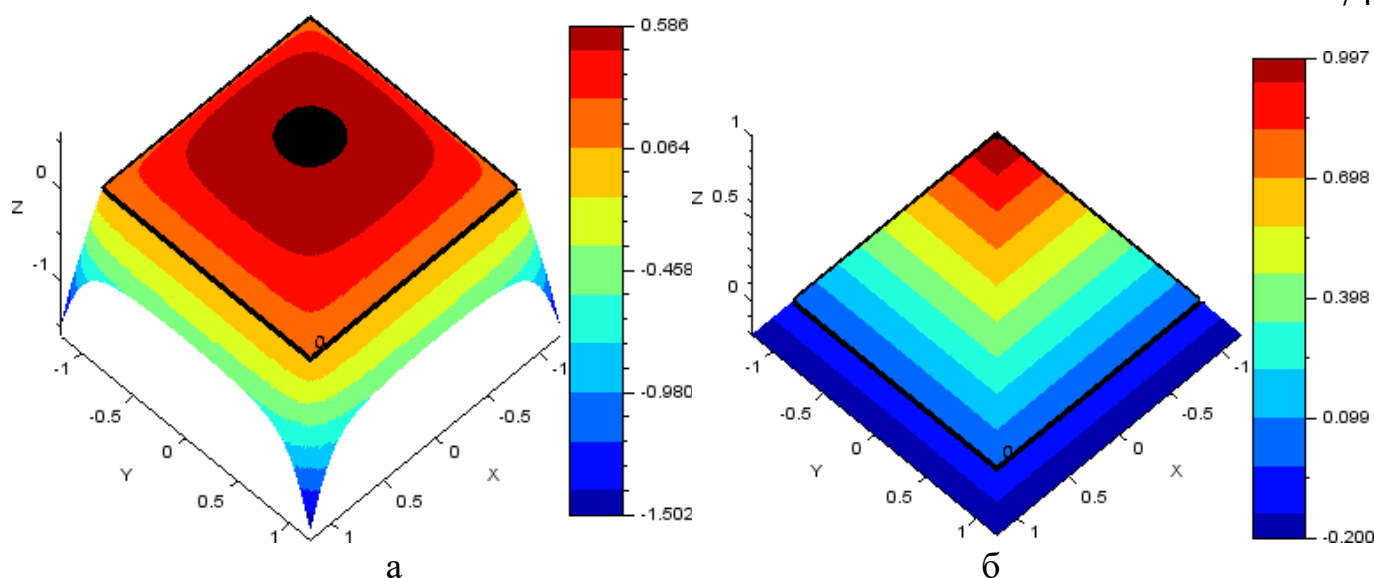


Рисунок 2.2 – Розподіл значень функцій, що подають квадратну область:

$$a - F(x, y) = 2 - x^2 - y^2 - \sqrt{(1-x^2)^2 + (1-y^2)^2}$$

$$б - Q(x, y) = \min(1+x, 1-x, 1+y, 1-y)$$

Нехай деяка дійсна функція $F(P) \geq 0$ для довільної точки P підмножини Ω евклідового простору, що відповідає геометричній формі об'єкта. В результаті, отримаємо неявне подання множини у вигляді

$$\Omega = \{P \mid F(P) \geq 0\}. \quad (2.1)$$

Відповідно, доповнення множини Ω – множина $\overline{\Omega}$, буде мати вигляд

$$\overline{\Omega} = \{P \mid -F(P) \geq 0\}. \quad (2.2)$$

Функцію $F(P)$ можна побудувати як суперпозиція функцій $f_i(P) \geq 0$, які відповідають примітивним формам. Найбільш прості форми (півпростори, півплощини, циліндричні об'єкти тощо) можна подати багаточленами, елементарними функціями та константами. Проте, в інженерній практиці необхідно описувати досить складні геометричні форми об'єктів, що найбільш просто здійснювати з використанням теоретико-множинних операцій об'єднання, перетину та доповнення.

Наприклад, можна побачити, що квадрат є перетином множин, заданих нерівностями $f_1(x, y)=1+x \geq 0$, $f_2(x, y)=1-x \geq 0$, $f_3(x, y)=1+y \geq 0$, $f_4(x, y)=1-y \geq 0$. Цю область можна подати єдиною нерівністю (рис. 2.2, б):

$$Q(x, y) = \min(f_1(x, y), f_2(x, y), f_3(x, y), f_4(x, y)) \geq 0,$$

де функція знаходження мінімуму $\min$, очевидно, відповідає операції перетину множин.

Використання функції $\min$ у даному випадку обумовлено спостереженням, що вона невід'ємна тоді і тільки тоді, коли її обидва аргументи невід'ємні. Тобто, якщо $\Omega_1 = \{P \mid f_1(P) \geq 0\}$ і $\Omega_2 = \{P \mid f_2(P) \geq 0\}$, тоді

$$\Omega_1 \cap \Omega_2 = \{P \mid \min(f_1(P), f_2(P)) \geq 0\}. \quad (2.3)$$

аналогічно функція $\max$ буде невід'ємною, якщо хоча б один її аргумент невід'ємний, що дозволяє використовувати її для подання об'єднання множин:

$$\Omega_1 \cup \Omega_2 = \{P \mid \max(f_1(P), f_2(P)) \geq 0\}. \quad (2.4)$$

Множини, які можна подати як скінченну комбінацію теоретико-множинних операцій над множинами виду (2.1), називають напіваналітичними [410, 416, 434]. Такі множини активно використовуються в комп'ютерній графіці й автоматизованому проектуванні.

Використання функцій $\min$ і $\max$ для опису об'єднання та перетину форм об'єктів, поданих функціонально, вперше було запропоновано в роботі [414]. Їх використання дозволяє описувати форми об'єктів шляхом логічного конструювання, але не дозволяє визначити диференціальні характеристики межі множини, що відповідає формі об'єкта.

Отже, функцію $F(P)$, що відповідає множини Ω , можна побудувати різними способами. Теорія R-функцій академіка Рвачова В. Л. [46, 114–123] пояснює,

систематизує й узагальнює математичний апарат, необхідний для побудови теоретично необмеженої множини форм і відповідних їм функцій. Але залишається актуальною задача спрощення процесів побудови і верифікації функцій, що подають форми складних об'єктів.

З точки зору моделювання форм об'єктів, основне призначення теорії R-функцій – це заміна логічних або теоретико-множинних операцій відповідними їм дійсними функціями при побудові неявного подання $F(P) \geq 0$ для форми довільного об'єкта. Такі дійсні функції дійсних аргументів називають логічно змінними (їх знак повністю визначається знаками аргументів і не залежить від величини значень аргументів).

Хоча в роботах Рвачова В. Л. [46, 114–123] показана принципова можливість побудови дійсних функцій (R-відображень), відповідних до k-значної логіки, на практиці, зазвичай, досить функцій двозначної логіки. У такому випадку дійсна функція $F: \mathbf{R}^n \rightarrow \mathbf{R}$ називається R-функцією, якщо існує функція двозначної логіки $\Phi: \mathbf{B}_2^n \rightarrow \mathbf{B}_2$, яка разом з F утворює комутативну діаграму

$$\begin{array}{ccc} \mathbf{R}^n & \xrightarrow{F} & \mathbf{R} \\ S_2^n \downarrow & & \downarrow S_2, \\ \mathbf{B}_2^n & \xrightarrow{\Phi} & \mathbf{B}_2 \end{array}$$

де $\mathbf{B}_2 = \{0, 1\}$;

$S_2: \mathbf{R} \rightarrow \mathbf{B}_2$ – сюр'єкція, яку можна визначити функцією Гевісайда [434]:

$$S_2(x) = \begin{cases} 0, & x \leq -0, \\ 1, & x \geq +0. \end{cases}$$

Функція двозначної логіки $\Phi: \mathbf{B}_2^n \rightarrow \mathbf{B}_2$, яка супроводжує, R-функцію $F: \mathbf{R}^n \rightarrow \mathbf{R}$. Функцію Φ , зазвичай, визначають з використанням логічних операцій: $\wedge$ (кон'юнкція), $\vee$ (диз'юнкція) і $\neg$ (заперечення) над n логічними змінними. Ці три

логічні операції утворюють повну (хоча не мінімальну) систему в множині булевих функцій.

Основні властивості R-функцій, серед яких замкнутість щодо композиції, доведені в роботі [116]. Множина R-функцій нескінченна. Проте, для додатків немає необхідності знати всі R-функції, а досить мати можливість конструювати їх.

Функція $F: \mathbf{R}^n \rightarrow \mathbf{R}$ є неявною функцією та, отже, її можна використовувати для подання форм об'єктів r -множинами. У більшості випадків для побудови F використовують систему R-функцій

$$\begin{cases} \neg x = -x, \\ x \wedge_{\alpha} y = \frac{1}{1+\alpha} (x+y-\sqrt{x^2+y^2-2\alpha xy}), \\ x \vee_{\alpha} y = \frac{1}{1+\alpha} (x+y+\sqrt{x^2+y^2-2\alpha xy}), \end{cases} \quad (2.5)$$

де x і y – значення деяких неявних функцій;

$$\alpha = \alpha(x, y), \quad -1 \leq \alpha \leq 1.$$

Найбільш часто обирають $\alpha=0$; 1; 0,9 або $\alpha(x, y) = (1+x^2+y^2)^{-1}$ [70].

Отже, при використанні системи R-функцій (2.5) теоретико-множинні операції об'єднання (2.3) і перетину (2.4) отримають вигляд

$$\begin{aligned} \Omega_1 \cup \Omega_2 &= \{P \mid f_1(P) \vee_{\alpha} f_2(P) \geq 0\} = \\ &= \left\{P \mid \frac{1}{1+\alpha} (f_1(P) + f_2(P) + \sqrt{f_1^2(P) + f_2^2(P) - 2\alpha f_1(P) f_2(P)}) \geq 0\right\}, \\ \Omega_1 \cap \Omega_2 &= \{P \mid f_1(P) \wedge_{\alpha} f_2(P) \geq 0\} = \\ &= \left\{P \mid \frac{1}{1+\alpha} (f_1(P) + f_2(P) - \sqrt{f_1^2(P) + f_2^2(P) - 2\alpha f_1(P) f_2(P)}) \geq 0\right\}. \end{aligned}$$

Доповнення множини, визначеної функціонально, збереже вигляд (2.2).

2.2 Моделювання двовимірних форм об'єктів на базі неявних функцій

Двовимірні математичні моделі форм об'єктів найчастіше використовують для подання призматичних тіл зі сталою товщиною (пластин, балок тощо), в яких можна знехтувати тривимірними ефектами. В даному випадку примітивами виступають множини точок на площині виду (2.1), подані дійсними функціями двох дійсних змінних. Базові примітиви, зазвичай, відповідають найпростішим формам, поданим елементарними функціями. Їх суперпозиції, отримані з використанням системи (2.5), визначають нові, більш складні форми.

2.2.1 Примітиви

1. Півплощини

Найпростішою формою є пряма. Вона ділить площину на дві півплощини. Наприклад, множина точок, розташованих нижче прямої $y=y_0$ (включно цю пряму), буде визначена неявною функцією $ru_d(x, y)=y_0-y$. Функція $ru_l(x, y)=-ru_d(x, y)=y-y_0$ визначить півплощину, розташовану вище. Аналогічно, функція $rx_l(x, y)=x_0-x$ буде відповідати півплощині, розташованій ліворуч прямої $x=x_0$, а функція $rx_r(x, y)=-rx_l(x, y)=x-x_0$ визначить півплощину, розташовану праворуч. На практиці пряму зручніше визначати парою точок, через які вона проходить. якщо $P_1(x_1; y_1)$ і $P_2(x_2; y_2)$ – дві різні точки, тоді функція

$$\text{line}(x, y, P_1, P_2) = \frac{(y - P_{1y})(P_{2x} - P_{1x}) - (x - P_{1x})(P_{2y} - P_{1y})}{\sqrt{(P_{2x} - P_{1x})^2 + (P_{2y} - P_{1y})^2}} \quad (2.6)$$

є нормальним орієнтованим рівнянням прямої [116], яке визначає півплощину, розташовану ліворуч під час руху по прямій в напрямку від точки P_1 до P_2 (рис. 2.3).

У формулі (2.6) P_{ix} і P_{iy} позначають, відповідно, абсцису й ординату точки P_i , $i=\{1, 2\}$.

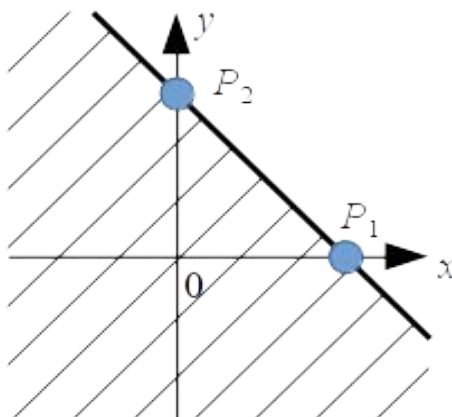


Рисунок 2.3 – Приклад півплощини, визначеної формулою (2.6)

2. Еліпси та кола

До примітивів, що визначаються елементарними функціями, можна віднести області, обмежені еліпсами або, в окремому, але поширеному, випадку – колами.

Область з центром на початку координат, обмежена еліпсом з півосями, рівними a і b , можна подати такою функцією:

$$\text{ellipse}(x, y, a, b) = 1 - \frac{x^2}{a^2} - \frac{y^2}{b^2}, \quad a > 0, \quad b > 0.$$

Якщо $a=b=r$, отримаємо область, обмежену колом радіуса r . Проте, з метою зменшення кількості арифметичних операцій, що використовується, таку область краще подати функцією

$$\text{circle}(x, y, r) = 1 - \frac{x^2 + y^2}{r^2}, \quad r > 0. \quad (2.7)$$

3. Многокутники

Довільний опуклий многокутник функціонально можна подати кон'юнкцією напівплощин, визначених формулою (2.6). Наприклад, якщо його вершини

впорядковані в порядку обходу проти годинникової стрілки, то опуклий багатокутник задається формулою

$$\text{convex}(x, y, P_1, P_2, \dots, P_n) = \text{line}(x, y, P_1, P_2) \wedge \text{line}(x, y, P_2, P_3) \wedge \dots \wedge \text{line}(x, y, P_{n-1}, P_n) \wedge \text{line}(x, y, P_n, P_1), \quad n \geq 3, \quad (2.8)$$

де $P_1, P_2, \dots, P_n$ – координати його вершин;

n – кількість вершин.

В формулі (2.8), а також в наступних, для економії місця в позначеннях кон'юнкції та диз'юнкції опущений символ α . У чисельному аналізі та при візуалізації за замовчуванням прийнято $\alpha=0$ (Якщо будуть використовуватися інші значення, то це буде вказуватися безпосередньо в місці їх використання).

Окремим, але вельми поширеним, випадком формули (2.8) є область, обмежена правильним багатокутником:

$$\begin{aligned} \text{regular}(x, y, r, n) &= \text{convex}(x, y, P_1, P_2, \dots, P_n), \\ P_i &= (r \cos(\alpha_i); r \sin(\alpha_i)), \quad \alpha_i = \frac{2\pi}{n}(i-1), \quad i=1, 2, \dots, n, \quad n \geq 3. \end{aligned} \quad (2.9)$$

Такі фігури як квадрат, паралелограм, прямокутник, ромб також є окремими випадками опуклого багатокутника. Проте, з обчислювальної точки зору їх краще розглядати як самостійні форми. Наприклад, неявну функцію, що відповідає прямокутнику з шириною w і висотою h , можна отримати з спостереження, що він є перетином вертикальної та горизонтальної смуг:

$$\text{rectagle}(x, y, w, h) = \text{stripe}(x, w) \wedge \text{stripe}(y, h), \quad w > 0, \quad h > 0, \quad (2.10)$$

де функція

$$\text{stripe}(x, w) = 1 - \left(\frac{2}{w} x \right)^2 \quad (2.11)$$

відповідає перпендикулярній до заданої осі смузі простору ширини w .

На основі формули (2.10) можна побудувати інший поширений примітив – прямокутник з округленими кутами, який може бути отриманий як об'єднання двох прямокутних областей з чотирма круглими. У такому випадку підсумкова формула набуде вигляду

$$\begin{aligned} \text{rectangle}_r(x, y, w, h, r) = & \text{rectangle}(x, y, w - 2r, h) \vee \text{rectangle}(x, y, w, h - 2r) \vee \\ & \vee \text{circle}\left(x - \frac{w}{2} + r, y - \frac{h}{2} + r, r\right) \vee \text{circle}\left(x + \frac{w}{2} - r, y - \frac{h}{2} + r, r\right) \vee \\ & \vee \text{circle}\left(x + \frac{w}{2} - r, y + \frac{h}{2} - r, r\right) \vee \text{circle}\left(x - \frac{w}{2} + r, y + \frac{h}{2} - r, r\right), \end{aligned} \quad (2.12)$$

$$0 \leq r \leq \min\left(\frac{w}{2}; \frac{h}{2}\right), \quad w > 0, \quad h > 0,$$

де функція rectangle визначена формулою (2.10);

функція circle має вигляд (2.7);

w – ширина прямокутника з округленими кутами;

h – його висота;

r – радіус округлення.

Якщо $r = \min\left(\frac{w}{2}; \frac{h}{2}\right)$, тоді на відповідних сторонах круглі області збігаються.

4. Рух та обертання двовимірних об'єктів

Використання неявних функцій дозволяє відносно просто маніпулювати формами за допомогою відображення, переміщення та обертання.

Зрозуміло, якщо функція $F(x, y)$ визначає геометричну форму деякого об'єкта Ω , тоді функція $F(-x, y)$ визначає симетричну щодо осі Oy форму, $F(x, -y)$ – відносно осі Ox , а $F(-x, -y)$ – відносно початку координат. Якщо необхідно визначити форму об'єкта, всі точки якого зміщені на вектор $\mathbf{v} = [x_0; y_0]^T$ відносно Ω , то можна використовувати функцію $F(x - x_0, y - y_0)$. У випадках, коли необхідно повернути форму об'єкта на кут α , використовують обертання системи координат: $F(x \cos(\alpha) + y \sin(\alpha), -x \sin(\alpha) + y \cos(\alpha))$.

Аналіз більш складних перетворень координат і суперпозицій з періодичними функціями для функціонального подання форм об'єктів з різними типами симетрій виконаний в роботах [69–76].

2.2.2 Приклади функціонального подання креслень двовимірних форм об'єктів

1. L-область.

Розглянемо L-область, зображену на рис. 2.4. Цю область можна подати об'єднанням двох областей, обмежених прямокутниками. В результаті отримаємо таку функцію:

$$\begin{aligned} & \text{lshape}(x, y, w, h, a) = \\ & = \text{rectangle}\left(x - \frac{w}{2}, y - \frac{a}{2}, w, a\right) \vee \text{rectangle}\left(x - \frac{a}{2}, y - \frac{h}{2}, a, h\right), \end{aligned} \quad (2.13)$$

де `rectangle` – функція виду (2.10).

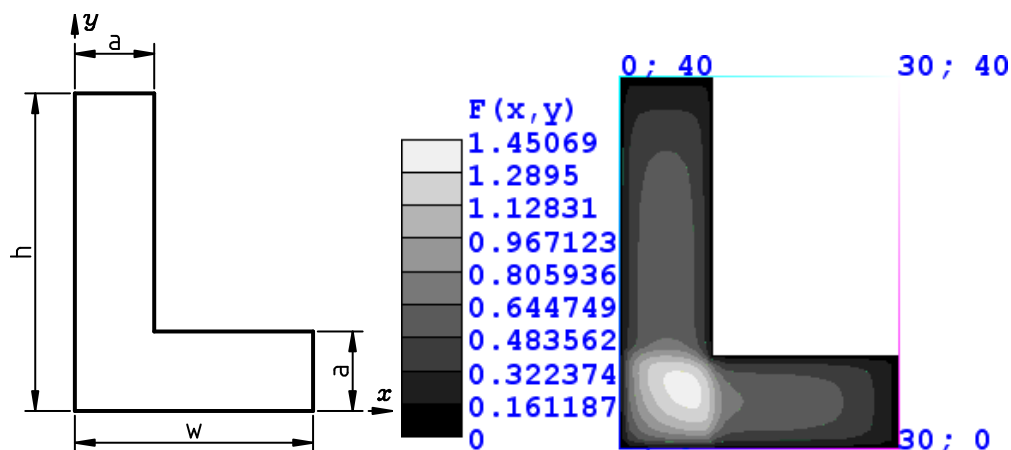


Рисунок 2.4 – L-область: креслення та розподіл значень функції (2.13), якщо $w=30$, $h=40$ і $a=10$

2. Квадратна пластина з трикутним отвором у центрі.

Квадратну пластину, що має трикутний отвір у центрі, (рис. 2.5) можна подати з використанням функції (2.9) у такому вигляді:

$$\text{qplate}(x, y, a, r) = \text{rectangle}(x, y, a, a) \wedge \neg \text{regular}(x, y, r, 3). \quad (2.14)$$

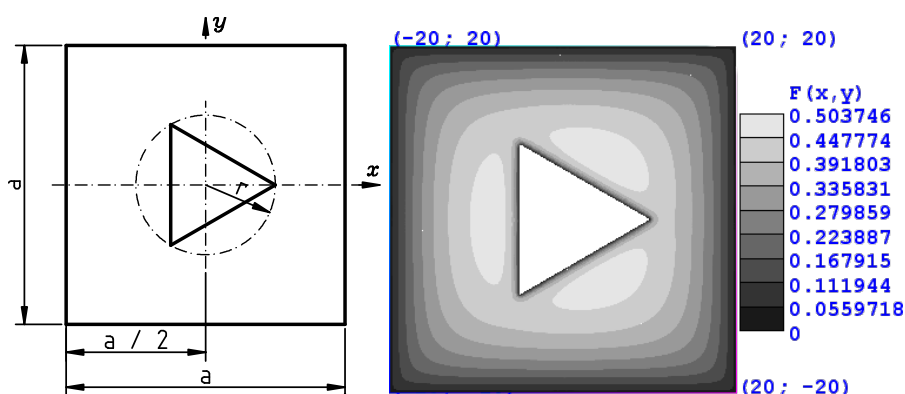


Рисунок 2.5 – Квадратна пластина з трикутним отвором: креслення і розподіл значень функції (2.14), якщо $a=40$ і $r=10$

3. Профіль гайки.

Двовимірну область у формі гайки (рис. 2.6), також можна задати за допомогою формул (2.7) і (2.9) (як логічну різницю області, обмеженої правильним шестикутником, і області, обмеженої колом):

$$\text{nut}(x, y, r_1, r_2) = \text{regular}(x, y, r_1, 6) \wedge \neg \text{circle}(x, y, r_2), \quad (2.15)$$

де радіуси r_1 і r_2 , що зображені на рис. 2.6.

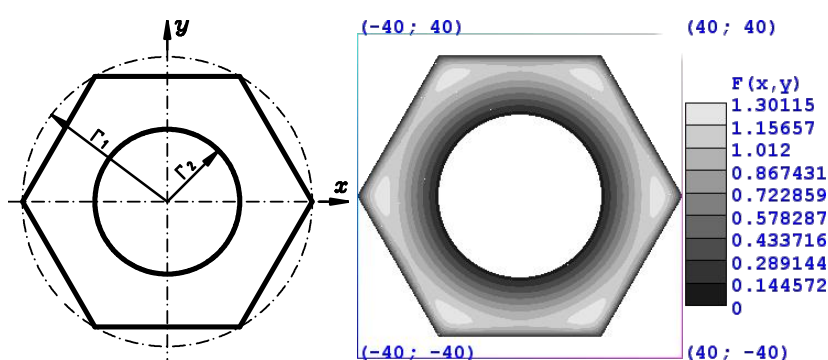


Рисунок 2.6 – Модель двовимірної гайки: креслення і розподіл значень функції (2.15), якщо $r_1=40$ і $r_2=20$

4. Фрикційний диск трансмісії.

Фрикційний диск трансмісії можна подати як геометричний об'єкт з точковою симетрією циклічного типу у вигляді такої функції:

$$\begin{aligned} & \text{clutch}(x, y, r, R, S, w, n) = \\ & = \left[\text{stripe} \left(\sin \left(\mu(n, \arctg_2(y, x)) \sqrt{x^2 + y^2} \right), w \right) \wedge \text{circle}(x, y, S) \vee \text{circle}(x, y, R) \right] \wedge \\ & \quad \wedge \neg \text{circle}(x, y, r), \end{aligned} \quad (2.16)$$

де r, R, S і w – розміри, зображені на рис. 2.7;

n – кількість зубців;

stripe – функція (2.11);

circle – функція (2.7);

$\mu(n, \theta)$ – часткова сума ряду Фур'є, визначена формулою [73]

$$\mu(n, \theta) = \frac{8}{\pi n} \sum_{k=1}^4 (-1)^{k+1} \sin \left[\frac{(2k-1) \frac{n\theta}{2}}{(2k-1)^2} \right]; \quad (2.17)$$

$\arctg_2(y, x)$ – спеціальна функція – арктангенс відношення y до x [381].

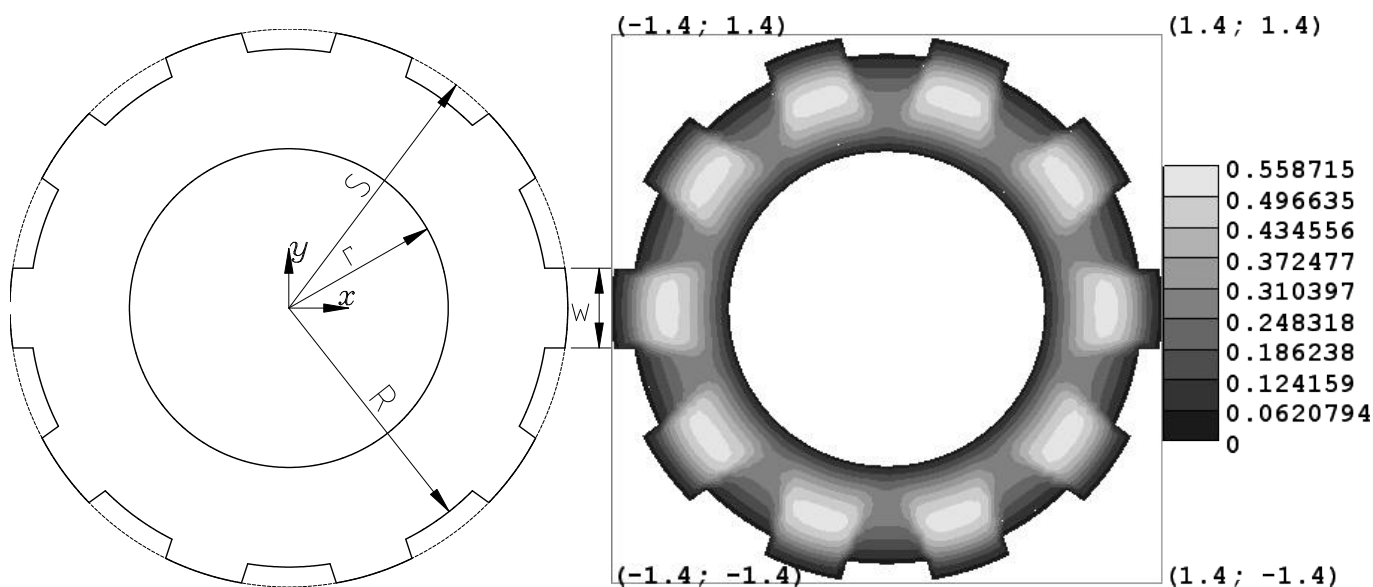


Рисунок 2.7 – Фрикційний диск трансмісії: креслення і розподіл значень функції (2.16), якщо $S=1,4$, $R=1,3$, $r=0,8$, $w=0,4$ і $n=10$

5. Планка

Контур спеціальної планки (рис. 2.8), можна подати такою формулою:

$$\begin{aligned} \text{planch}(x, y, w, h, R, r) = & \left[\text{stripe}(x, w) \wedge \text{stripe}\left(y, h + (2R - h)\frac{2x + w}{2w}\right) \right] \vee \text{circle}\left(x - \frac{w}{2}, y, R\right) \Bigg\} \wedge \\ & \wedge \neg \text{circle}\left(x - \frac{w}{2}, y, r\right), \end{aligned} \quad (2.18)$$

де w і h – відповідно, ширина і висота планки;

R – радіус заокруглення;

r – радіус отвору.

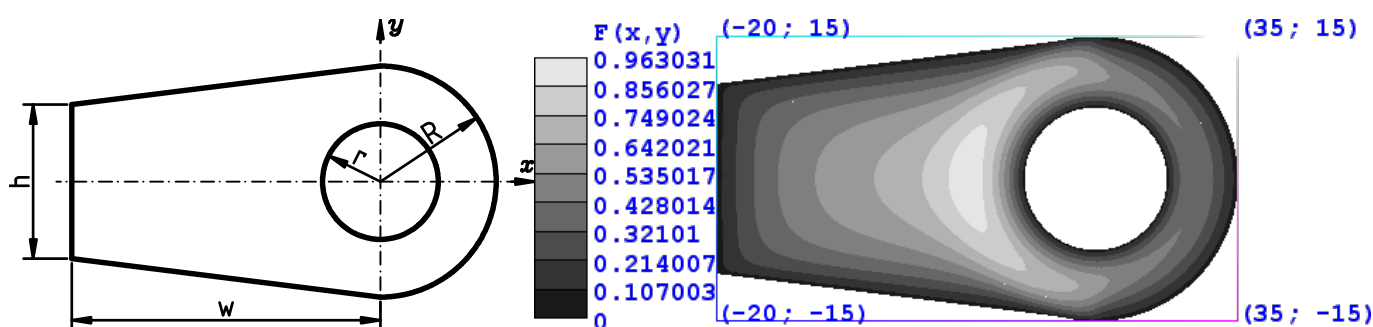


Рисунок 2.8 – Контур спеціальної планки: креслення і розподіл значень функції (2.18), якщо $h=20$, $w=40$, $R=15$ і $r=7,5$

6. Прокладка

Модель прокладки (рис. 2.9) можна подати у такому вигляді:

$$\begin{aligned} \text{layer}(x, y) = & \left[\text{circle}(x, y, r_1) \vee \text{circle}(x, y + h_2, r_5) \vee \text{rectangle}\left(x, y + \frac{h_2}{2}, w_2, h_2\right) \right] \wedge \\ & \wedge \neg \text{circle}(x, y, r_2) \wedge \neg \text{rectangle}(x, y - r_2, w_1, 2h_1) \wedge \\ & \wedge \neg \text{circle}\left(r_{xy} \cos\left(\mu\left(4, \theta + \frac{\pi}{4}\right)\right) - r_3, r_{xy} \sin\left(\mu\left(4, \theta + \frac{\pi}{4}\right)\right), r_4\right) \wedge \\ & \neg \text{rectangle}\left(x, y + h_2 - \frac{h_3}{2}, 2r_6, h_3\right) \wedge \neg \text{circle}(x, y + h_2, r_6) \wedge \neg \text{circle}(x, y + h_2 - h_3, r_6), \end{aligned}$$

де параметри $r_1 = \frac{80}{2} = 40$, $r_2 = \frac{40}{2} = 20$, $r_3 = \frac{60}{2} = 30$, $r_4 = \frac{10}{2} = 5$, $r_5 = 24$, $r_6 = 14$,

$w_1 = 14$, $h_1 = 5$, $w_2 = 2r_5$, $h_2 = 100$, $h_3 = 48$ відповідають розмірам на кресленні;

$$r_{xy} = \sqrt{x^2 + y^2};$$

$\arctg_2(y, x)$ – спеціальна функція – арктангенс відношення y до x [381];

$\mu(n, \theta)$ – функція (2.17), яка забезпечує трансляцію отвору.

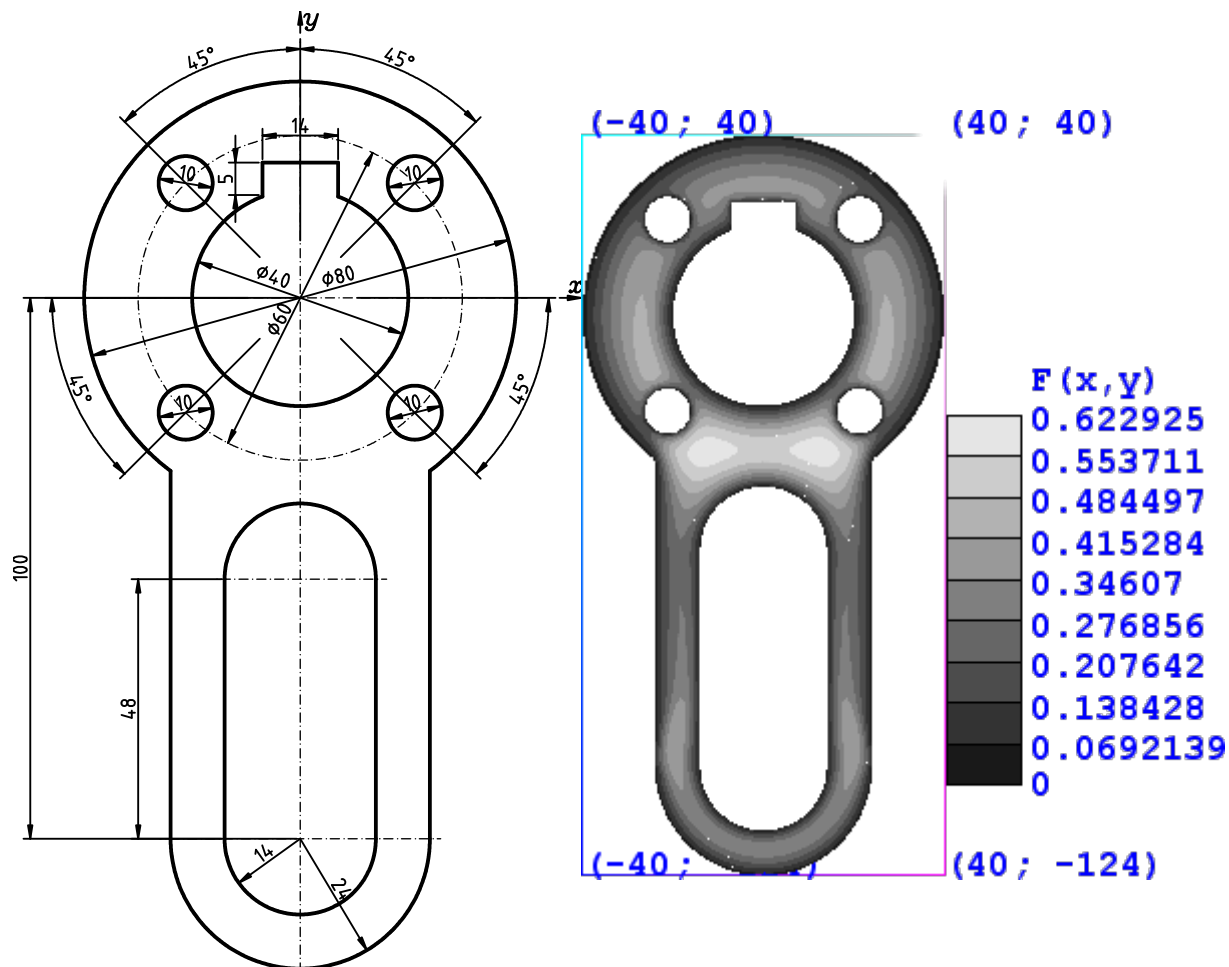


Рисунок 2.9 – Прокладка: креслення і розподіл значень функції $\text{layer}(x, y)$

2.3 Моделювання тривимірних форм об'єктів на базі неявних функцій

Чимало моделей тривимірних примітивів можна отримати за аналогією з відповідними двовимірними. Крім того, наведені вище неявні функції у тривимірному просторі визначають нескінченні (уздовж осі Oz) призматичні області, що можна використовувати як самостійні тривимірні примітиви (наприклад, для опису форм отворів).

2.3.1 Примітиви

1. Півпростори

За аналогією з двовимірним випадком найпростішою формою буде площина. Вона ділить простір на два півпростори. Так, наприклад, неявна функція $px(x, y, z) = x_0 - x$ буде відповідати півпростору, розташованому лівіше площини $x = x_0$, функція $py(x, y, z) = y_0 - y$ визначає множину точок, розташованих нижче площини $y = y_0$, а функція $pz(x, y, z) = z_0 - z$ визначить півпростір, розташований глибше площини $z = z_0$.

На практиці площину досить часто доводиться визначати трьома точками, через які вона проходить. Якщо $P_1(x_1; y_1; z_1)$, $P_2(x_2; y_2; z_2)$ і $P_3(x_3; y_3; z_3)$ – різні точки, що не лежать на одній прямій, то функція

$$\text{plane}(x, y, z, P_1, P_2, P_3) = \frac{(x - x_1)A + (y - y_1)B + (z - z_1)C}{\sqrt{A^2 + B^2 + C^2}}, \quad (2.19)$$

де x_i , y_i і z_i позначають, відповідно, абсцису, ординату і аплікату точки P_i , $i = \{1, 2, 3\}$;

$$A = (y_2 - y_1)(z_3 - z_1) - (z_2 - z_1)(y_3 - y_1);$$

$$B = (z_2 - z_1)(x_3 - x_1) - (x_2 - x_1)(z_3 - z_1);$$

$$C = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1).$$

За аналогією з формулою (2.6), функцію (2.19) можна вважати нормальним орієнтованим рівнянням площини.

2. Еліпсоїди, сфери та циліндри

До тривимірних примітивів, що визначаються елементарними функціями, можна віднести області, обмежені еліпсоїдами або, в окремому випадку, сферами.

Тривимірне тіло з центром на початку координат, обмежене еліпсоїдом з півосями, рівними a , b і c , можна подати такою функцією:

$$\text{ellipsoid}(x, y, z, a, b, c) = 1 - \frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2}, \quad a > 0, \quad b > 0, \quad c > 0.$$

Аналогічно двовимірному випадку, якщо $a=b=c=r$ отримаємо модель тіла, обмеженого сферою радіусу r , але з практичних міркувань таку область краще подати функцією

$$\text{sphere}(x, y, z, r) = 1 - \frac{x^2 + y^2 + z^2}{r^2}. \quad (2.20)$$

Циліндри можна подати функцією (2.7) з обмеженням по довжині твірної. Наприклад, якщо припустити, що твірна паралельна осі Oz , то такий циліндр задається формулою

$$\text{cylinder}(x, y, z, r, h) = \text{circle}(x, y, r) \wedge \text{stripe}(z, h), \quad (2.21)$$

де r – радіус основи;

h – довжина твірної;

stripe – функція (2.11).

Циліндри з твірними, які спрямовані вздовж інших осей, можна отримати перестановками координат у формулі (2.21).

3. Багатогранники

Довільний опуклий багатогранник можна подати перетином (кон'юнкцією) граней, які обмежують його, у такому вигляді:

$$\begin{aligned} \text{polyhedron}(x, y, z, (P_{11}, P_{12}, P_{13}), \dots, (P_{k1}, P_{k2}, P_{k3})) = \\ = \text{plane}(x, y, z, P_{11}, P_{12}, P_{13}) \wedge \dots \wedge \text{plane}(x, y, z, P_{k1}, P_{k2}, P_{k3}), \end{aligned} \quad (2.22)$$

де k – кількість граней;

P_{ij} ($i=\overline{1, k}, j=\{1, 2, 3\}$) – точки, які належать i -й грані у порядку їх обходу.

Наприклад, октаедр з ребром $\sqrt{2}$ (рис. 2.10) з використанням формули (2.22) подається у такому вигляді:

$$\begin{aligned} \text{octahedron}_2(x, y, z) = \\ = \text{polyhedron}(x, y, z, (P_3, P_2, P_1), (P_1, P_2, P_4), (P_5, P_3, P_1), (P_1, P_4, P_5), \\ (P_6, P_2, P_3), (P_4, P_2, P_6), (P_6, P_3, P_5), (P_5, P_4, P_6)), \end{aligned} \quad (2.23)$$

де $P_1(0; 0; 1)$, $P_2(0; -1; 0)$, $P_3(1; 0; 0)$, $P_4(-1; 0; 0)$, $P_5(0; 1; 0)$, $P_6(0; 0; -1)$ – вершини октаедра.

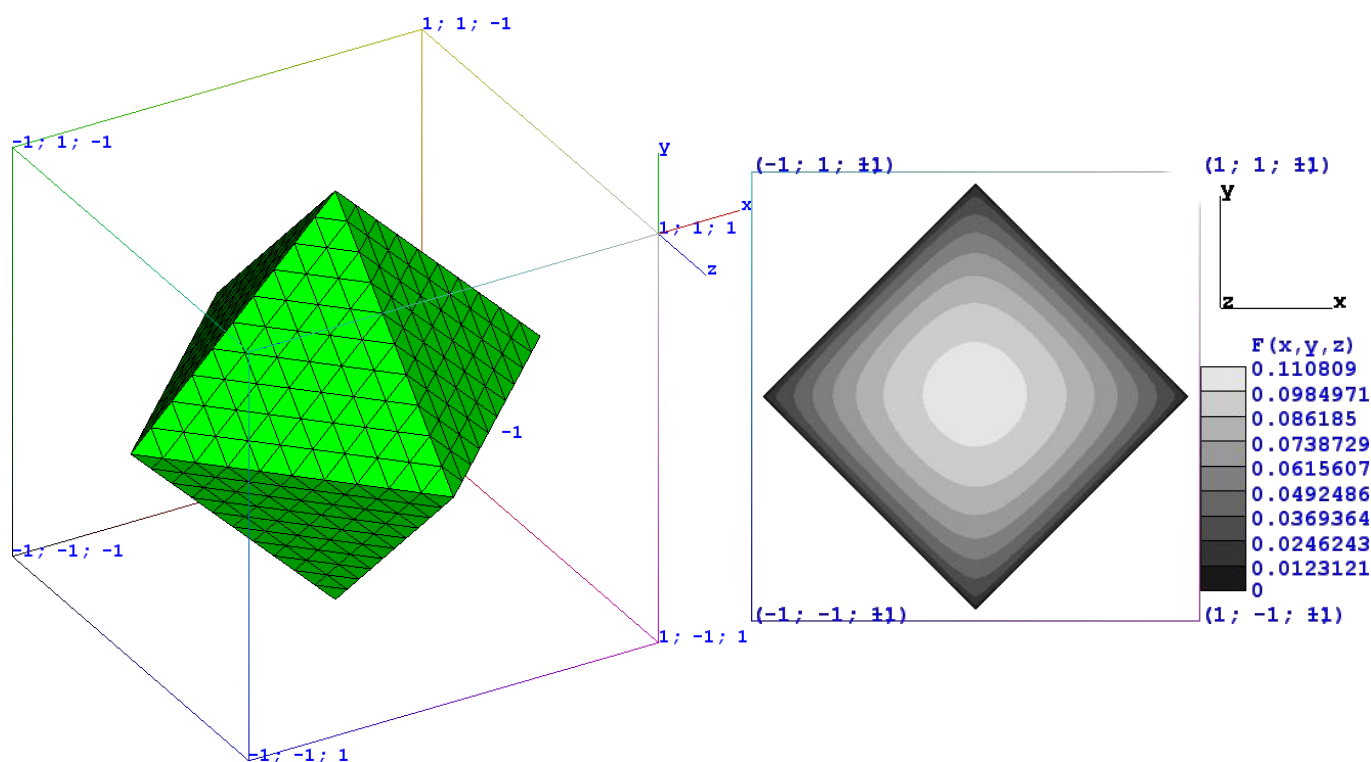


Рисунок 2.10 – Октаедр: загальний вигляд і розподіл значень функції (2.23) у перерізі площиною $z=0$

Тривимірним аналогом формули (2.10) для прямокутника є формула (2.24), яка також заснована на функції (2.11). Формулу (2.24) можна використовувати для визначення прямокутного паралелепіпеда зі сторонами, рівними w , h і d :

$$\begin{aligned} \text{cuboid}(x, y, z, w, h, d) = \text{stripe}(x, w) \wedge \text{stripe}(y, h) \wedge \text{stripe}(z, d), \\ w > 0, h > 0, d > 0. \end{aligned} \quad (2.24)$$

2.3.2 Приклади функціонального подання креслень тривимірних форм об'єктів

1. Гайка з метричною різьбою.

Гайка є циліндричним тілом (рис. 2.11). Її основа – правильний шестикутник, вписаний в коло радіуса $R=S/\sqrt{3}$, (S – розмір «під ключ»). Висота циліндра дорівнює H . У ньому є осьовий отвір з радіусом $r_i=0,85 r$ ($r=d/2$ – зовнішній радіус різьби відповідного болта, r_i – внутрішній радіус різьби), а також дві фаски: під кутом 30° на відстані $d_w/2$ від центру та під кутом 120° на відстані r_i від центру.

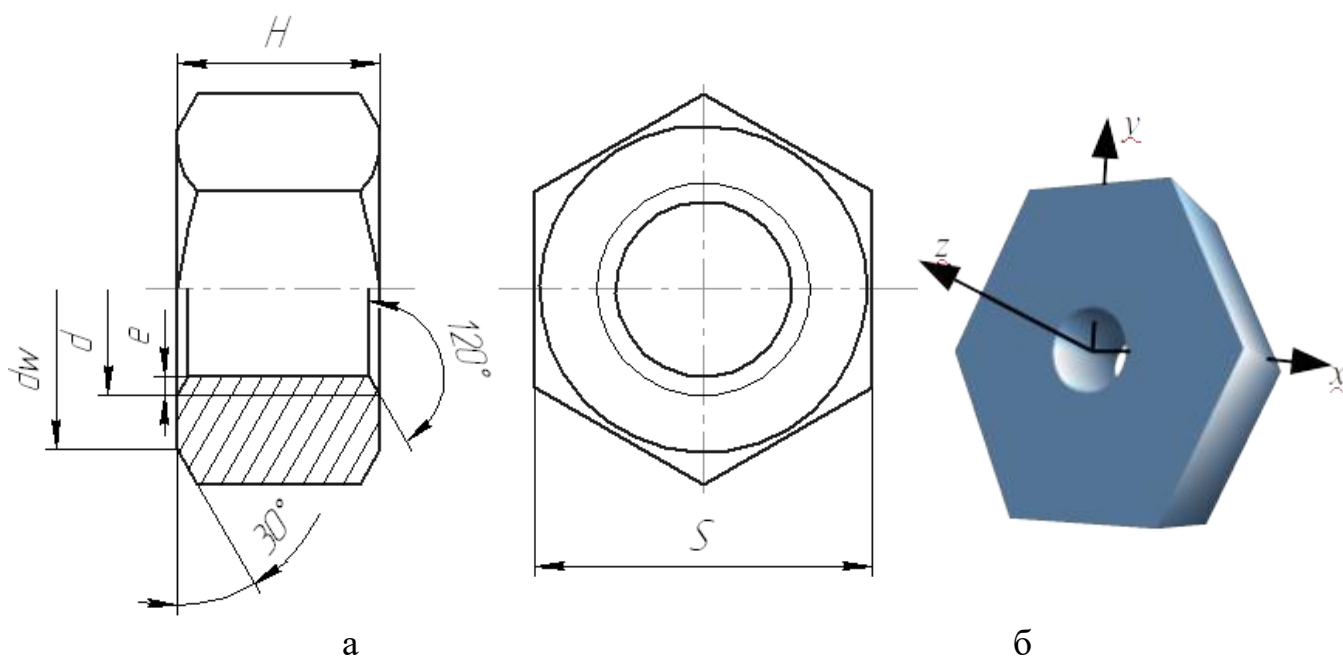


Рисунок 2.11 – Гайка:

а – креслення

б – схема розташування осей координат

Якщо систему координат вибрати таким чином, щоб вісь Ox проходила через будь-яких два протилежних кута гайки, ортогональна їй вісь Oy – через середини двох протилежних граней, а вісь Oz збігалася з віссю гайки (рис. 2.11, б), то гайку можна подати такою формулою:

$$\begin{aligned} \text{nut}_{3D}(x, y, z) = \\ = \text{nut}(x, y, R, r_i) \wedge \left(\frac{H^2}{4} - z^2 \right) \wedge \neg \text{rbevel}(\sqrt{x^2 + y^2}, z) \wedge \neg \text{thread}(x, y, z), \end{aligned} \quad (2.25)$$

де функція $\text{rbevel}(x, y)$ визначає фаски в тілі обертання, її вигляд такий:

$$\begin{aligned} \text{rbevel}(x, y) = & \text{line} \left(x, y, \left(\frac{d_w}{2}; -\frac{H}{2} \right), \left(R; -\frac{H}{2} + b \right) \right) \wedge \\ & \wedge \text{line} \left(x, y, \left(R; \frac{H}{2} - b \right), \left(\frac{d_w}{2}; \frac{H}{2} \right) \right) \wedge \\ & \wedge \text{line} \left(x, y, \left(r - e; -\frac{H}{2} + c \right), \left(r; -\frac{H}{2} \right) \right) \wedge \text{line} \left(x, y, \left(r; \frac{H}{2} \right), \left(r - e; \frac{H}{2} - c \right) \right). \end{aligned} \quad (2.26)$$

Функція $\text{thread}(x, y, z)$ відповідає метричній різьбі, яку можна подати як гвинтовий геометричний об'єкт, утворений рухом правильного трикутного профілю зі стороною, що дорівнює кроку різьби:

$$\begin{aligned} \text{thread}(x, y, z) = & \text{regular} \left(\sqrt{x^2 + y^2} - r_i + \delta, \mu(x, y, z), \frac{m}{\sqrt{3}}, 3 \right) \wedge \\ & \wedge \neg \text{circle}(x, y, r), \end{aligned} \quad (2.27)$$

де m – крок різьби;

$$\delta = m \frac{\sqrt{3}}{12} - \frac{r - r_i}{2} \quad \text{– зсув трикутника, що забезпечує центральне положення поверхні}$$

змійовика щодо внутрішньої та зовнішньої поверхонь різьби (рис. 2.12);

$$\mu(x, y, z) = m \frac{4}{\pi^2} \sum_{k=1}^{\infty} (-1)^{k+1} \frac{\sin \left[(2k-1) \frac{\pi}{m} \left(z - \frac{m}{2\pi} \text{atan}_2(y, x) \right) \right]}{(2k-1)^2} \quad \text{– перетворення, що}$$

забезпечує повторення кілець змійовика [73].

$$\text{В формулах (2.25)–(2.26) введено такі позначення: } b = \left(R - \frac{d_w}{2} \right) \text{tg} \frac{\pi}{6}; \quad c = e \text{tg} \frac{\pi}{3};$$

$e = r - r_i = 0,15r$ – розмір зубця різьби; $\text{nut}(x, y, r_1, r_2)$ – функція, що має вигляд (2.15); $\text{line}(x, y, P_1, P_2)$ – функція (2.6).

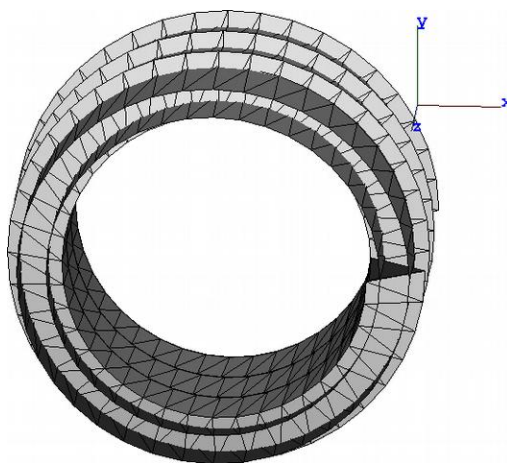


Рисунок 2.12 – Взаємне розташування поверхонь різьби

При обчисленні функції (2.27) досить підсумовування чотирьох членів ряду. На рис. 2.22 наведено загальний вигляд поверхні гайки, визначеної формулою (2.25), і розподіл значень функції в перерізі площиною $y=0$.

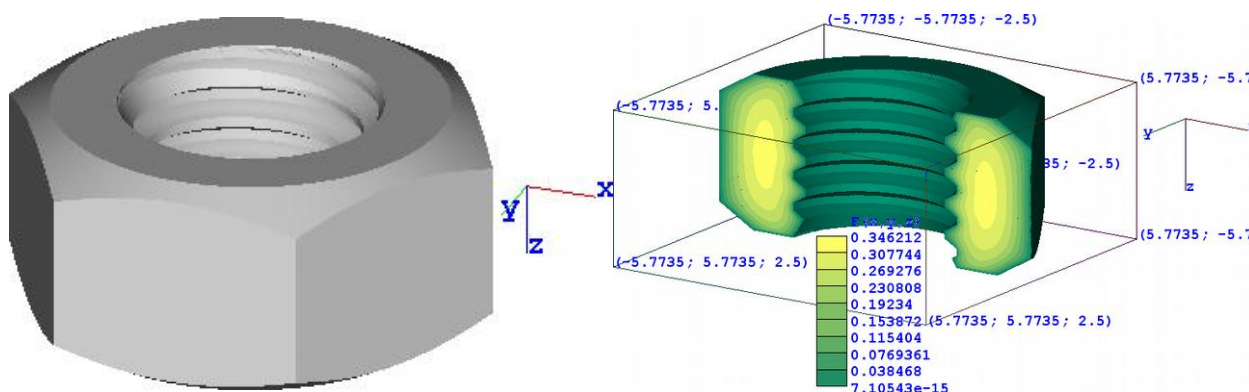


Рисунок 2.13 – Гайка: загальний вигляд і розподіл значень функції (2.25) в перерізі площиною $y=0$, якщо $S=10$, $H=5$, $d=6$ і $m=1$

2. Перехідник задньої ступиці.

Перехідник задньої ступиці (рис. 2.14) можна розглядати як циліндричне тіло з направляючою в формі прямокутника з округленими кутами. Всі кругові вирізи, крім найбільшого, можна подати як логічне віднімання кругових областей відповідного радіуса.

Якщо припустити, що площина xOy збігається з серединною площиною профілю, причому початок координат розташовано в центрі найбільшого отвору, то перехідник задньої ступиці можна подати такою формулою:

$$\begin{aligned}
 w(x, y, z) = & \text{rectangle}_r(x, y, 110, 84, 10) \wedge \\
 & \wedge \neg \text{circle}(x-43, y-20, 5) \wedge \neg \text{circle}(x+43, y-20, 5) \wedge \\
 & \wedge \neg \text{circle}(x-38.5, y+28, 5) \wedge \neg \text{circle}(x+38.5, y+28, 5) \wedge \\
 & \wedge \neg \text{circle}(x-30.5, y-17, 5) \wedge \neg \text{circle}(x+30.5, y-17, 5) \wedge \\
 & \wedge \neg \text{circle}(x, y+36, 6) \wedge \neg [\text{circle}(x, y, 28) \wedge z] \wedge \text{stripe}(z, 20),
 \end{aligned}$$

де rectangle_r – функція (2.12);

circle – функція (2.7);

stripe – функція (2.11).

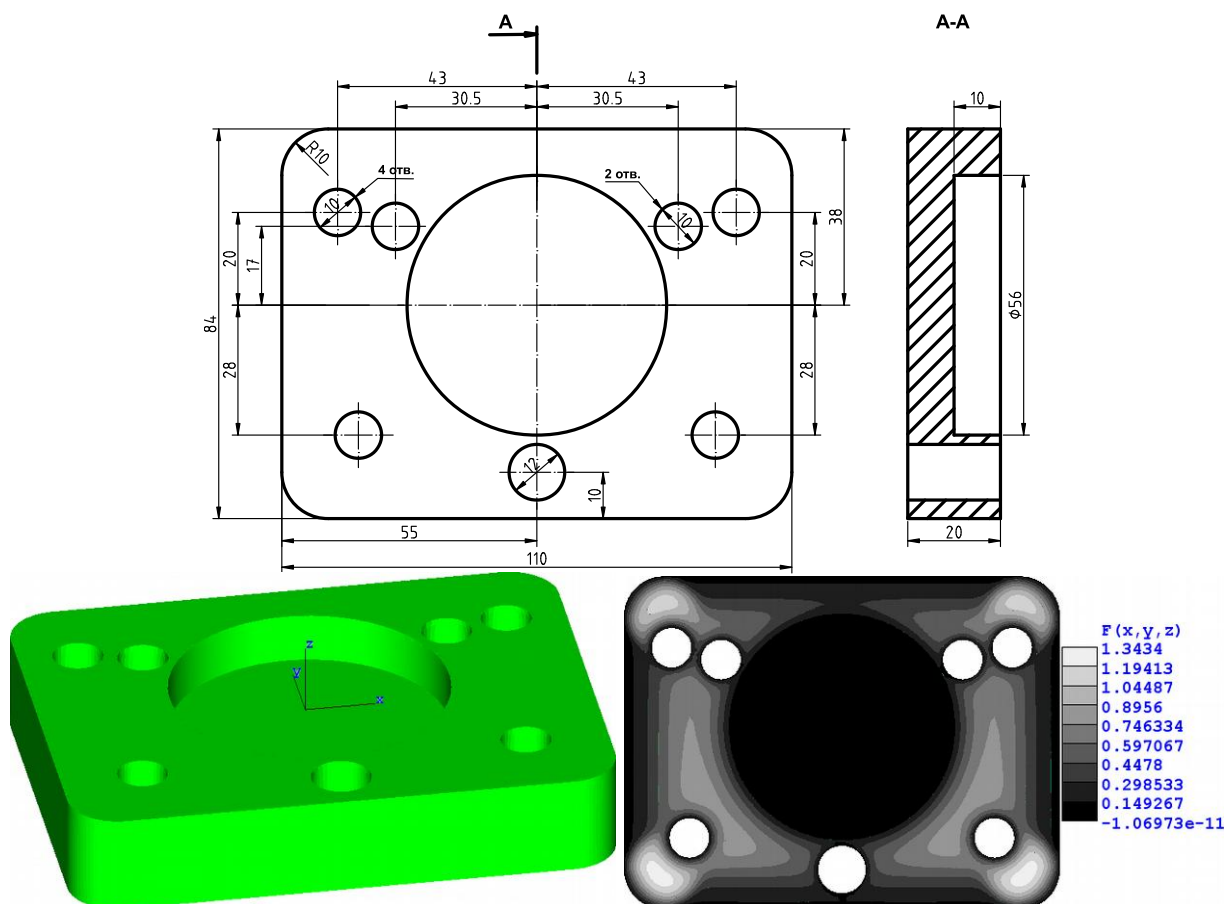


Рисунок 2.14 – Перехідник задньої ступиці: креслення, загальний вигляд і розподіл значень функції в перерізі площиною $z=0$

3. Подовжувач важеля передньої підвіски.

Направляючу подовжувача важеля передньої підвіски можна подати логічним об'єднанням прямокутної області з двома круговими, а також відніманням двох напівплощин, що забезпечують звуження у верхній частині (рис. 2.15), і двох кругових областей (циліндричні отвори):

$$p(x, y) = [\text{circle}(x, y, 21) \vee \text{rectangle}(x, y - 115/2, 42, 115) \vee \text{circle}(x, y - 115, 14)] \wedge \\ \wedge \text{line}(x, y, (-14; 115), (-21; 81)) \wedge \text{line}(x, y, (21; 81), (14; 115)) \wedge \\ \wedge \neg \text{circle}(x, y - 115, 5) \wedge \neg \text{circle}(x, y - 72, 5).$$

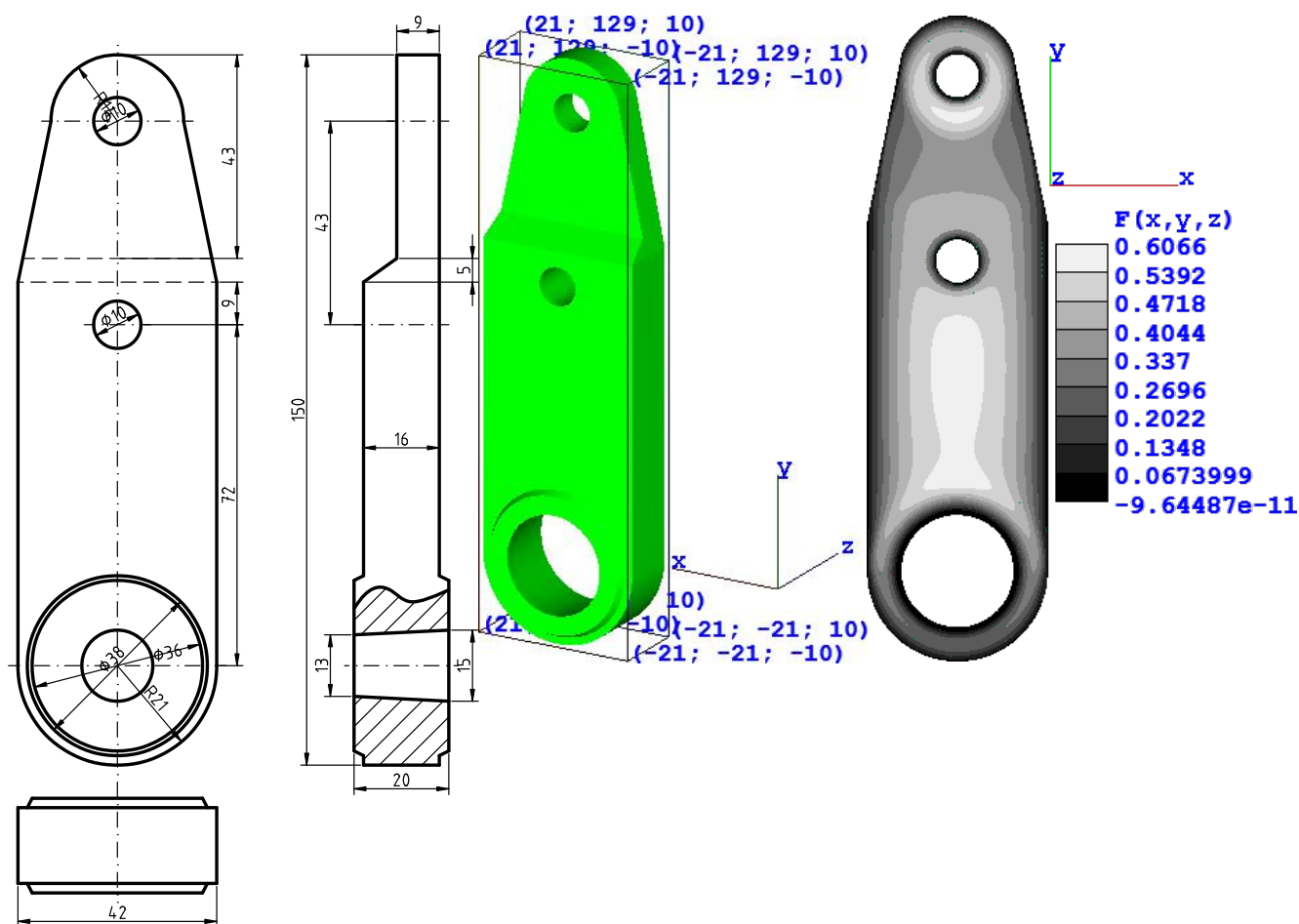


Рисунок 2.15 – Подовжувач важеля передньої підвіски: загальний вигляд і розподіл значень функції в перерізі площиною $z=0$

Підсумкова модель подовжувача важеля передньої підвіски буде отримана в результаті додавання відповідних обмежень щодо глибини, а також конічних виступів й отвору:

$$\begin{aligned} w(x, y, z) = & \left\{ \left[p(x, y) \wedge \text{stripe}(z, 16) \wedge \neg(\text{line}(z, y, [-8; 81], [-1; 86]) \wedge z - 1) \right] \vee \right. \\ & \left. \vee \left[\text{circle}\left(x, y, 15 + \frac{z}{2}\right) \wedge 10 - z \wedge z - 7 \right] \vee \left[\text{circle}\left(x, y, 15 - \frac{z}{2}\right) \wedge z + 10 \wedge -z - 7 \right] \right\} \wedge \\ & \wedge \neg \text{circle}\left(x, y, 14 + \frac{z}{10}\right). \end{aligned} \quad (2.28)$$

4. Пробка.

Пробка (рис. 2.16) являє собою циліндр змінного радіуса, довжина твірної якого дорівнює 35 мм. Уздовж твірної пробки розташований циліндричний виріз з перетином шестикутної форми.

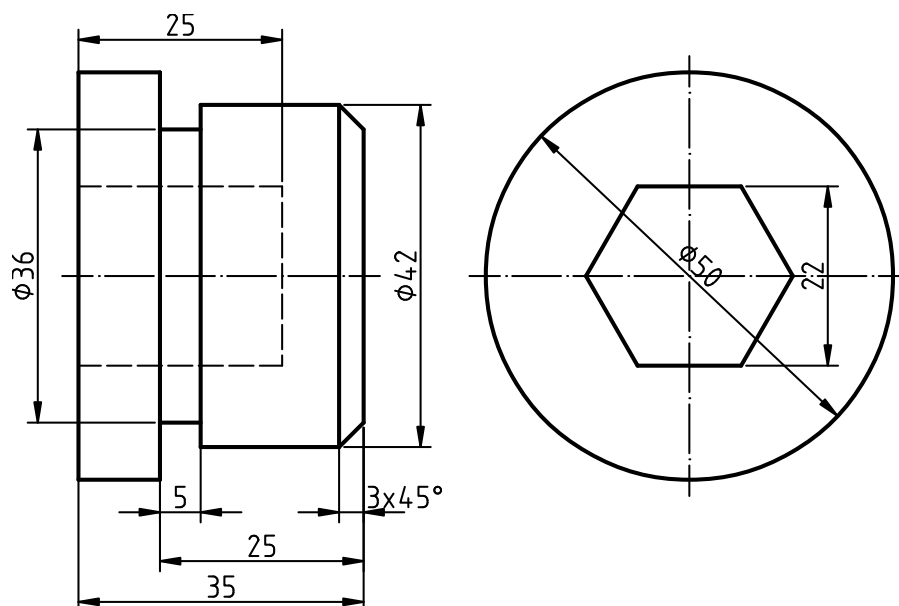


Рисунок 2.16 – Креслення пробки

Модель пробки (рис. 2.17) можна побудувати, «вирізаючи» отвори необхідної форми з циліндра, довжина якого дорівнює 35 мм, а радіус – 50 мм:

$$\begin{aligned}
 w(x, y, z) = & \text{cylinder}(x, y, z, 25, 35) \wedge \text{groove}(x, y, z+5, 18, 5) \wedge \\
 & \wedge \text{groove}(x; y; z-18,5; 21; 20) \wedge \text{circle}(x; y; 35,5-z) \wedge \\
 & \wedge \neg \left[\text{regular} \left(x, y, \frac{22}{\sqrt{3}}, 6 \right) \wedge (7,5-z) \right],
 \end{aligned} \tag{2.29}$$

де cylinder – функція (2.21);

circle – функція (2.7);

regular – функція (2.9);

groove – функція, яка подає кільцеву канавку радіуса r і ширини w :

$$\text{groove}(x, y, z, r, w) = \text{circle}(x, y, r) \vee \neg \text{stripe}(z, w). \tag{2.30}$$

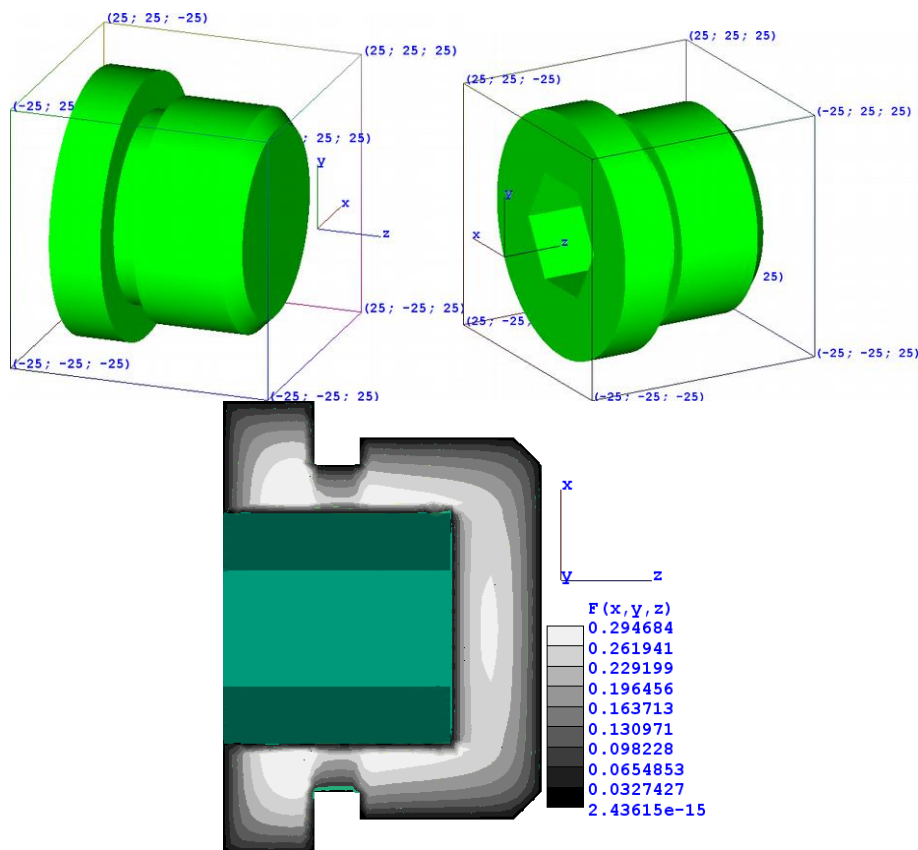


Рисунок 2.17 – Модель пробки: загальний вигляд і розподіл значень функції в перерізі площиною $y=0$

5. Поршень.

Поршень, аналогічно розглянутій вище моделі пробки, являє собою циліндричне тіло (рис. 2.18). У ньому є три кільцеві канавки, які можна подати за допомогою функції `groove`, що визначена формулою (2.30). Поршень володіє двома осьовими вирізами: 1) діаметра 69,3 мм ($75,9 - 6,6 = 69,3$) і глибиною 53 мм; 2) діаметра 66,9 мм і глибиною 74 мм ($79,7 - 5,7 = 74$). Стінки поршня потовщуються в площині кріплення шатуна. В результаті отримано таку модель (рис. 2.18):

$$\begin{aligned}
 \text{piston}(x, y, z) = & \text{cylinder}(x, y, z, R, L) \wedge \neg \text{rectangle}_r\left(x, z + \frac{L}{2}, R - a, 2h, r\right) \wedge \\
 & \wedge \neg \left[\left\{ \text{hzcylinder}\left(x, y, z + \frac{L}{2} - d_1, R_1\right) \vee \text{hzcylinder}\left(x, y, z + \frac{L}{2} - d_2, R_2\right) \right\} \wedge \text{stripe}(y, w) \right] \wedge \\
 & \wedge \neg \text{circle}\left(x, z + \frac{L}{2} - d_1 + R_c, R_c\right) \wedge \text{groove}(x, y, z + \delta_1, R_1, rw_1) \wedge \\
 & \wedge \text{groove}(x, y, z + \delta_2, R_1, rw_2) \wedge \text{groove}(x, y, z + \delta_3, R_1, rw_3),
 \end{aligned}$$

де $R = \frac{75,9}{2}$ мм – радіус поршня;

$L = 79,7$ мм – довжина поршня;

$a = 16$ мм – відстань до прямокутного вирізу на «спідниці» поршня;

$h = 8$ мм – глибина прямокутного вирізу на «спідниці» поршня;

$r = 4$ мм – радіус округлення кутів прямокутника на «спідниці» поршня;

$R_1 = \frac{69,3}{2}$ мм – радіус першого вісєвого вирізу;

$d_1 = 53$ мм – глибина першого вісєвого вирізу;

$R_2 = \frac{66,9}{2}$ мм – радіус другого вісєвого вирізу;

$d_2 = 74$ мм – глибина другого вісєвого вирізу;

$w = 50$ мм – ширина потовщення «спідниці» поршня;

$R_c = \frac{22,2}{2}$ мм – радіус поперечного кругового вирізу;

$\delta_1 = \frac{L}{2} - 57,6 - \frac{rw_1}{2}$ – відстань до центра першої кільцевої канавки;

$rw_1 = 4,2$ мм – ширина першої кільцевої канавки;

$\delta_2 = \frac{L}{2} - 57,6 - rw_1 - 3 - \frac{rw_2}{2}$ – відстань до центра другої кільцевої канавки;

$rw_2 = 2,2$ мм – ширина другої кільцевої канавки;

$\delta_3 = \frac{L}{2} - 57,6 - rw_1 - 3 - rw_2 - 3 - \frac{rw_3}{2}$ – відстань до центра третьої кільцевої

канавки;

$rw_3 = 1,8$ мм – ширина третьої кільцевої канавки;

cylinder – функція (2.21);

rectangle_r – функція (2.12);

hzcylinder(x, y, z, r) = circle(x, y, r) $\wedge \neg z$ – функція, яка подає нескінчений циліндр з однією основою ($z=0$) і віссю $-Oz$;

circle – функція (2.7);

groove – функція (2.30).

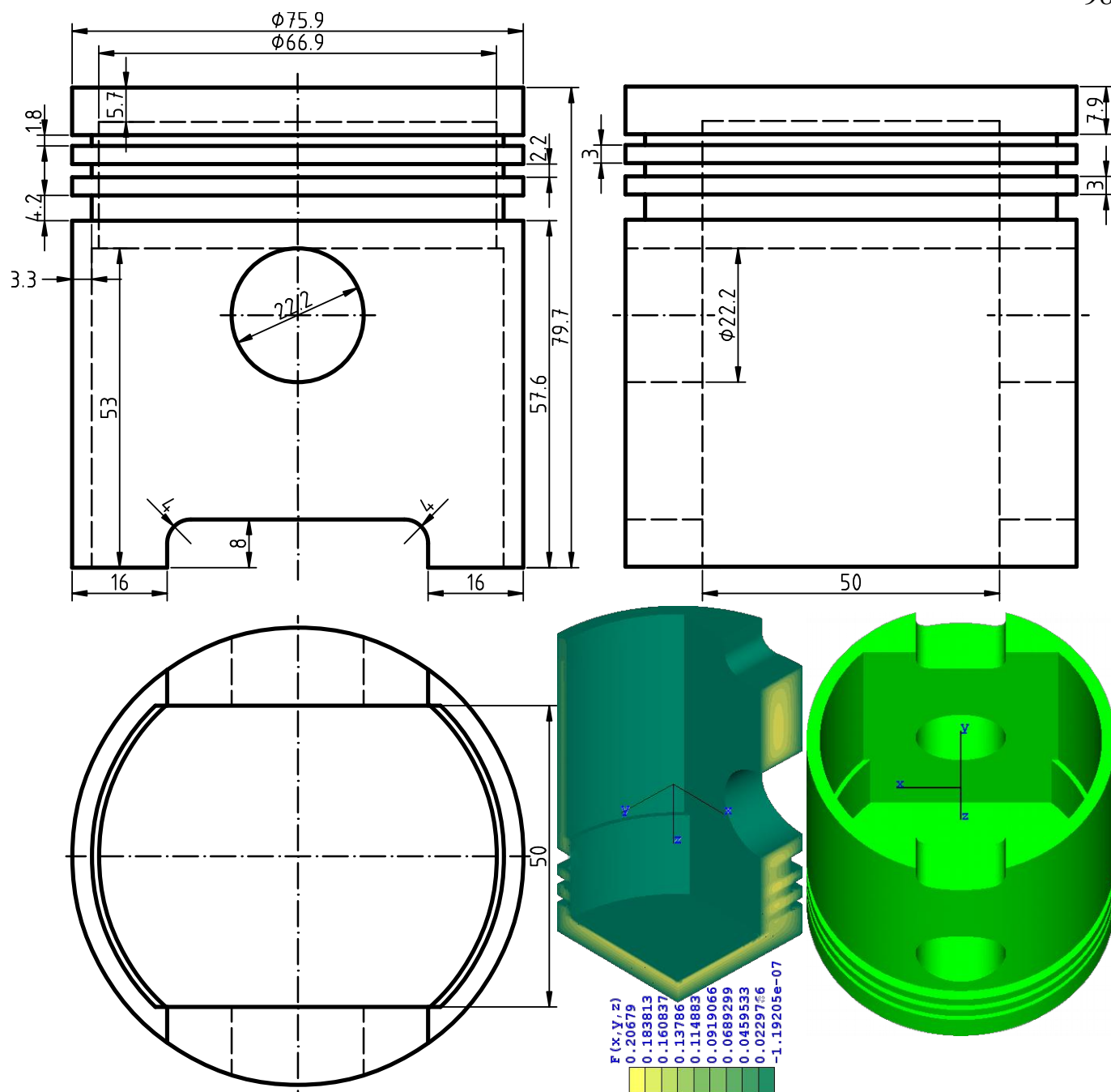


Рисунок 2.18 – Креслення та загальний вигляд поршня

2.4 Гібридний підхід до подання форм об'єктів машинобудування

Функціональне подання з використанням неявних функцій є досить універсальним і дозволяє описувати довільні форми об'єктів. Проте, в авіа-, авто- і суднобудуванні дуже поширене використання параметричних кривих або поверхонь,

які обмежують об'єкт (граничне подання). За такої умови, як зазначалося раніше, побудова системи меж для нестандартного об'єкта (наприклад, зі розгалуженою структурою технологічних отворів) також досить складна. Можливий спосіб спрощення процесу моделювання форм об'єктів – комбінування елементів функціонального та граничного подань (гібридне подання) шляхом перетворення граничних моделей в функціональні. Перехід від граничного подання примітива (системи параметричних кривих або поверхонь) до еквівалентного функціонального (неявного рівняння) за допомогою розв'язання відповідної системи рівнянь можливий лише в окремих відносно простих випадках і, отже, актуальною є розробка методів розв'язання цієї задачі.

В основі граничного подання покладено теорему Жордана-Брауера (узагальнення теореми Жордана для n -мірного простору), яка гарантує, що будь-який $(n-1)$ -вимірний підмноговид в $\mathbf{R}^n$, гомеоморфний сфері, розбиває простір на дві зв'язні компоненти та є їх спільною межею. Тобто, в двовимірному випадку плоска проста (яка не має самоперетину) замкнута крива розбиває площину на дві зв'язні компоненти, одну з яких можна вважати внутрішньою, а іншу – зовнішньою.

У тривимірному просторі (аналогічно до двовимірному випадку) проста замкнена поверхня розбиває простір на дві зв'язні компоненти: внутрішню та зовнішню. Отже, можна вважати, що проста обмежена замкнена множина $L \subset \mathbf{R}^n$ подає певну форму об'єкта Ω (L є межею Ω). Одночасно, для будь-якої замкнутої множини $L \in \mathbf{R}^n$ можна побудувати неперервне неявне рівняння [116], найпростіший варіант якого – нормальне рівняння такого вигляду:

$$F(X) \equiv \inf_{Y \in L} \|X - Y\| = 0, \quad X \in \mathbf{R}^n.$$

Функція

$$\text{dist}(X, L) = \inf_{Y \in L} \|X - Y\| \quad (2.31)$$

є відстанню від точки $X \in \mathbf{R}^n$ до найближчої до неї точки множини $\mathbf{L}$. Ця функція у довільно обраній точці X приймає значення, що дорівнює радіусу найбільшої гіперсфери з центром в X , повністю укладеної (вписаної) в $\mathbf{L}$, якщо X – внутрішня точка $\mathbf{L}$, і дорівнює радіусу найменшої гіперсфери з центром в X , що дотична до $\mathbf{L}$, якщо X – зовнішня точка $\mathbf{L}$.

Припустимо, що дана функція

$$\text{sign}(X, \mathbf{L}) = \begin{cases} +1, & X \in \Omega, \\ -1, & X \notin \Omega, \end{cases} \quad (2.32)$$

яка дорівнює $+1$ у внутрішніх точках $\mathbf{L}$ і дорівнює -1 у зовнішніх точках $\mathbf{L}$. Відповідно, функція

$$f(X, \mathbf{L}) = \text{sign}(X, \mathbf{L}) \text{dist}(X, \mathbf{L}) \quad (2.33)$$

буде додатною у внутрішніх точках $\mathbf{L}$, дорівнюватиме нулю, якщо $X \in \mathbf{L}$, і від’ємна у зовнішніх точках $\mathbf{L}$. Така функція буде неявно визначати форму об’єкта Ω . Отже, завдання полягає в побудові функцій (2.31) і (2.32) за умови, що $\mathbf{L}$ – подана параметрично замкнена межа Ω .

2.4.1 Двовимірні форми

Нехай проста замкнена межа деякої двовимірної області подана параметрично у такому вигляді:

$$\text{curve}(t) = \mathbf{p} + x(t)\mathbf{i} + y(t)\mathbf{j}, \quad t_0 \leq t \leq t_1, \quad (2.34)$$

де t – скалярний аргумент;

$x(t)$ і $y(t)$ – неперервні функції параметра t ;

$\mathbf{p}=(x_p; y_p)$ – радіус-вектор довільної точки двовимірного простору (точка прив'язки для характерної точки кривої);

$\{\mathbf{i}, \mathbf{j}\}$ – ортонормований базис декартової системи координат двовимірного простору.

Найбільш поширеними на практиці прикладами примітивів-кривих вигляду (2.34) є:

$$\text{segment}(t)=(1-t)\mathbf{p}_0+t\mathbf{p}_1, \quad 0\leq t\leq 1, \quad (2.35)$$

$$\text{circle}(t)=\mathbf{c}+r\cos(t)\mathbf{i}+r\sin(t)\mathbf{j}, \quad 0\leq t\leq 2\pi, \quad 0<r, \quad (2.36)$$

$$\text{ellipse}(t)=\mathbf{c}+a\cos(t)\mathbf{i}+b\sin(t)\mathbf{j}, \quad 0\leq t\leq 2\pi, \quad 0<a, 0<b, \quad (2.37)$$

$$\text{cbezier}(t)=\sum_{i=0}^n B_i^n(t)\mathbf{p}_i=\sum_{i=0}^n \frac{n!}{i!(n-i)!} t^i(1-t)^{n-i} \mathbf{p}_i, \quad 0\leq t\leq 1, \quad (2.38)$$

де функція (2.35) визначає відрізок між точками з радіус-векторами $\mathbf{p}_0$ і $\mathbf{p}_1$;

функція (2.36) – коло радіуса r з центром в точці з радіус-вектором $\mathbf{c}$;

функція (2.37) – еліпс з центром в точці з радіус-вектором $\mathbf{c}$ і півсями, рівними a і b ;

функція (2.38) – крива Безьє, побудована на послідовності радіус-векторів $\mathbf{p}_0, \dots, \mathbf{p}_n$, які визначають $n+1$ контрольну точку.

Крива $\text{curve}(t)$, також може бути сплайном або складовою. Найпростішим прикладом останньої є ламана, побудована на послідовності радіус-векторів її вершин $\mathbf{p}_0, \dots, \mathbf{p}_n$:

$$\text{segments}(t)=(1-w)\mathbf{p}_i+w\mathbf{p}_{i+1}, \quad w=t-i, \quad i=\lfloor t \rfloor, \quad 0\leq t\leq n, \quad (2.39)$$

де $\lfloor t \rfloor$ – операція знаходження найбільшого цілого, меншого, ніж t .

Якщо в формулах (2.38) і (2.39) $\mathbf{p}_0=\mathbf{p}_n$, то відповідні криві будуть замкненими.

Знаходження відстані від довільної точки простору до найближчої точки кривої – задача досить екстенсивна, тому замінімо криву $\text{curve}(t)$ її дискретної моделлю для наближеного обчислення функції (2.31).

Нехай дискретна модель кривої – це плоский граф, поданий списком ребер у такому вигляді:

$$\mathbf{E}_2 = \{e_i = (\mathbf{p}_{i0}; \mathbf{p}_{i1})\}, \quad i = \overline{1, n},$$

де $\mathbf{p}_{i0}$ і $\mathbf{p}_{i1}$ – відповідно, координати (радіус-вектори) початку та кінця ребра e_i ;
 n – кількість ребер.

У такому випадку функція (2.31) отримає вигляд

$$\text{dist}(x, y, \text{curve}(t)) = \text{dist}(\mathbf{p}, \text{curve}(t)) \approx \text{dist}_2(\mathbf{p}, \mathbf{E}_2) = \min_{e_i = (\mathbf{p}_{i0}; \mathbf{p}_{i1}) \in \mathbf{E}_2} d_2(\mathbf{p}, \mathbf{p}_{i0}, \mathbf{p}_{i1}), \quad (2.40)$$

де $\mathbf{p} = (x; y) \in \mathbf{R}^2$ – довільна точка площини;

$d_2(\mathbf{p}, \mathbf{p}_{i0}, \mathbf{p}_{i1})$ – функція для визначення відстані від точки $\mathbf{p}$ до відрізка, заданого точками $\mathbf{p}_{i0}$ і $\mathbf{p}_{i1}$ (вершини ребра $e_i = (\mathbf{p}_{i0}; \mathbf{p}_{i1})$).

Функцію $d_2(\mathbf{p}, \mathbf{p}_{i0}, \mathbf{p}_{i1})$ можна задати такою формулою:

$$d_2(\mathbf{p}, \mathbf{p}_{i0}, \mathbf{p}_{i1}) = \begin{cases} \|\mathbf{p} - \mathbf{p}_{i0}\|, & \text{якщо } (\mathbf{p} - \mathbf{p}_{i0}) \cdot (\mathbf{p}_{i1} - \mathbf{p}_{i0}) \leq 0, \\ \|\mathbf{p} - \mathbf{p}_{i1}\|, & \text{якщо } (\mathbf{p}_{i1} - \mathbf{p}_{i0}) \cdot (\mathbf{p}_{i1} - \mathbf{p}_{i0}) \leq (\mathbf{p} - \mathbf{p}_{i0}) \cdot (\mathbf{p}_{i1} - \mathbf{p}_{i0}), \\ \left\| \mathbf{p} - \left(\mathbf{p}_{i0} + \frac{(\mathbf{p} - \mathbf{p}_{i0}) \cdot (\mathbf{p}_{i1} - \mathbf{p}_{i0})}{(\mathbf{p}_{i1} - \mathbf{p}_{i0}) \cdot (\mathbf{p}_{i1} - \mathbf{p}_{i0})} (\mathbf{p}_{i1} - \mathbf{p}_{i0}) \right) \right\|, & \text{інакше.} \end{cases} \quad (2.41)$$

Варто відзначити, що функцію (2.41) також можна використовувати для знаходження відстані від точки до відрізка в тривимірному випадку [427].

Очевидно, що зі збільшенням числа ребер в графі $\mathbf{E}_2$ точність формули (2.40) буде збільшуватися. Однак кривина кривої $\text{curve}(t)$ може бути змінною, отже, для досягнення найкращої апроксимації необхідно забезпечити згущення ребер в області найбільшої кривини. У такому випадку можна скористатися наступним алгоритмом.

algorithm nonuniform-curve-mesh

input:

$\text{curve}(t)$, $t_0 \leq t \leq t_1$ – формула кривої, що описує границю області

$n \geq 2$ – кількість вузлів в початковій дискретній моделі контуру

δ – максимальне відхилення середини ребра від границі

output:

E_2 – дискретна модель кривої

begin

$$E_t \leftarrow \{e_i = (t_{i0}; t_{i1})\}, \quad t_{i0} = t_0 + ih, \quad t_{i1} = t_0 + (i+1)h, \quad h = \frac{t_1 - t_0}{n-1}, \quad i = \overline{0, n-2}$$

repeat

$f \leftarrow \text{false}$

for each $e_i \in E_t$ **do**

$$\text{if } \frac{\left\| \frac{\text{curve}(t_{i0}) + \text{curve}(t_{i1})}{2} - \text{curve}\left(\frac{t_{i0} + t_{i1}}{2}\right) \right\|}{\|\text{curve}(t_{i1}) - \text{curve}(t_{i0})\|} > \delta \text{ then}$$

$$E_t \leftarrow E_t \setminus e_i$$

$$E_t \leftarrow E_t \cup \left(t_{i0}; \frac{t_{i0} + t_{i1}}{2}\right)$$

$$E_t \leftarrow E_t \cup \left(\frac{t_{i0} + t_{i1}}{2}; t_{i1}\right)$$

$f \leftarrow \text{true}$

end if

end for

until $f = \text{true}$

$$E_2 \leftarrow \{e_i = (\text{curve}(t_{i0}); \text{curve}(t_{i1}))\}, \quad i = \overline{1, |E_t|}$$

return E_2

В результаті буде побудована адаптивна дискретна модель межі області, яку можна використовувати для апроксимації за формулою (2.40). Наприклад, область, яка обмежена замкнутою кривою Безьє, що побудована на послідовності з 10 контрольних точок: $\mathbf{p}_0 = (0; 0)$, $\mathbf{p}_1 = (0; 3)$, $\mathbf{p}_2 = (1; 3)$, $\mathbf{p}_3 = (1; 1)$, $\mathbf{p}_4 = (4; 1)$, $\mathbf{p}_5 = (4; -1)$, $\mathbf{p}_6 = (1; -1)$, $\mathbf{p}_7 = (1; -3)$, $\mathbf{p}_8 = (0; -3)$ і $\mathbf{p}_9 = (0; 0)$, подається дискретною моделлю, зображеною на рис. 2.19 (вважається, що $t_0 = 0$, $t_1 = 1$, $n = 41$ і $\delta = 0,02$).

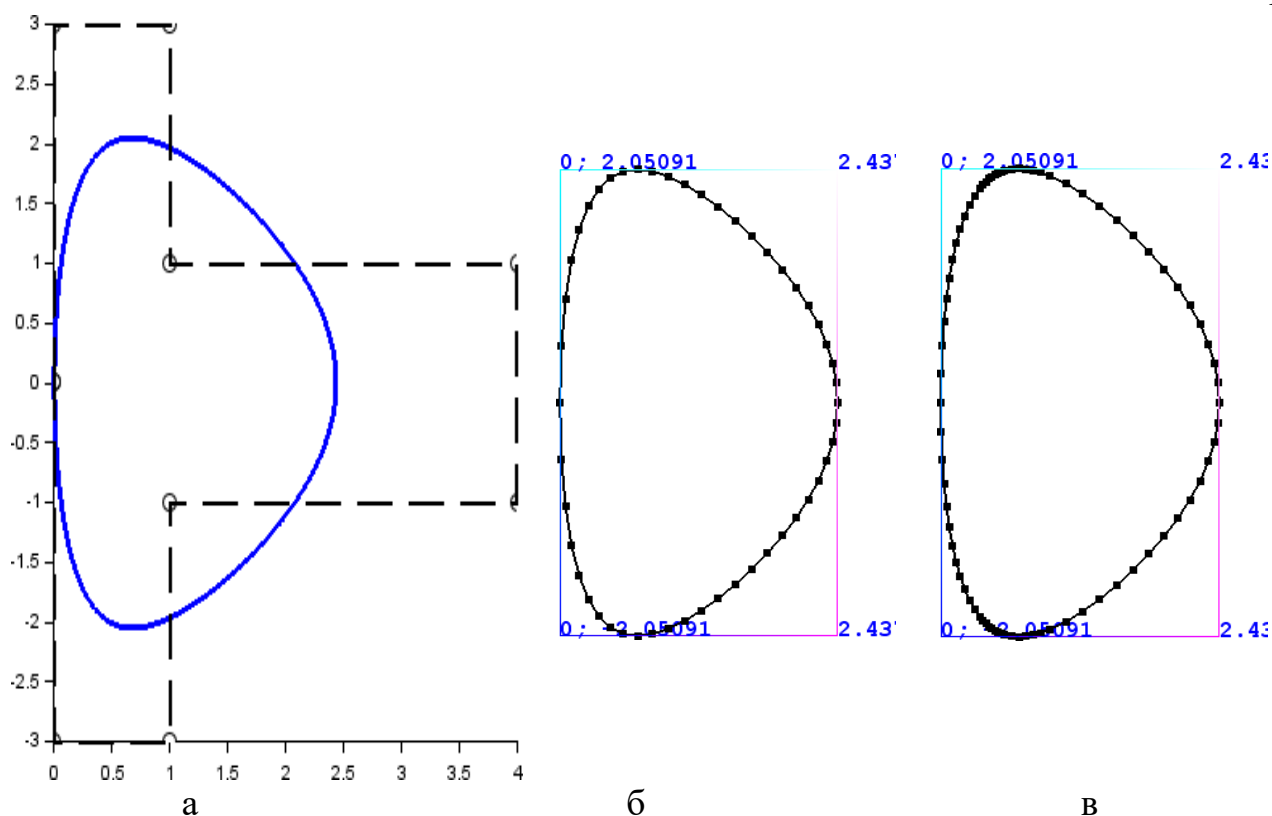


Рисунок 2.19 – Замкнена крива Безьє:

а – вихідна крива та її контрольні точки

б – початкова дискретна модель контура

в – адаптивна дискретна модель контура

Граф E_2 , власне, є замкнутим багатокутником, який апроксимує криву. Відповідно до теореми Жордана для побудови функції (2.32) можна скористатися тестом парності, можливо, перша реалізація якого розглянута в роботі [443]. Він полягає в тому, що з точки P випускають промінь, наприклад, в позитивному напрямку осі Ox . Далі для визначення належності точки багатокутника досить підрахувати число c – кількість перетинів променя та ребер графа. Єдиний вироджений випадок буде, якщо промінь перетинає ребра у вершині. У такому випадку на практиці використовують такий прийом: ребро вважають пересіченим, якщо одна його вершина строго нижче променя, а друга вище або лежить на промені. В результаті функція (2.32) отримає такий вигляд:

$$\text{sign}(x, y, \text{curve}(t)) = \text{sign}(\mathbf{p}, \text{curve}(t)) \approx \text{sign}_2(\mathbf{p}, E_2) = \begin{cases} +1, & c \text{ парне,} \\ -1, & c \text{ непарне.} \end{cases} \quad (2.42)$$

Отже, абсолютне значення функції (2.33) можна обчислити з використанням формул (2.40) і (2.41) з деякою точністю. Точність визначають параметри метода: кількістю точок у початковій дискретній моделі контуру n і параметр апроксимації δ . Водночас знак цієї функції можна обчислити за допомогою тесту парності (2.42).

Наприклад, для замкнутої кривої Безьє, зображеної на рис. 2.19, розподіл значень апроксимації функції (2.33), якщо $n=41$ і $\delta=0,02$, матиме вигляд, наведений на рис. 2.20.

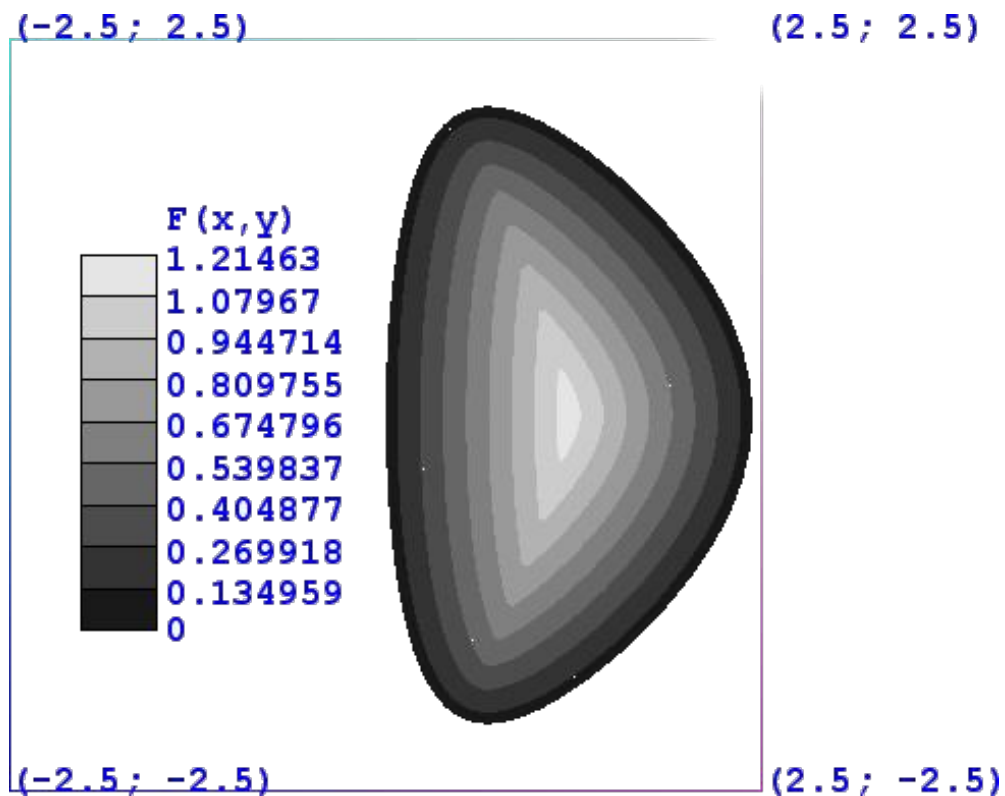


Рисунок 2.20 – Розподіл значень функції (2.33) в області,

яка обмежена замкнутою кривою Безьє

Якщо вихідною кривою буде, наприклад, багатокутник, визначений замкнутою ламаною (2.39) з послідовністю вершин $\mathbf{p}_0=(0; 0)$, $\mathbf{p}_1=(3; 0)$, $\mathbf{p}_2=(3; 2)$, $\mathbf{p}_3=(2; 2)$, $\mathbf{p}_4=(2; 1)$, $\mathbf{p}_5=(1; 1)$, $\mathbf{p}_6=(1; 4)$, $\mathbf{p}_7=(2; 4)$, $\mathbf{p}_8=(2; 3)$, $\mathbf{p}_9=(3; 3)$, $\mathbf{p}_{10}=(3; 5)$, $\mathbf{p}_{11}=(0; 5)$, $\mathbf{p}_{12}=(0; 0)$, $t_0=0$, $t_1=12$ і $n=13$ (значення параметра δ у даному випадку не впливає на результат), то розподіл значень функції (2.33) матиме вигляд, наведений на рис. 2.21.

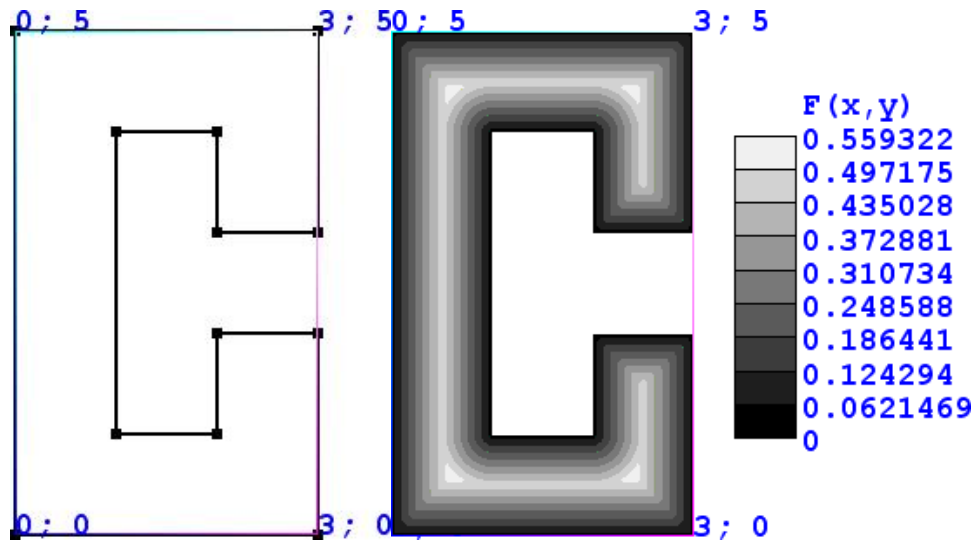


Рисунок 2.21 – Розподіл значень функції (2.33) для багатокутника

В роботах [264, 390, 419] розглянуті особливості моделювання профілів крил з використанням кривих Безьє. Отримані у цих роботах результати дозволяють досліджувати аеродинамічні властивості, але у задачах дослідження міцності необхідно враховувати також технологічні отвори. Наприклад, симетричний профіль NASA 0011 [264] (рис. 2.22) можна подати складеною кривою

$$\text{nasa}_{0011}(t) = \begin{cases} \sum_{i=0}^n \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \mathbf{t}_i, & 0 \leq t \leq 1, \\ \sum_{i=0}^n \frac{n!}{i!(n-i)!} (t-1)^i (2-t)^{n-i} \mathbf{b}_i, & 1 < t \leq 2, \end{cases}$$

де $n=5$ – степені кривих;

$\mathbf{t}_0=(0; 0)$, $\mathbf{t}_1=(1,14034; 2,44284)$, $\mathbf{t}_2=(11,1025; 3,65375)$, $\mathbf{t}_3=(30,8809; 1,13309)$, $\mathbf{t}_4=(39,9276; 0,09712)$, $\mathbf{t}_5=(40; 0)$ – контрольні точки кривої Безьє п'ятого степеня, які відповідають верхньому вигину крила;

$\mathbf{b}_0=(0; 0)$, $\mathbf{b}_1=(1,14034; -2,44284)$, $\mathbf{b}_2=(11,1025; -3,65375)$, $\mathbf{b}_3=(30,8809; -1,13309)$, $\mathbf{b}_4=(39,9276; -0,09712)$, $\mathbf{b}_5=(40; 0)$ – контрольні точки кривої Безьє п'ятого степеня, які відповідають нижньому вигину крила.

Варто зазначити, що функції виду (2.33), побудовані для контурів двовимірних областей, можна використовувати як операнди (примітиви) R-операцій разом з

розглянутими вище неявними функціями. Наприклад, профіль крила з трьома технологічними отворами (рис. 2.23) можна подати такою функцією:

$$f_{naca}(x, y) = f(x, y, naca_{0011}(t)) \wedge \neg \text{rectangle}(x - x_1, y, w_1, h_1) \wedge \\ \wedge \neg \text{circle}(x - x_2, y, r_2) \wedge \neg \text{rectangle}(x - x_3, y, w_3, w_3),$$

де circle – функція (2.7);

rectangle – функція (2.10);

$x_1, x_2, x_3, w_1, h_1, r_2, w_3$ – параметри положення та розмірів отворів.

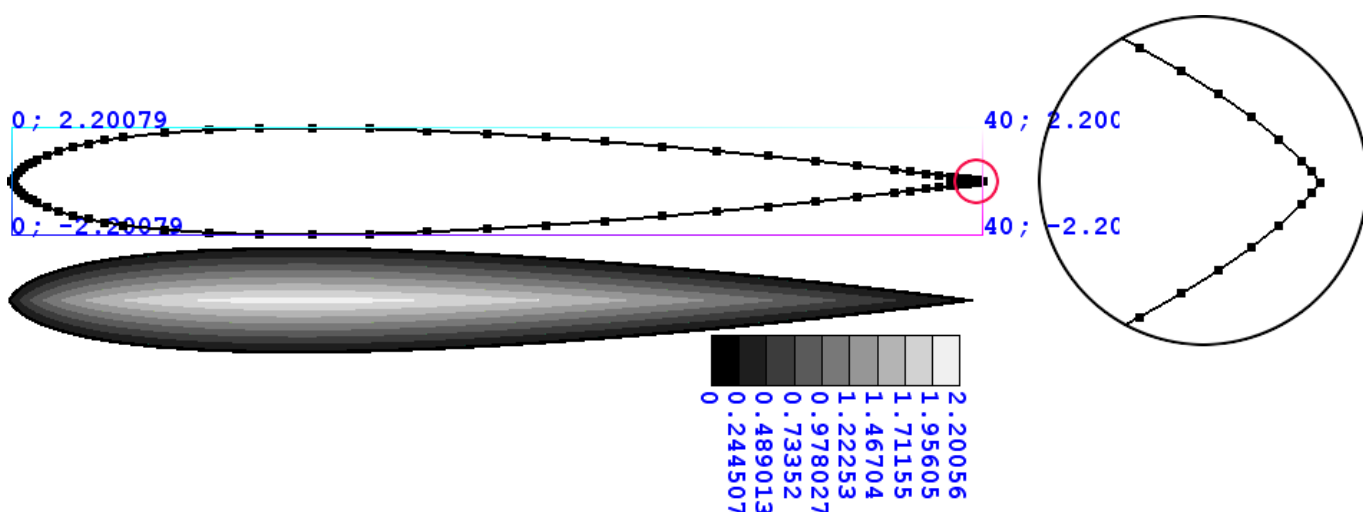


Рисунок 2.22 – Профіль NACA 0011: дискретна модель контуру та розподіл значень функції (2.33)

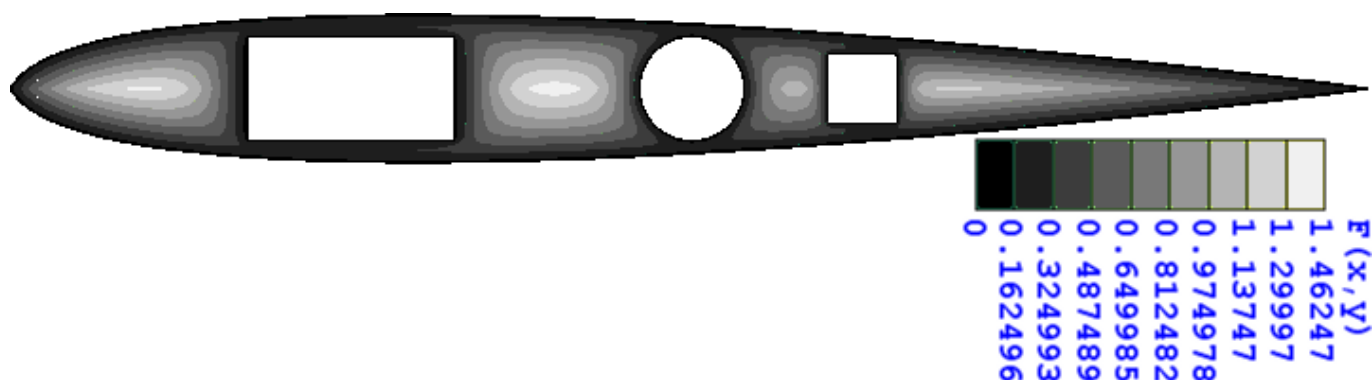


Рисунок 2.23 – Профіль NACA 0011 з технологічними отворами,

якщо $x_1=10, x_2=20, x_3=25, w_1=6, h_1=3, r_2=1,5, w_3=2$

2.4.2 Тривимірні форми

Нехай проста замкнена межа деякої тривимірної області задана m частинами (клаптями, патчами, англійською – patches), які є параметричними поверхнями:

$$\text{surface}_l(u, v) = \mathbf{q} + x(u, v)\mathbf{i} + y(u, v)\mathbf{j} + z(u, v)\mathbf{k}, \quad (2.43)$$

$$u_{l0} \leq u \leq u_{l1}, \quad v_{l0} \leq v \leq v_{l1} \quad l = \overline{1, m},$$

де u і v – скалярні аргументи;

$x(u, v), y(u, v), z(u, v)$ – неперервні функції параметрів u і v ;

$\mathbf{q} = (x_q; y_q; z_q)$ – радіус-вектор довільної точки тривимірного простору (точки прив'язки характерної точки поверхні);

$\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ – ортонормований базис декартової системи координат тривимірного простору.

Як у двовимірному випадку, частини $\text{surface}_j(u, v)$, також, можуть бути сплайнами або складеними поверхнями Безьє.

Для знаходження відстані від довільної точки простору до найближчої точки поверхні побудуємо дискретну модель межі області, використовуючи трикутні елементи. До такої дискретної моделі висувається тільки вимога узгодженості вузлів (і ребер в порядку їх обходу) елементів.

Нехай дискретна модель поверхні – це множина, що складається з трикутних елементів, нормаль до яких (в порядку обходу вузлів) спрямована назовні об'єкта:

$$\mathbf{E}_3 = \{tr_i = (\mathbf{q}_{i0}; \mathbf{q}_{i1}; \mathbf{q}_{i2})\}, \quad i = \overline{1, n},$$

де $\mathbf{q}_{i0}, \mathbf{q}_{i1}$ і $\mathbf{q}_{i2}$ – координати вузлів трикутника tr_i , впорядковані супроти годинникової стрілки.

У такому випадку при наближеному обчисленні формули (2.31) границю $\mathbf{L}$ можна замінити дискретною моделлю $\mathbf{E}_3$:

$$\text{dist}(x, y, z, \mathbf{L}) = \text{dist}(\mathbf{q}, \mathbf{L}) \approx \text{dist}_3(\mathbf{q}, \mathbf{E}_3) = \min_{t_i = (\mathbf{q}_{i0}; \mathbf{q}_{i1}; \mathbf{q}_{i2}) \in \mathbf{E}_3} d_3(\mathbf{q}, \mathbf{q}_{i0}, \mathbf{q}_{i1}, \mathbf{q}_{i2}), \quad (2.44)$$

де $\mathbf{q} = (x, y, z) \in \mathbf{R}^3$ – радіус-вектор довільної точки простору;

$d_3(\mathbf{q}, \mathbf{q}_{i0}, \mathbf{q}_{i1}, \mathbf{q}_{i2})$ – функція для визначення відстані від точки $\mathbf{q}$ до трикутника, заданого точками $\mathbf{q}_{i0}, \mathbf{q}_{i1}$ і $\mathbf{q}_{i2}$.

Функцію $d_3(\mathbf{q}, \mathbf{q}_{i0}, \mathbf{q}_{i1}, \mathbf{q}_{i2})$, яка використана у формулі (2.44), можна обчислити за допомогою такої формули:

$$d_3(\mathbf{q}, \mathbf{q}_{i0}, \mathbf{q}_{i1}, \mathbf{q}_{i2}) = \begin{cases} |l|, & v_0 \geq 0 \wedge v_1 \geq 0 \wedge v_2 \geq 0, \\ \|\mathbf{q} - \mathbf{q}_{i0}\|, & v_0 < 0 \wedge v_1 \geq 0 \wedge v_2 < 0, \\ \|\mathbf{q} - \mathbf{q}_{i1}\|, & v_0 < 0 \wedge v_1 < 0 \wedge v_2 \geq 0, \\ \|\mathbf{q} - \mathbf{q}_{i2}\|, & v_0 \geq 0 \wedge v_1 < 0 \wedge v_2 < 0, \\ d_2(\mathbf{q}, \mathbf{q}_{i0}, \mathbf{q}_{i1}), & v_0 < 0 \wedge v_1 \geq 0 \wedge v_2 \geq 0, \\ d_2(\mathbf{q}, \mathbf{q}_{i1}, \mathbf{q}_{i2}), & v_0 \geq 0 \wedge v_1 < 0 \wedge v_2 \geq 0, \\ d_2(\mathbf{q}, \mathbf{q}_{i2}, \mathbf{q}_{i0}), & v_0 \geq 0 \wedge v_1 \geq 0 \wedge v_2 < 0, \end{cases} \quad (2.45)$$

де $v_0 = \mathbf{n} \cdot (\mathbf{e}_{01} \times \mathbf{c}_0)$, $v_1 = \mathbf{n} \cdot (\mathbf{e}_{12} \times \mathbf{c}_1)$ і $v_2 = \mathbf{n} \cdot (\mathbf{e}_{20} \times \mathbf{c}_2)$ – значення, які використовуються для визначення взаємного розташування точки і вершин трикутника;

$\mathbf{e}_{01} = \mathbf{q}_{i1} - \mathbf{q}_{i0}$, $\mathbf{e}_{12} = \mathbf{q}_{i2} - \mathbf{q}_{i1}$ і $\mathbf{e}_{20} = \mathbf{q}_{i0} - \mathbf{q}_{i2}$ – вектори, утворені сторонами трикутника;

$\mathbf{n} = \mathbf{e}_{01} \times (-\mathbf{e}_{20})$ – зовнішня нормаль до площини трикутника;

$l = \frac{(\mathbf{q} - \mathbf{q}_{i0}) \cdot \mathbf{n}}{\|\mathbf{n}\|}$ – відстань від $\mathbf{q}$ до площини трикутника;

$\mathbf{b} = \mathbf{q} - \frac{l}{\|\mathbf{n}\|} \mathbf{n}$ – радіус-вектор проекції $\mathbf{q}$ на площину трикутника;

$\mathbf{c}_0 = \mathbf{b} - \mathbf{q}_{i0}$, $\mathbf{c}_1 = \mathbf{b} - \mathbf{q}_{i1}$ і $\mathbf{c}_2 = \mathbf{b} - \mathbf{q}_{i2}$ – вектори, що з'єднують вершини трикутника з вершиною радіус-вектора $\mathbf{b}$.

З точки зору раціональності використання необхідних для обчислення формули (2.44) комп'ютерних ресурсів, як і у двовимірному випадку, граф $\mathbf{E}_3$

повинен бути адаптивним до кривини границі. Побудову графа для множини параметричних клаптів можна виконати за допомогою такого алгоритму:

algorithm nonuniform-surface-mesh

input:

M – кількість частин поверхні

$\text{surface}_l(u, v)$, $u_{l0} \leq u \leq u_{l1}$, $v_{l0} \leq v \leq v_{l1}$, $l = \overline{1, M}$ – формула поверхні кожного клаптя

$n_l \geq 2$ – кількість вузлів в початковій дискретної моделі контуру уздовж абсциси параметричних координат

$m_l \geq 2$ – кількість вузлів в початковій дискретної моделі контуру уздовж ординати параметричних координат

δ – максимальне відхилення середини ребра від границі

output:

E_3 – дискретна модель границі

begin

$E_3 \leftarrow \emptyset$

for $l \leftarrow 1$ **to** M **do**

for $i \leftarrow 0$ **to** n_l **do**

for $j \leftarrow 0$ **to** m_l **do**

$$\mathbf{p}_{ij} \leftarrow \left(u_{l0} + i \frac{u_{l1} - u_{l0}}{n_l - 1}; v_{l0} + j \frac{v_{l1} - v_{l0}}{m_l - 1} \right)$$

end for

end for

$E_{3l} \leftarrow \emptyset$

for $i \leftarrow 0$ **to** $n_l - 1$ **do**

for $j \leftarrow 0$ **to** $m_l - 1$ **do**

if $\mathbf{p}_{ij} \neq \mathbf{p}_{i+1j}$ **and** $\mathbf{p}_{ij} \neq \mathbf{p}_{i+1j+1}$ **and** $\mathbf{p}_{i+1j} \neq \mathbf{p}_{i+1j+1}$ **then**

$$E_{3l} \leftarrow E_{3l} \cup (\mathbf{p}_{ij}; \mathbf{p}_{i+1j}; \mathbf{p}_{i+1j+1})$$

```

end if

if  $\mathbf{p}_{ij} \neq \mathbf{p}_{i+1j+1}$  and  $\mathbf{p}_{ij} \neq \mathbf{p}_{ij+1}$  and  $\mathbf{p}_{ij+1} \neq \mathbf{p}_{i+1j+1}$  then
     $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup (\mathbf{p}_{ij}; \mathbf{p}_{i+1j+1}; \mathbf{p}_{ij+1})$ 
end if

end for

end for

repeat
     $f \leftarrow \text{false}$ 
    for each  $t = (\mathbf{p}_0; \mathbf{p}_1; \mathbf{p}_2) \in \mathbf{E}_{3l}$  do
        if  $\frac{\left\| \frac{\text{surface}_l(\mathbf{p}_0) + \text{surface}_l(\mathbf{p}_1)}{2} - \text{surface}_l\left(\frac{\mathbf{p}_0 + \mathbf{p}_1}{2}\right) \right\|}{\|\text{surface}_l(\mathbf{p}_1) - \text{surface}_l(\mathbf{p}_0)\|} > \delta$  then
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \setminus t$ 
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_0; \frac{\mathbf{p}_0 + \mathbf{p}_1}{2}; \mathbf{p}_2 \right)$ 
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_2; \frac{\mathbf{p}_0 + \mathbf{p}_1}{2}; \mathbf{p}_1 \right)$ 
             $f \leftarrow \text{true}$ 
        else if  $\frac{\left\| \frac{\text{surface}_l(\mathbf{p}_1) + \text{surface}_l(\mathbf{p}_2)}{2} - \text{surface}_l\left(\frac{\mathbf{p}_1 + \mathbf{p}_2}{2}\right) \right\|}{\|\text{surface}_l(\mathbf{p}_2) - \text{surface}_l(\mathbf{p}_1)\|} > \delta$  then
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \setminus t$ 
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_1; \frac{\mathbf{p}_1 + \mathbf{p}_2}{2}; \mathbf{p}_0 \right)$ 
             $\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_0; \frac{\mathbf{p}_1 + \mathbf{p}_2}{2}; \mathbf{p}_2 \right)$ 
             $f \leftarrow \text{true}$ 
        else if  $\frac{\left\| \frac{\text{surface}_l(\mathbf{p}_2) + \text{surface}_l(\mathbf{p}_0)}{2} - \text{surface}_l\left(\frac{\mathbf{p}_2 + \mathbf{p}_0}{2}\right) \right\|}{\|\text{surface}_l(\mathbf{p}_0) - \text{surface}_l(\mathbf{p}_2)\|} > \delta$  then

```

```


$$\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \setminus t$$


$$\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_2; \frac{\mathbf{p}_2 + \mathbf{p}_0}{2}; \mathbf{p}_1 \right)$$


$$\mathbf{E}_{3l} \leftarrow \mathbf{E}_{3l} \cup \left( \mathbf{p}_1; \frac{\mathbf{p}_2 + \mathbf{p}_0}{2}; \mathbf{p}_0 \right)$$


$$f \leftarrow \text{true}$$

end if
end for
until  $f = \text{true}$ 

$$\mathbf{E}_{tl} = \left\{ tr_i = \left( \text{surface}(\mathbf{p}_{i0}); \text{surface}(\mathbf{p}_{i1}); \text{surface}(\mathbf{p}_{i2}) \right) \right\}, \quad i = 1, |\mathbf{E}_{3l}|$$


$$\mathbf{E}_3 = \mathbf{E}_3 \cup \mathbf{E}_{tl}$$

end for
return  $\mathbf{E}_3$ 

```

В результаті виконання алгоритму буде побудована адаптивна до кривини поверхні дискретна модель, яку можна використовувати для апроксимації за формулою (2.44). Деякі приклади адаптації при різних значеннях параметра δ показані на рис. 2.24.

Результатом роботи алгоритму nonuniform-surface-mesh, вхідними даного якого є частини замкненої обмеженої поверхні, буде замкнений багатогранник з трикутними гранями.

Для визначення знака функції відстані в точці $\mathbf{q} = (x_q; y_q; z_q)$ на першому кроці будуються багатокутники, утворені перетином багатогранника площиною $z = z_q$ (аналогічно двовимірному випадку, для виправлення можливих вироджених випадків, визначених дотиком площиною до трикутника, вважається, що трикутник перетинається площиною, якщо хоча б один вузол строго глибше та хоча б один ближче або лежить в площини $z = z_q$). Потім використовується процедура підрахунку числа перетинів променя та багатокутника, яка повністю еквівалентна двовимірному випадку. В результаті функція визначення знака набуде вигляду (2.42).

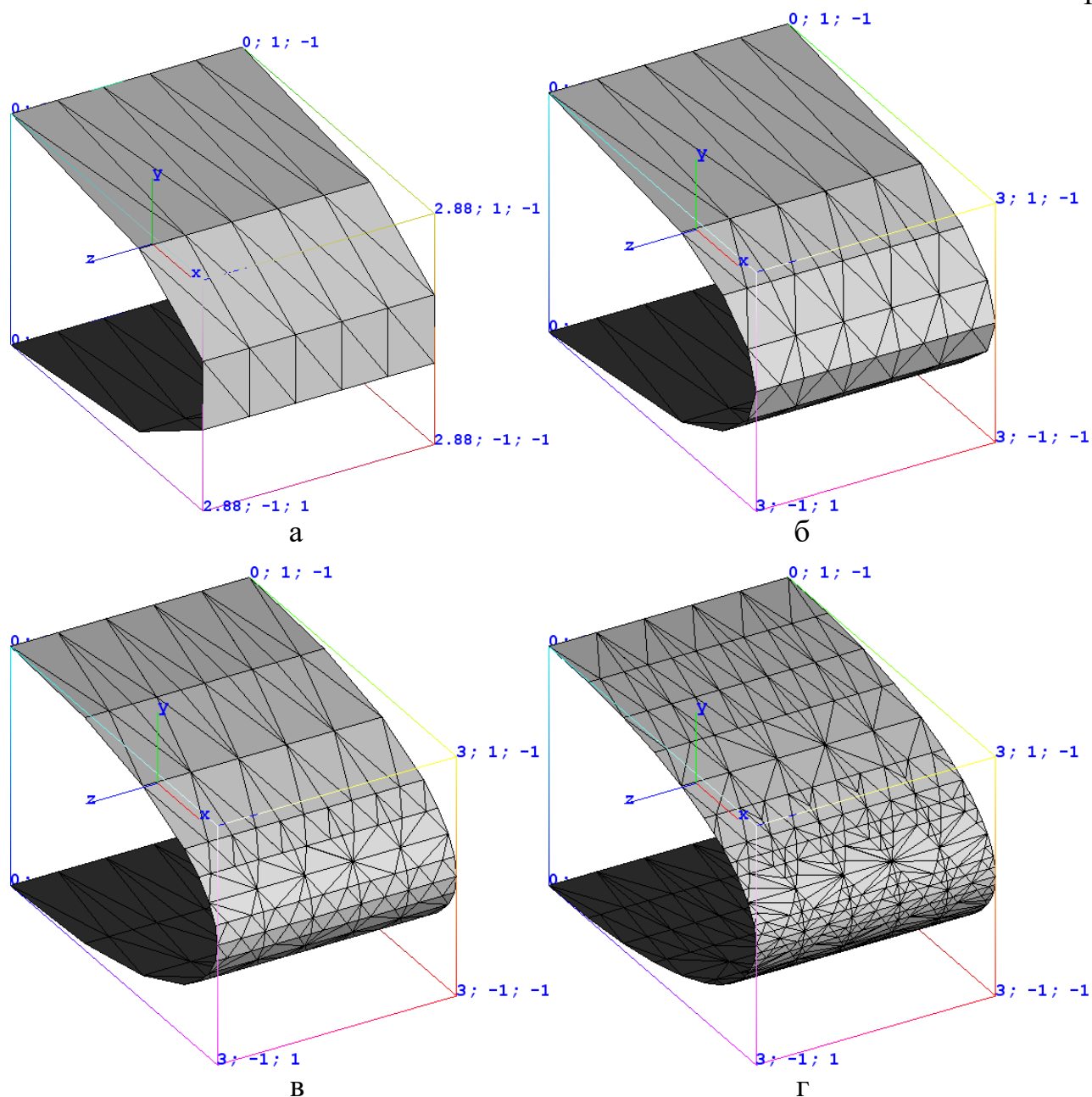


Рисунок 2.24 – Дискретизація поверхні, якщо $n_l=6$ і $m_l=6$:

а – без адаптації до кривини

б – адаптація до кривини поверхні, якщо $\delta=0,1$

в – адаптація до кривини поверхні, якщо $\delta=0,05$

г – адаптація до кривини поверхні, якщо $\delta=0,025$

Наприклад, розглянемо поверхню моделі літального апарату (рис. 2.25), визначеного шістьма частинами (поверхнями Безьє) з контрольними точками, заданими матрицями:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} (0; -4; 0) & (0; -3,9; 0,01) & (0; -1,1; 0,4) & (0; 0; 3) \\ (4; -2; 0) & (4; -1,9; 0,01) & (4; -0,1; 0,4) & (4; 0; 3) \\ (7; -3; 0) & (7; -2,9; 0,01) & (7; -0,1; 0,4) & (2; 0; 1) \\ (8; -1; 0) & (8; -0,9; 0) & (8; 0; 0) & (8; 0; 0) \end{bmatrix}, \\
\mathbf{B} &= \begin{bmatrix} (8; 1; 0) & (8; 0,9; 0) & (8; 0; 0) & (8; 0; 0) \\ (7; 3; 0) & (7; 2,9; 0,01) & (7; 0,1; 0,4) & (2; 0; 1) \\ (4; 2; 0) & (4; 1,9; 0,01) & (4; 0,1; 0,4) & (4; 0; 3) \\ (0; 4; 0) & (0; 3,9; 0,01) & (0; 1,1; 0,4) & (0; 0; 3) \end{bmatrix}, \\
\mathbf{C} &= \begin{bmatrix} (8; -1; 0) & (8; -0,9; 0) & (8; 0; 0) & (8; 0; 0) \\ (7; -3; 0) & (7; -2,9; -0,01) & (7; -0,1; -0,4) & (7; 0; -0,4) \\ (4; -2; 0) & (4; -1,9; -0,01) & (4; -0,1; -0,4) & (4; 0; -0,4) \\ (0; -4; 0) & (0; -3,9; -0,01) & (0; -1,1; -0,4) & (0; 0; -0,4) \end{bmatrix}, \\
\mathbf{D} &= \begin{bmatrix} (0; 4; 0) & (0; 3,9; -0,01) & (0; 1,1; -0,4) & (0; 0; -0,4) \\ (4; 2; 0) & (4; 1,9; -0,01) & (4; 0,1; -0,4) & (4; 0; -0,4) \\ (7; 3; 0) & (7; 2,9; -0,01) & (7; 0,1; -0,4) & (7; 0; -0,4) \\ (8; 1; 0) & (8; 0,9; 0) & (8; 0; 0) & (8; 0; 0) \end{bmatrix}, \\
\mathbf{E} &= \begin{bmatrix} (0; -4; 0) & (0; -3,9; -0,01) & (0; -1,1; -0,4) & (0; 0; -0,4) \\ (0; -4; 0) & (0; -3,9; 0,01) & (0; -1,1; 0,4) & (0; 0; 3) \end{bmatrix}, \\
\mathbf{F} &= \begin{bmatrix} (0; 4; 0) & (0; 3,9; 0,01) & (0; 1,1; 0,4) & (0; 0; 3) \\ (0; 4; 0) & (0; 3,9; -0,01) & (0; 1,1; -0,4) & (0; 0; -0,4) \end{bmatrix}.
\end{aligned}$$

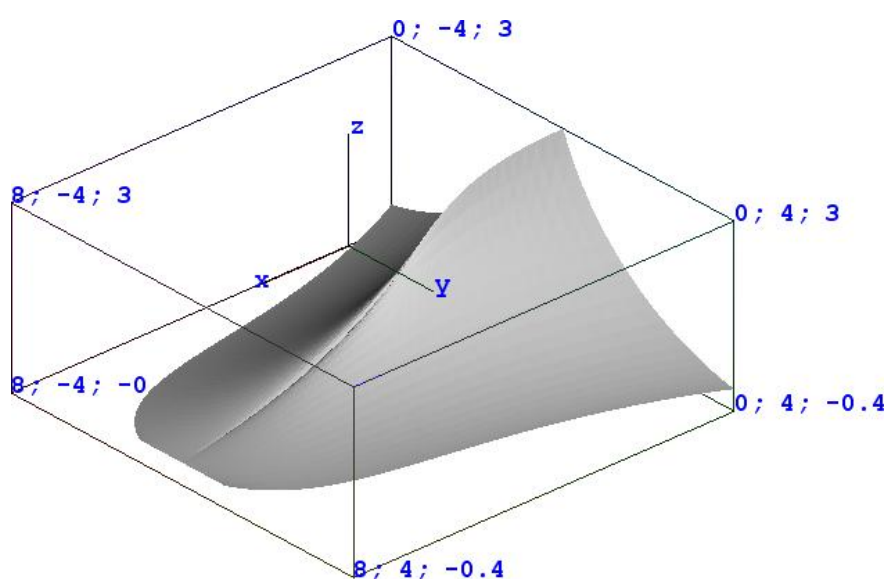


Рисунок 2.25 – Поверхня літального апарату

Поверхня кожного такого клаптя описується параметричною формулою [184]:

$$\text{sbezier}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) Q_{ij}, \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1, \quad (2.46)$$

де $B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$ – поліном Бернштейна;

Q_{ij} – елемент i -го рядку j -го стовпця матриці контрольних точок;

n і m – відповідно, кількість рядків і стовпців у матриці контрольних точок.

Розподіли значень функції відстані зі знаком для моделі літального апарату в перетинах площинами $y=0$ і $z=0$, відповідно, наведено на рис. 2.26 і 2.27.

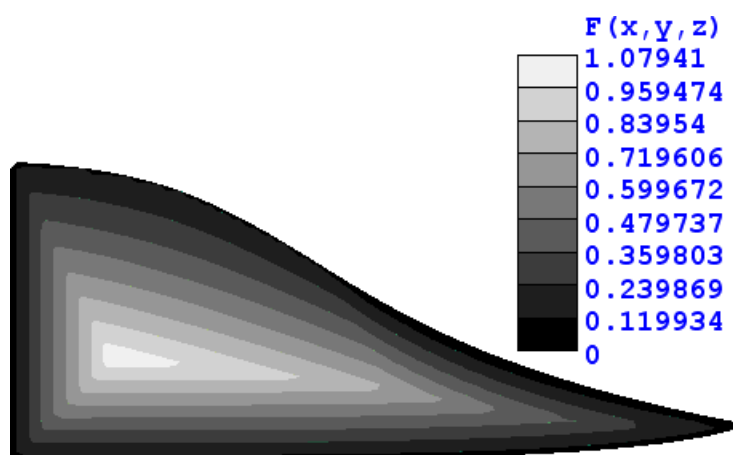


Рисунок 2.26 – Розподіл значень функції відстані зі знаком у перерізі площиною $y=0$

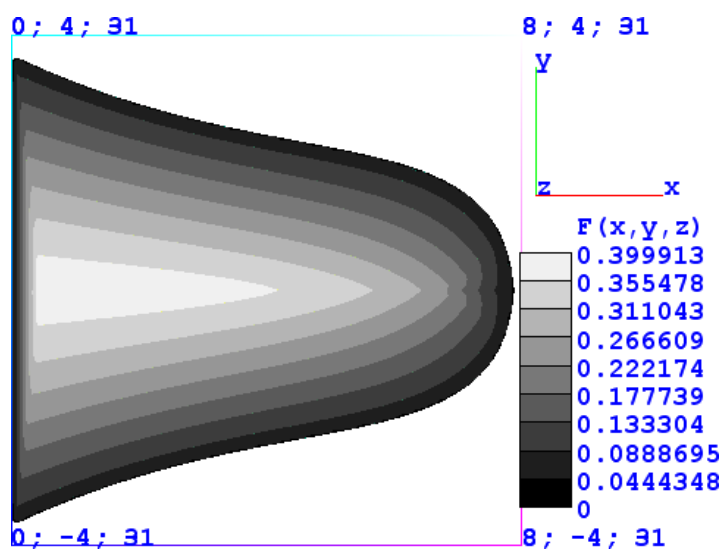


Рисунок 2.27 – Розподіл значень функції відстані зі знаком у перерізі площиною $z=0$

Аналогічно до двовимірного випадку, тривимірні функції відстані можуть виступати як операнди R-операцій для визначення більш складних об'єктів.

Наприклад, для подання моделі літального апарату у вигляді, наведеному на рис. 2.28, можна використати таку формулу:

$$F(x, y, z) = f(x, y, z, \text{surface}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F})) \wedge \text{Torp}(x; y-1,5; z) \wedge \text{Torp}(x; y+1,5; z), \quad (2.47)$$

$$\text{Torp}(x, y, z) = x \wedge (r^2(x) - y^2 - z^2) \wedge (4 - x),$$

$$r(x) = \begin{cases} 1 - 0,5x, & x \leq 1, \\ 0,5, & 1 < x \leq 3, \\ 0,5 - 0,5(x - 3), & 3 < x. \end{cases}$$

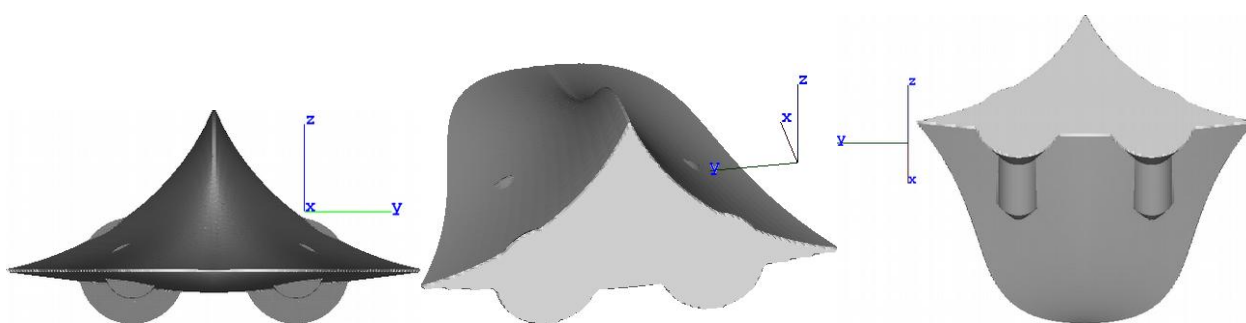


Рисунок 2.28 – Модель літального апарату

Розподіл значень функції (2.47) в перерізі площиною $z=0$ зображено на рис. 2.29.

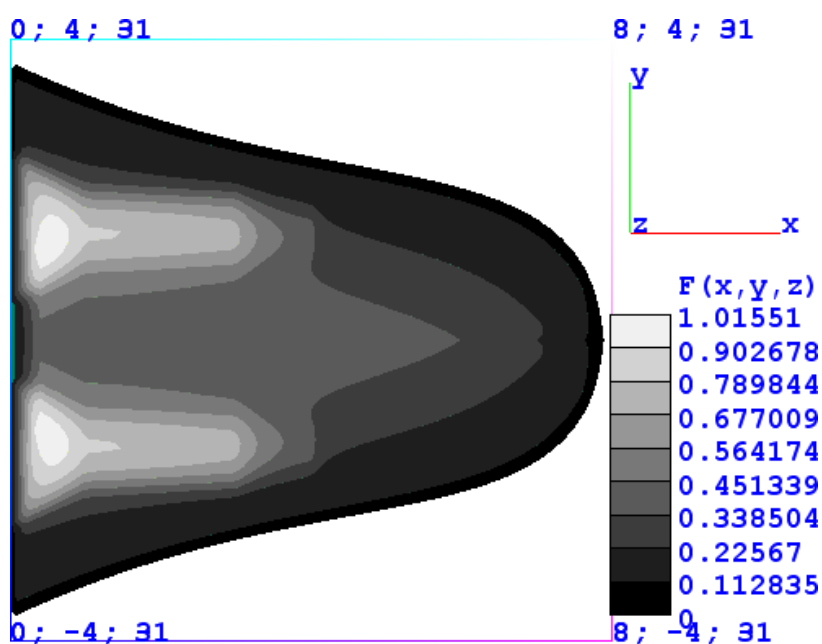


Рисунок 2.29 – Розподіл значень функції (2.47) у перерізі площиною $z=0$

Як альтернативу до формули (2.27), поверхню метричної різьби можна подати як результат руху правильного трикутника по спіралеподібній кривій (подібні поверхні в літературі також відомі як поверхні замітання [14]). Сторона трикутника дорівнює кроку різьблення. Кількість витків визначається відношенням висоти робочої поверхні (гайки або болта) до кроку різьби.

Нехай твірна крива – спіраль, радіус якої дорівнює внутрішньому радіусу різьби. Така крива подається формулою її радіус-вектора:

$$\mathbf{g}(v) = \mathbf{i} r_i \cos\left(\frac{2\pi v}{m}\right) + \mathbf{j} r_i \sin\left(\frac{2\pi v}{m}\right) + \mathbf{k} v, \quad 0 \leq v \leq L,$$

де $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ – ортонормований базис декартової системи координат тривимірного простору;

r_i – внутрішній радіус різьби;

m – крок різьби;

L – висота робочої поверхні.

Радіус-вектор твірної (правильний трикутник) подається формулою

$$\mathbf{p}(u) = \begin{cases} -\frac{m}{2}(1-u)\mathbf{i} + \left(t m \sin\left(\frac{\pi}{3}\right) - \delta\right)\mathbf{j}, & 0 \leq u \leq 1, \\ \frac{m}{2}(u-1)\mathbf{i} + \left[(2-u)m \sin\left(\frac{\pi}{3}\right) - \delta\right]\mathbf{j}, & 1 < u \leq 2, \\ \frac{m}{2}(5-2u)\mathbf{i} - \delta\mathbf{j}, & 2 < u \leq 3, \end{cases}$$

де $\delta = \frac{m \sin\left(\frac{\pi}{3}\right) - (r - r_i)}{2}$ – зсув трикутника;

r – зовнішній радіус різьби.

Радіус-вектор поверхні замітання, яка подає метричну різьбу, задається формулою

$$\text{surf}(u, v) = g(v) + m(v) \mathbf{A}(v) \times (p(u))^T, \quad 0 \leq u \leq 3, \quad 0 \leq v \leq L, \quad (2.48)$$

де $m(v) = \begin{cases} 5 \frac{v}{L}, & v \leq \frac{L}{5}, \\ 1, & \frac{L}{5} < v < \frac{4}{5} L, \\ 5 \left(1 - \frac{v}{L}\right), & \frac{4}{5} L \leq v, \end{cases}$ – залежність для масштабного коефіцієнта, що

забезпечує замкненість поверхні різьби (рис. 2.30);

$$\mathbf{A}(v) = \begin{bmatrix} u_1^T(v) & u_2^T(v) & u_3^T(v) \end{bmatrix} = \begin{bmatrix} u_{1,x} & u_{2,x} & u_{3,x} \\ u_{1,y} & u_{2,y} & u_{3,y} \\ u_{1,z} & u_{2,z} & u_{3,z} \end{bmatrix} \quad - \text{ матриця повороту локальної}$$

системи координат під час руху за траєкторією $g(v)$;

$$u_1(v) = r_i \cos(2\pi v) \mathbf{i} + r_i \sin(2\pi v) \mathbf{j};$$

$$u_2(v) = -\frac{2r_i\pi}{\sqrt{4\pi^2 r_i^2 + 1}} \sin(2\pi v) \mathbf{i} + \frac{2r_i\pi}{\sqrt{4\pi^2 r_i^2 + 1}} \cos(2\pi v) \mathbf{j} + \frac{1}{\sqrt{4\pi^2 r_i^2 + 1}} \mathbf{k};$$

$$u_3(v) = \frac{1}{\sqrt{4\pi^2 r_i^2 + 1}} \sin(2\pi v) \mathbf{i} - \frac{1}{\sqrt{4\pi^2 r_i^2 + 1}} \cos(2\pi v) \mathbf{j} + \frac{2r_i\pi}{\sqrt{4\pi^2 r_i^2 + 1}} \mathbf{k}.$$

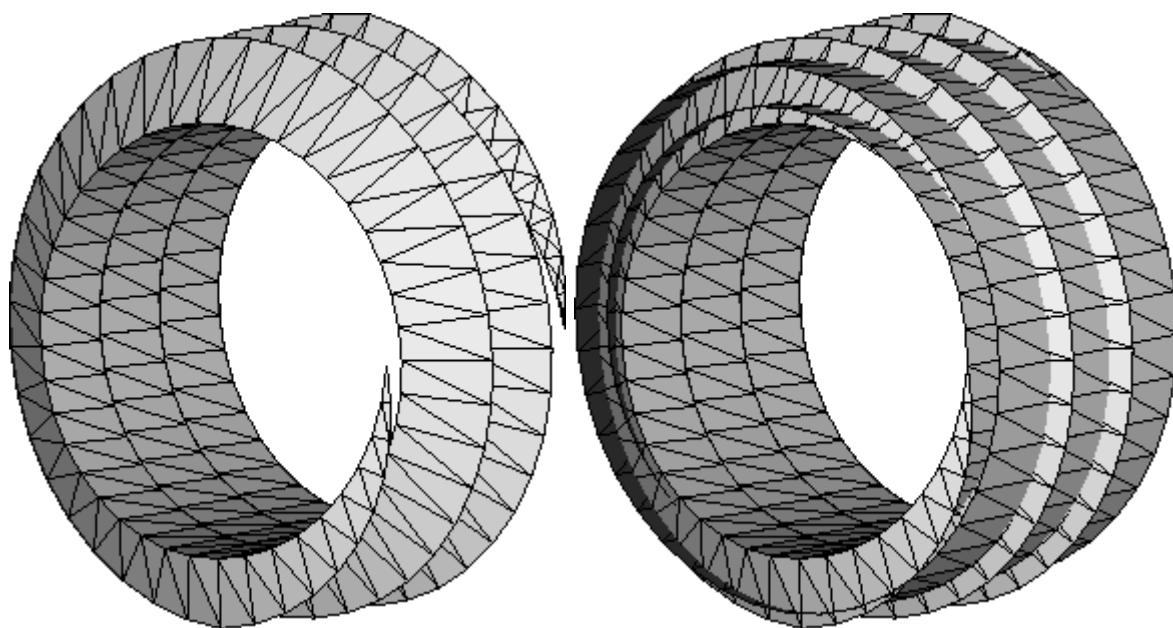


Рисунок 2.30 – Дискретна модель параметричної поверхні різьби

В результаті функція, що подає різьбу, у формулі (2.25), отримає вигляд

$$\text{thread}(x, y, z) = f(x, y, z, \text{surf}(u, v)).$$

Загальний вигляд отриманої моделі і розподіл значень функції (2.25) при параметричному поданні поверхні різьби показані на рис. 2.31.

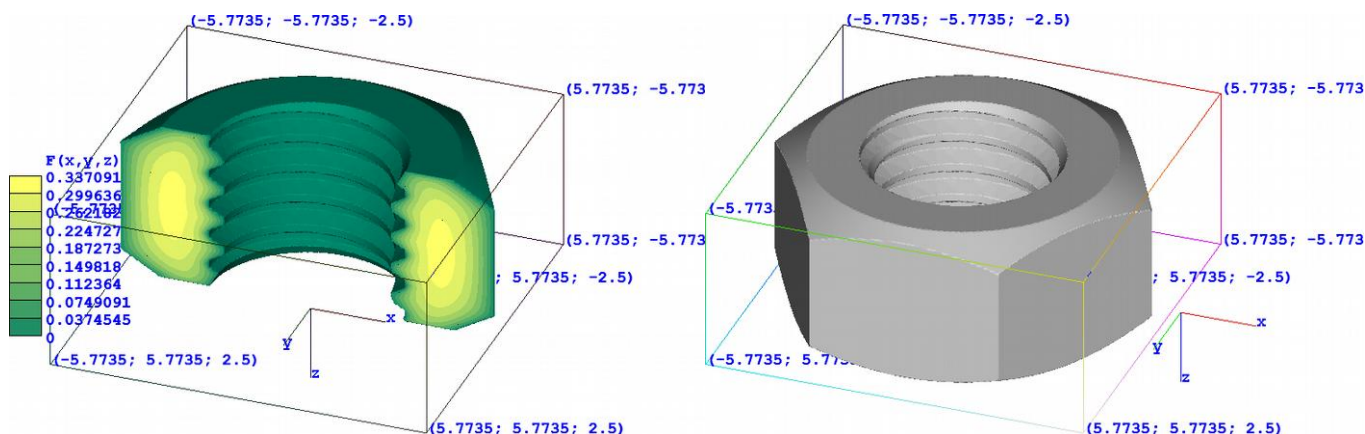


Рисунок 2.31 – Розподіл значень функції (2.25) при гібридному поданні гайки

2.5 Подання оболонкових об'єктів і конструкцій

Автоматизоване проектування багатьох об'єктів аерокосмічної, автомобіле- і суднобудівної галузей пов'язано з моделюванням тривимірних тіл, у яких один з розмірів (товщина) значно менше двох інших. Такі тіла з метою економії обчислювальних ресурсів модельно подають оболонками в формі параметричних поверхонь виду (2.43). Одночасно, подання оболонкових об'єктів, що володіють вирізами або технологічними отворами різної форми, за допомогою клаптів (параметричних поверхонь, що задають ділянки межі) не завжди зручне. У таких випадках модель оболонкового об'єкта можна розглядати у такому вигляді:

$$\Omega = \{(u, v) \in P \mid F(\text{surface}(u, v)) \geq 0\}, \quad (2.49)$$

де $\text{surface}(u, v)$ – параметрична поверхня, визначена на прямокутнику $P=\{(u, v) | a \leq u \leq b, c \leq v \leq d\}$;

$F(x, y, z)$ – неявна функція, яка від’ємна в точках, які не належать об’єкту (параметрична форма функції $\text{surface}(u, v)$ дозволяє явно перерахувати всі точки, що належать об’єкту, тому функцію $F(x, y, z)$ досить побудувати у вигляді, що дозволяє виключити точки, що потрапили в вирізи або отвори).

Наприклад, оболонку, яка задана поверхнею Безьє з отвором (рис. 2.32), отриманим логічним відніманням циліндра з направляючою уздовж осі абсцис і радіусом 0,4, можна подати такою парою функцій:

$$\begin{cases} \text{surface}(u, v) = \text{sbezier}(u, v), \\ F(x, y, z) = -\text{circle}(y; z; 0, 4), \end{cases} \quad (2.50)$$

де sbezier – параметрична функція (2.46) з матрицею контрольних точок

$$Q = \begin{bmatrix} (0; -1; -1) & (0; -1; 1) \\ (4; -1; -1) & (4; -1; 1) \\ (4; 1; -1) & (4; 1; 1) \\ (0; 1; -1) & (0; 1; 1) \end{bmatrix};$$

circle – функція (2.7).

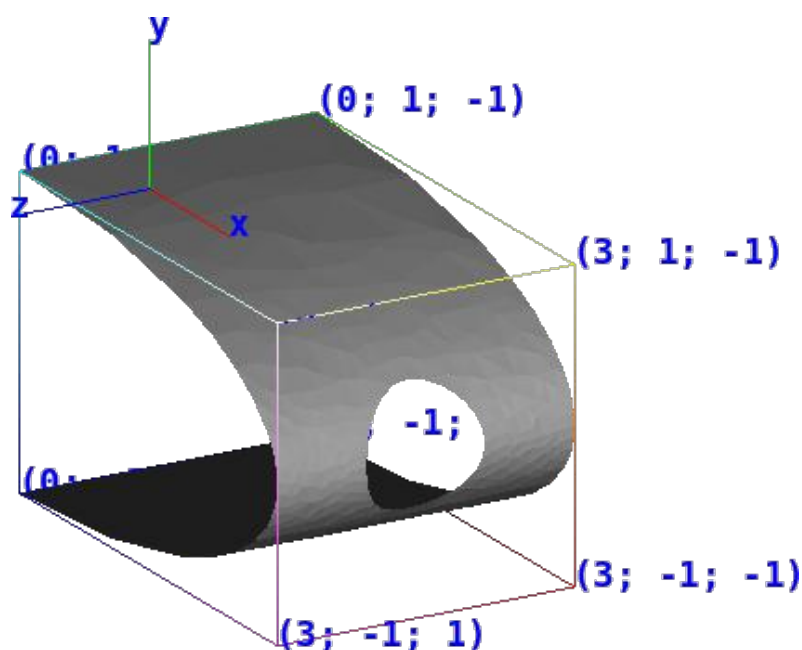


Рисунок 2.32 – Поверхня Безьє з отвором

Конічну оболонку зі складною конфігурацією технологічних отворів, висота якої спрямована уздовж осі Oy , можна подати у вигляді параметричної функції

$$\text{cone}(u, v) = [r_b + u(r_t - r_b)] \cos(2\pi v) \mathbf{i} + l u \mathbf{j} + [r_b + u(r_t - r_b)] \sin(2\pi v) \mathbf{k}, \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1,$$

де r_b – радіус нижньої основи;

r_t – радіус верхньої основи;

l – висота.

Технологічні отвори в конічній оболонці у формі прямокутників з округленими кутами (три з яких наскрізні), а також одним круговим отвором можна задати функцією

$$\begin{aligned} F(x, y, z) = & \text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_1), y - y_1, w_1, h_1, r_1) \vee \\ & \vee \text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_2), y - y_2, w_2, h_2, r_2) \vee \\ & \vee \text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_3), y - y_3, w_3, h_3, r_3) \vee \\ & \vee [\text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_4), y - y_4, w_4, h_4, r_4) \wedge \text{rot}_{z,y}(x, z, \alpha_4)] \vee \\ & \vee [\text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_5), y - y_5, w_5, h_5, r_5) \wedge \text{rot}_{z,y}(x, z, \alpha_5)] \vee \\ & \vee [\text{rectangle}_r(\text{rot}_{x,y}(x, z, \alpha_6), y - y_6, w_6, h_6, r_6) \wedge \text{rot}_{z,y}(x, z, \alpha_6)] \vee \\ & \vee [\text{circle}(\text{rot}_{x,y}(x, z, \alpha_7), y - y_7, r_7) \wedge \text{rot}_{z,y}(x, z, \alpha_7)], \end{aligned} \quad (2.51)$$

де $\text{rot}_{x,y}(x, z, \alpha) = x \cos(\alpha) + z \sin(\alpha)$;

$\text{rot}_{z,y}(x, z, \alpha) = -x \sin(\alpha) + z \cos(\alpha)$;

rectangle_r – функція (2.12);

circle – функція (2.7).

Наприклад, якщо вважати $\alpha_1 = -0,65$, $\alpha_2 = -1,31$, $\alpha_3 = -2,09$, $\alpha_4 = -1,31$, $\alpha_5 = 2,09$, $\alpha_6 = 0,65$, $\alpha_7 = -2,43$, $w_1 = h_1 = w_2 = h_2 = w_3 = 0,4$, $r_1 = r_2 = 0,055$, $r_3 = 0,065$, $w_4 = w_5 = 0,68$, $h_4 = h_5 = 0,45$, $r_4 = r_5 = r_6 = 0,075$, $w_6 = 0,44$, $h_6 = 0,88$, $y_1 = y_2 = 3,434$, $y_3 = 3,529$, $y_4 = y_5 = 0,52$, $y_6 = 0,7$, $y_7 = 3,554$, а також параметри конічної оболонки $r_b = 1,5$, $r_t = 1,99$ і $l = 4,014$, то отримаємо конфігурацію отворів у вигляді, наведеному на рис. 2.33.

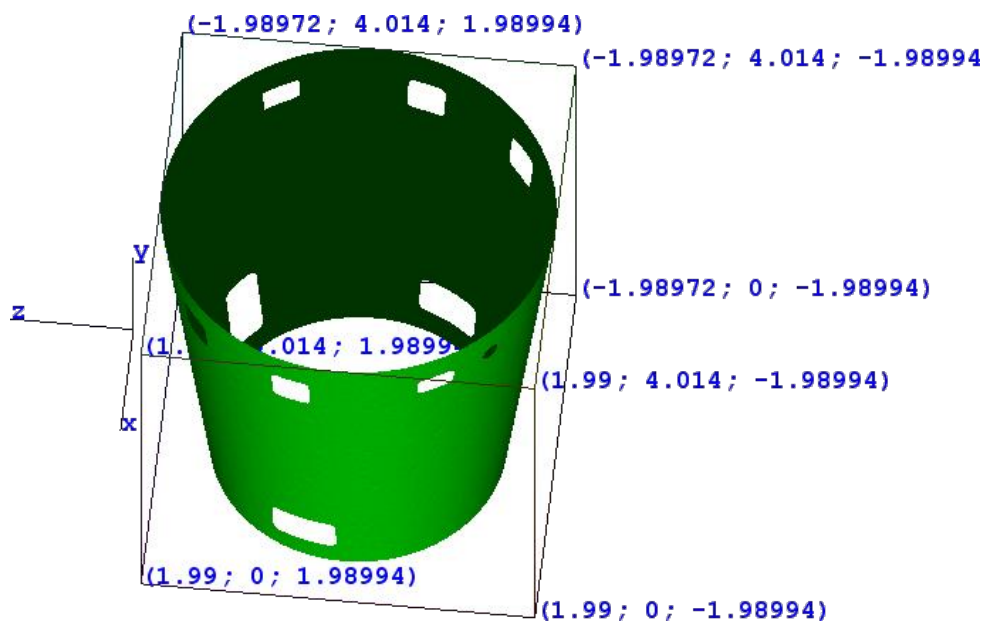


Рисунок 2.33 – Конічна оболонка з отворами

При такому підході процедури, пов'язані з обробкою математичної моделі об'єкта (наприклад, побудову та поліпшення дискретних моделей), можна виконувати в двовимірному параметричному просторі, зменшуючи, таким чином, розмірність задачі.

Висновки до розділу 2

Отже, типова задача побудови математичної моделі, що описує форму об'єкта машинобудування як вихідні дані має креслення. Далі, використовуючи функціональний підхід, можна побудувати модель об'єкта як єдину неявну функцію. Проте, у машинобудуванні зустрічаються об'єкти, форма яких описується (повністю або частково) параметричними функціями. У таких випадках побудова адекватних функціональних моделей засобами виключно неявних функцій є вельми складною. У результаті виникає необхідність одночасного використання примітивів, поданих неявно та параметрично, для формування підсумкової моделі форми об'єкта.

Запропонований математичний апарат дозволяє одночасно використовувати примітиви, межі яких визначені за допомогою параметричних і неявних функцій.

Водночас моделі на базі параметричних функцій можуть виступати як операнди R-операцій. Отримані моделі можна розглядати як гібридні, що поєднують функціональне подання на базі неявних функцій і граничне подання на базі параметричних функцій.

На відміну від робіт [46, 65–67, 69–76, 114–123], присвячених моделюванню форм об'єктів з використанням теорії неявних функцій і R-функцій, запропонований підхід дозволяє простіше описувати області, обмежені кривими або поверхнями Безьє, b-сплайнами або NURBS. Перехід від граничного подання форми до дискретної моделі дозволяє порівняно легко автоматизувати даний математичний апарат. Недоліком такого переходу є екстенсивність необхідних обчислень.

Наукова новизна результатів, отриманих в розділі, полягає в тому, що вперше запропоновано гібридний підхід до подання форм об'єктів з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання), що дозволяє зменшити час побудови математичних моделей. Практичне значення отриманих результатів полягає в тому, що розроблено програмне забезпечення, що реалізує запропонований метод, на основі якого розв'язані задачі побудови моделей складних об'єктів машинобудування. Перспективи подальших досліджень пов'язані з оптимізацією обчислення функцій відстані зі знаком, наприклад, за допомогою технологій паралельних обчислень.

Результати другого розділу опубліковано у роботах [82, 92, 112, 170–171, 182].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилення [14, 46, 73, 114–123, 381, 410, 414, 416, 434, 443].

3 ФОРМАЛІЗАЦІЯ ДИСКРЕТНИХ МОДЕЛЕЙ ФОРМ ОБ'ЄКТІВ У САПР МАШИНОБУДУВАННЯ

Моделі, які ґрунтуються на чисельному розв'язанні рівнянь в частинних похідних, часто використовують в задачах проектування та дослідження технічних об'єктів. Один з найбільш поширених на практиці чисельних методів – метод скінченних елементів – (як і деякі інші) замінює вихідну неперервну модель дискретною, яку можна обробити комп'ютером. В результаті на першому кроці моделювання для деякої неперервної області необхідно побудувати скінченну множину простих фігур (елементів), наприклад, трикутників, чотирикутників, тетраедрів, шестигранників тощо. Взаємне розташування вершин елементів істотно впливає на точність чисельного аналізу. Отже, автоматизація побудови дискретних моделей геометричних об'єктів є одним з ключових етапів реалізації САПР машинобудування.

Інтуїтивно, дискретна модель форми об'єкта – це колекція вузлів і елементів, яка визначає форму цього об'єкта. Координати вузлів визначають геометрію об'єкта, елементи – його топологію. У загальному випадку, елементи можуть мати тільки спільні вершини, ребра і грані. За такої умови для кожної пари вузлів, яка є ребром в одному елементі, має існувати ребро в іншому елементі, якщо обидва елементи інцидентні в цих вузлах. Аналогічну вимогу можна сформулювати щодо граней. Ці дві вимоги необхідні для забезпечення неперервності чисельного рішення, відповідно, вздовж ребер і граней.

Формальне визначення дискретної моделі форми об'єкта – це введення відповідної структури даних. Воно необхідне з точки зору, що пара алгоритм і структура даних є основною для створення програми. Водночас розробка загальної структури даних для довільної дискретної моделі є досить складною задачею.

Історично першими отримали поширення полігональні дискретні моделі поверхонь об'єктів. Для них розроблені досить ефективні програмні й апаратні методи візуалізації (наприклад, за допомогою зворотного трасування променів).

Багатокутник в порядку обходу однозначно визначається координатами своїх вершин. Відповідно, у задачах візуалізації для кожного полігону досить зберігати тільки координати його вершин. В результаті на кожен полігон буде виділятися $3kf$ байт пам'яті (k – кількість вершин, f – кількість байт, що використовується для зберігання одного числа з рухомою комою).

У задачах генерації й обробки дискретних моделей виникає необхідність вставки або видалення вузлів. У разі зберігання елементів у вигляді масивів координат необхідний пошук елементів зі спільними вузлами. В результаті, в системах інженерного аналізу поширення отримали структури даних, засновані на поданні вершин елементів вказівниками на позиції у масиві або списку координат вершин. Водночас елементи володіють додатковою топологічною інформацією, яка спрощує знаходження сусідів. Піонером у цьому напрямку можна вважати схему «крилате ребро» (winged edge) [207]. У ньому кожен вузол зберігає інформацію про свої координати та посилання на одне зі своїх ребер, грань зберігає посилання на ребра в порядку їх обходу. Ребро зберігає посилання на два вузла та дві грані (умовно ліву та праву), а також чотири «крила»: посилання на попередні та наступні ребра в порядку обходу обох граней. Схожі ідеї закладені в схеми «радіальне ребро» (radial-edge) [369], «орієнтоване ребро» (directed edge) [220], «півребро» (half-edge) [315], а також узагальнення останнього [189, 242, 445]. Подібні схеми досить ефективні під час зберігання та обробки полігональних моделей, але їх дуже проблематично застосовувати для об'ємних елементів.

Компактні схеми збереження об'ємних елементів розроблені для тетраедрів у вигляді узагальнення схем для трикутників [224, 245]. Одночасно, у роботі [303] запропоновано схему стисненого збереження дискретних моделей на основі шестигранників.

Розробка систем інженерного аналізу потребує єдиної абстрактної структури даних, яка не залежить від форми елемента. Її введення дозволить використовувати єдину формальну нотацію для викладання алгоритмів.

3.1 Формальне визначення дискретної моделі форми об'єкта машинобудування

Формально, дискретна модель форми об'єкта – це пара множин:

$$\mathbf{M} = (\mathbf{V}, \mathbf{E}), \quad (3.1)$$

де $\mathbf{V} = (v_{k_1}, v_{k_2}, \dots, v_{k_m})$ – множина координат вузлів;

$$\mathbf{E} = (e_1 = (k_{1,1}, k_{1,2}, \dots, k_{1,n_1}), e_2 = (k_{2,1}, k_{2,2}, \dots, k_{2,n_2}), \dots, e_q = (k_{q,1}, k_{q,2}, \dots, k_{q,n_q}))$$
 –

множина елементів, кожен з яких має $n_i = |e_i|$ вершин, – кортежів кодів вузлів з множини $\mathbf{V}$, що визначають положення вершин елемента;

m – кількість вузлів;

q – кількість елементів.

Координати j -го елемента з n вершинами визначаються підстановкою

$$\mathbf{V}(e_j) = (v_{k_{j,1}}, v_{k_{j,2}}, \dots, v_{k_{j,n}}).$$

Множину $\mathbf{V}$, у загальному випадку, можна вважати асоціативним масивом, який зберігає пари ключ-значення. Ключем виступає код вузла (номер, або адреса в пам'яті), значенням – його координати. Відповідно, запис $\mathbf{V}[k]$ позначає координати (радіус-вектор) вузла $v_k \in \mathbf{V}$.

Аналогічне твердження справедливе для множини $\mathbf{E}$, в якій ключами можуть виступати коди (номери або адреси) елементів, а значеннями – відповідні списки або масиви, що зберігають кортежі. В такому випадку запис $\mathbf{E}[i]$ позначає елемент $e_i \in \mathbf{E}$. Враховуючи, що e_i є кортежем, $e_i[j] = \mathbf{E}[i][j]$ позначає код вузла k_{ij} , який визначає положення j -ї вершини.

Для більш короткого позначення операцій щодо дискретної моделі $\mathbf{M}$ у квадратних дужках записані операції доступу до елементів множини $\mathbf{V}$, а у круглих дужках – до елементів множини $\mathbf{E}$. Тобто, $\mathbf{M}[k] = \mathbf{V}[k]$, $\mathbf{M}(i) = \mathbf{E}[i]$, $\mathbf{M}(i, j) = \mathbf{E}[i][j]$. У такому випадку $|\mathbf{M}(i)|$ – кількість вершин (вузлів) у елементі з кодом i .

3.2 Операції вставки та видалення, пошуку сусідів

Алгоритми вставки та видалення вузлів та елементів для дискретної моделі (3.1) можна віднести до елементарних. Додавання нового елементу потребує почергового додавання всіх його вершин. В свою чергу, вставка нового вузла потребує перевірки наявності вузла з такими ж координатами. Результатом операції вставки буде код вузла в множині V (новий або наявний, якщо такий само вузол був знайдений). При видаленні вузла також необхідно буде видалити інцидентні елементи, а під час видалення елемента необхідно видалити тільки вузли, які більше не беруть участь у зв'язках.

Для дискретної моделі, що складається з m вузлів і n елементів, кожен з яких має q вершин, операція вставки вузла може бути реалізована за допомогою лінійного алгоритму з обчислювальною складністю $O(m)$. Додавання елемента, заданого координатами власних вершин, q раз вимагатиме виконання вставки вузла та, відповідно, буде мати обчислювальну складність $O(q m)$. Для формування множини інцидентних деякому вузлу елементів досить буде одного проходу по елементам множини E , отже, обчислювальна складність цієї процедури буде дорівнювати n операцій. В результаті під час видалення одного елемента буде потрібно $n q$ операцій, а під час видалення вузла – $(k n + 1)q$ операцій (k – кількість інцидентних у вузлі елементів).

Очевидним недоліком дискретної моделі форми об'єкта (3.1) є необхідність використання лінійного пошуку для формування списків інцидентних елементів. Ці операції виникають під час видалення вузлів та елементів, а також поширені під час обробці дискретних моделей (наприклад, локальному згладжуванню). Для позбавлення від цього недоліку можна припустити, що кожен вузол крім своїх координат зберігає ще інформацію про множину суміжних елементів. В результаті дискретну модель форми об'єкта можна подати у такому вигляді:

$$M = (N, E), \quad (3.2)$$

де $\mathbf{N} = ((v_{k_1}, \mathbf{I}_{k_1}), (v_{k_2}, \mathbf{I}_{k_2}), \dots, (v_{k_m}, \mathbf{I}_{k_m}))$ – множина пар: v_{k_i} – координати вузла з кодом k_i та $\mathbf{I}_{k_i}$ – коди інцидентних йому елементів;

$\mathbf{E}$ – таке ж, як у (3.1).

Для дискретної моделі форми об'єкта (3.2) операції доступу можна визначити у такому вигляді: $\mathbf{M}[k] = \mathbf{N}[k] = v_k$, $\mathbf{M}(i) = \mathbf{E}[i]$, $\mathbf{M}(i, j) = \mathbf{E}[i][j]$. До цих операцій можна додати такі: $\mathbf{M}.\text{incident}(k) = \mathbf{I}_k$ – отримання множини кодів елементів, інцидентних у вузлі з кодом k , і $\mathbf{M}.\text{code}(n)$ – визначення коду параметра (вузла або елемента). Також припустимо, що для кожного $n \in \mathbf{N}$ запис $n.v$ відповідає координатам, $n.\mathbf{I}$ – множині інцидентних йому елементів.

Вставка вузла у дискретну модель (3.2) аналогічна описаній вище, але вимагає ініціалізації множин кодів інцидентних елементів порожніми. У результаті вставку вузла можна формалізувати таким алгоритмом:

algorithm insert_node

input:

$\mathbf{M}$ – дискретна модель форми об'єкта

p – координати вузла

ε – чисельний ноль

output:

k – код вузла

begin

for each $n \in \mathbf{M}.\mathbf{N}$ **do**

if $\|p - n.v\| < \varepsilon$ **then**

return $\mathbf{M}.\text{code}(n)$

end if

end for

$\mathbf{M}.\mathbf{N} \leftarrow \mathbf{M}.\mathbf{N} \cup \{(p, \emptyset)\}$

return $\mathbf{M}.\text{code}((p, \emptyset))$

Алгоритм вставки нового елемента, заданого списком координат власних вершин, в дискретну модель в обох випадках можна записати у такому вигляді:

algorithm insert_element

input:

$\mathbf{M}$ – дискретна модель форми об’єкта

$\mathbf{V}$ – список координат вершин

ε – чисельний нуль

begin

$(e, k_e) \leftarrow$ Новий елемент з $|\mathbf{V}|$ вершинами та його код у $\mathbf{M.E}$

for $i \leftarrow 1$ **to** $|\mathbf{V}|$ **do**

$e[i] \leftarrow \text{insert\_node}(\mathbf{M}, \mathbf{V}[i], \varepsilon)$

$\mathbf{M.incident}(e[i]) \leftarrow \mathbf{M.incident}(e[i]) \cup \{k_e\}$

end for

$\mathbf{M.E} \leftarrow \mathbf{M.E} \cup \{e\}$

end

Операція видалення елемента з дискретної моделі (3.2) потребує оновлення множин інцидентних елементів у вузлах, відповідних вершин. У результаті, цю операцію можна описати у вигляді такого алгоритму:

algorithm remove_element

input:

$\mathbf{M}$ – дискретна модель форми об’єкта

k_e – код елемента

begin

$e \leftarrow \mathbf{M}(k_e)$

for $i \leftarrow 1$ **to** $|e|$ **do**

$\mathbf{M.incident}(e[i]) \leftarrow \mathbf{M.incident}(e[i]) \setminus \{k_e\}$

if $\mathbf{M.incident}(e[i]) = \emptyset$ **then**

```

    remove_node(M,  $e[i]$ )
  end if
end for
M.E  $\leftarrow$  M.E  $\setminus$  { $e$ }
end

```

Операцію видалення вузла, яка використовується вище, можна реалізувати у вигляді такого алгоритму:

algorithm remove_node

input:

M – дискретна модель форми об’єкта
 k_n – код вузла

begin

I $\leftarrow$ **M.incident**(k_n)

for each $k_e \in$ **I** **do**

 remove_element(**M**, k_e)

end for

M.N $\leftarrow$ **M.N** $\setminus$ {**M.N**[k_n]}

end

Операції видалення можна вважати тісно зв’язними між собою, тому що, реалізація однієї з них залежить від реалізації іншої, що підкреслює важливість акуратності їх програмної реалізації.

Операція пошуку сусідніх вузлів ґрунтується на аналізі інцидентних елементів і для усунення неоднозначності потребує введення обмежень на порядок обходу вершин у визначенні елемента. Припустимо, що у двовимірних елементів порядок обходу вершин проти годинникової стрілки, а у тривимірних – у порядку обходу проти годинникової стрілки спочатку нижніх граней, а потім верхніх (з урахуванням обертання, де це може бути застосовано, рис. 3.1).

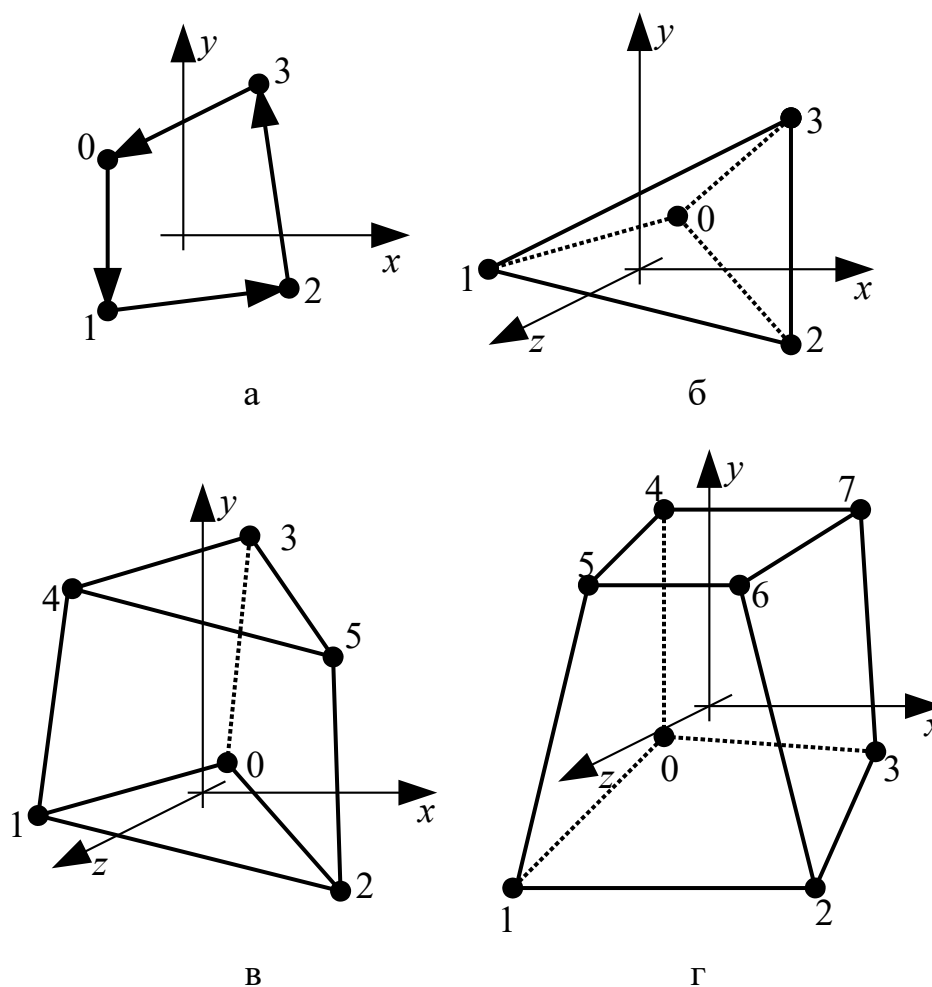


Рисунок 3.1 – Порядок обходу в елементах:

а – порядок обходу для двовимірних елементів

б – порядок обходу для тетраедрів

в – порядок обходу для трикутних призм

г – порядок обходу для шестигранників

Для забезпечення можливості циклічного обходу вузлів одного елемента припустимо, що операція доступу до коду вершини по її номеру визначена у вигляді

$$\mathbf{M}(i, j) = \mathbf{M} \cdot \mathbf{E}[i][j] = \begin{cases} j \bmod |e_j|, & j \geq 0 \\ 0, & j < 0 \wedge (j \bmod |e_j|) = 0, \\ |e_j| + (j \bmod |e_j|), & j < 0 \wedge (j \bmod |e_j|) \neq 0. \end{cases}$$

В результаті множина номерів (кодів) вузлів, сусідніх з j -ю вершиною i -го елемента, можна визначити у наступному вигляді:

$\text{neighbours}(\mathbf{M}, i, j) = \begin{cases} \{\mathbf{M}(i, j-1), \mathbf{M}(i, j+1)\}, & \text{для плоских елементів,} \\ \{\mathbf{M}(i, j-1), \mathbf{M}(i, j+1), \mathbf{M}(i, j+f)\}, & \text{для об'ємних елементів,} \end{cases}$

де f кількість вузлів у нижній грані ($f=3$ у тетраедрів і трикутних призм, $f=4$ у шестигранників).

Для формування множини номерів (кодів) вузлів, суміжних у вузлі з номером k_n , необхідно об'єднати множини номерів вузлів, сусідніх з цією вершиною всіх інцидентних у k_n елементів. Цю процедуру можна подати таким алгоритмом.

algorithm adjacent

input:

$\mathbf{M}$ – дискретна модель форми об'єкта

k_n – код вузла

output:

$\mathbf{A}$ – множина номерів (кодів) вузлів, суміжних у вузлі з номером k_n

begin

$\mathbf{A} \leftarrow \emptyset$

$\mathbf{I} \leftarrow \mathbf{M}.\text{incident}(k_n)$

for each $k_e \in \mathbf{I}$ **do**

$e \leftarrow \mathbf{M}(k_e)$

$j \leftarrow \overline{(j \in 0, \dots, |e|-1 : \mathbf{M}(k_e, j) = k_n)}$ // j – номер вершини у e з кодом k_n

$\mathbf{A} \leftarrow \mathbf{A} \cup \text{neighbors}(\mathbf{M}, k_e, j)$

end for

return $\mathbf{A}$

Алгоритм `adjacent` є лінійним і потребує тільки обробки множини інцидентних елементів, яка, зазвичай, є невеликою. Його обчислювальна складність для вузла з номером k_n є величиною порядку $O(|\mathbf{M}.\text{incident}(k_n)|)$ операцій. У подальшому викладі побудова множини кодів суміжних вузлів для вузла з кодом k_n позначена $\mathbf{M}.\text{adjacent}(k_n)$.

3.3 Структура даних

Зберігання дискретної моделі форми об'єкта (3.2) можливе декількома способами. Ключові відмінності між ними знаходяться у типах даних, які використовуються для зберігання вершин та елементів. Традиційно, коли мова йде про множини однотипних екземплярів, протиставляють дві структури даних: векторну і спискову.

Векторна структура даних базується на розміщенні екземплярів у послідовних адресах пам'яті комп'ютера, що дозволяє здійснювати доступ до елементу за його індексом за $O(1)$ часу. За такої умови видалення або вставка у середину нових екземплярів для такої структури даних пов'язані з лінійними витратами часу, викликаними зсувами масивів даних.

Спискова структура даних ґрунтується на включенні додатково до інформаційних полів адресних полів-показчиків, які служать для визначення зв'язків у списку. Вони дозволяють швидко видаляти та вставляти примірники у список, але призводять до збільшення витрат пам'яті комп'ютера. Також їх ітерування, зазвичай, є більш повільним за рахунок більшої фрагментації оперативної пам'яті.

Розглянемо дві можливі реалізації: 1 – що ґрунтується на використанні динамічних масивів; 2 – що ґрунтується на використанні двозв'язних списків для зберігання вузлів та елементів. Для дослідження продуктивності використаємо модельні задачі: 1 – додавання довільного набору восьмивершинних елементів, заданих координатами вершин; 2 – видалення довільного набору вузлів.

Як тестовий використовується комп'ютер з процесором Intel Core i5, 8 гігабайтами оперативної пам'яті, який знаходиться під управлінням операційної системи на базі Linux. Кожен тест запускається тричі, у результаті фіксується найменший час, витрачений на виконання завдання.

На рис. 3.2 показана залежність часу побудови дискретної моделі від кількості елементів. На осі абсцис основне значення показує кількість елементів у моделі, а значення у дужках – вузлів.

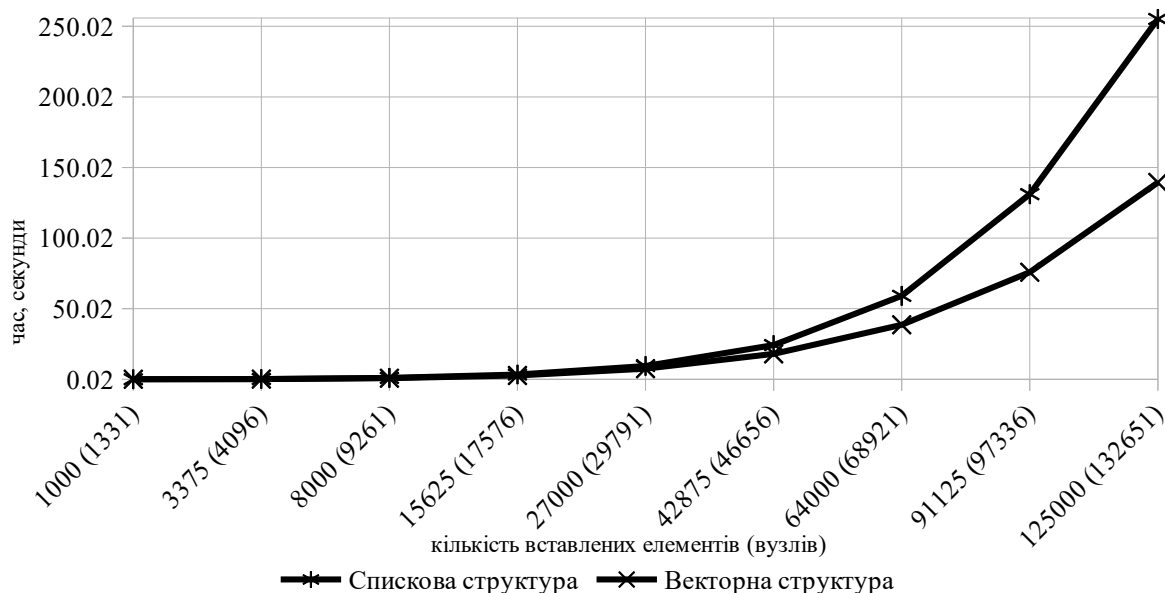


Рисунок 3.2 – Залежність часу побудови моделі від кількості елементів і вузлів

Побудова дискретної моделі, яка ґрунтується на динамічних масивах (векторна структура), потребує меншого часу відносно аналогічної, заснованої на списковій структурі. В обох випадках графіки мають вигляд степеневої функції. Вигляд графіка обумовлюється тим, що обчислювальну складність вставки m вершин з перевіркою

існування можна оцінити величиною $O\left(\frac{m(m-1)}{2}\right)$.

На рис. 3.3 показана залежність часу видалення тисячі вузлів від загальної їх кількості у дискретній моделі форми об'єкта. Необхідність лінійних зсувів обумовлює істотно більш повільне видалення (близько 8 разів) при використанні векторних структур для зберігання вузлів і елементів у порівнянні зі списковими структурами. В обох випадках залежність часу від кількості вузлів у моделі близька до лінійної.

Для дослідження впливу кількості видалених вузлів на час алгоритму роботи розглянемо дискретну модель (рис. 3.4), яка складається з 27000 вузлів і 24389 елементів. Як у попередньому випадку, видалення вузлів при використанні спискових структур для зберігання вузлів і елементів істотно швидше векторних.

Необхідно відзначити, що операції видалення вузлів і елементів при побудові дискретних моделей використовуються рідше за вставку та ітерування вузлів, в яких більш ефективними себе показали векторні структури даних.

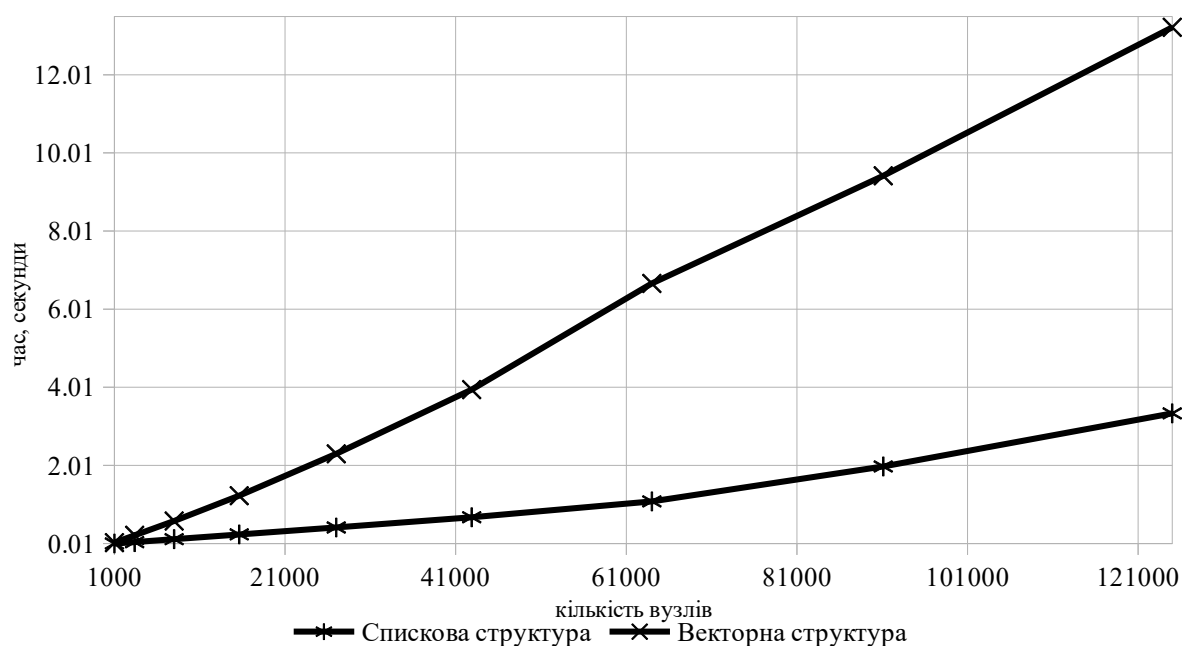


Рисунок 3.3 – Залежність часу видалення 1000 вузлів від кількості вузлів у моделі

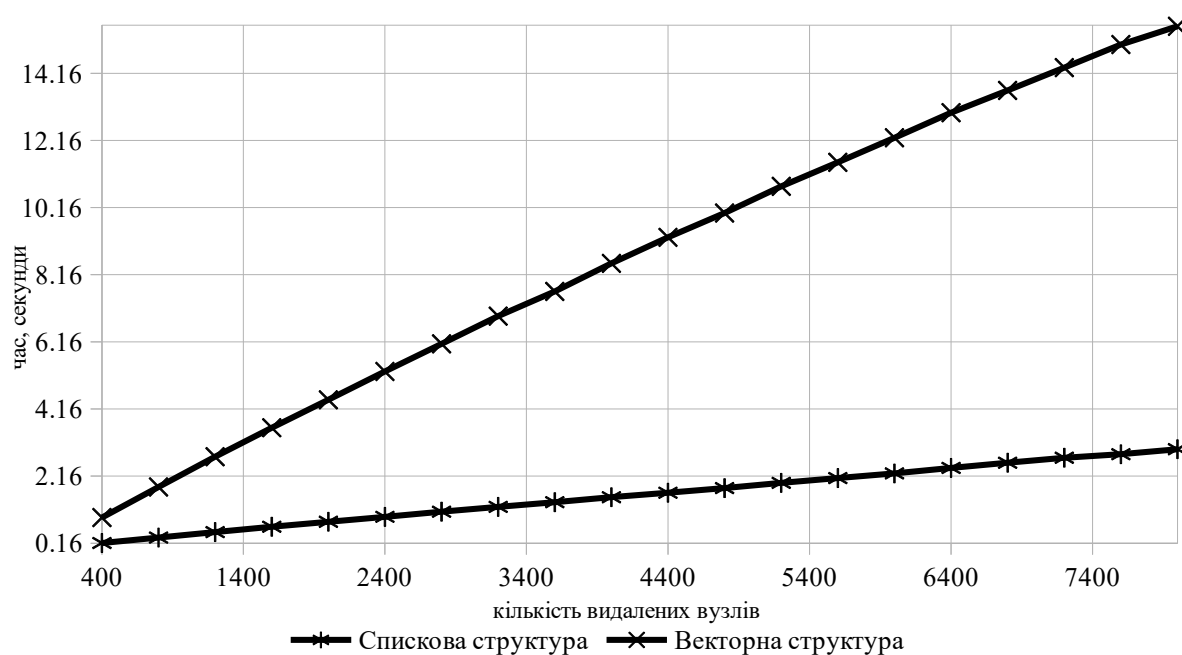


Рисунок 3.4 – Залежність часу від кількості видалених вузлів

Висновки до розділу 3

Отже, основним науковим результатом розділу є підхід до формального опису дискретних моделей форм об'єктів, який дозволяє використовувати єдину нотацію при викладі методів їх автоматичної генерації. Подання дискретної моделі

множинами вузлів і елементів водночас зі зберіганням для кожного вузла разом з його координатами списку інцидентних елементів дозволяє знизити порядок складності алгоритмів вставки та видалення вузлів.

Серія обчислювальних експериментів для порівняння двох реалізацій: на основі спискових та векторних структур даних показала, що перші (спискові) більш ефективні при видаленні вузлів, а другі – при безпечному додаванні (при вставці з перевіркою існування вузлів з однаковими координатами).

Більш висока ефективність спискових структур при видаленні обґрунтовується тим, що видалення одного екземпляра зі списку має обчислювальну складність порядку $O(1)$. Недоліком таких структур є відсутність можливості довільного доступу (загалом, доступ до екземпляра за індексом у списку потребує ітерування списку).

Векторні структури розміщують екземпляри у послідовних адресах пам'яті комп'ютера та дозволяють здійснювати довільний доступ до екземпляра. У результаті алгоритми лінійного пошуку, які використовуються при вставці для пошуку однакових вершин, працюють швидше (зокрема, за рахунок більш ефективного використання кешів процесора). Отже, операції вставки для векторних структур у 1,2–2 рази швидші відносно спискових. Одночасно, векторні структури практично на порядок повільніші спискових у видаленні вузлів і елементів моделі, що пояснюється необхідністю зсувів масивів даних та оновлення відповідних індексів. Проте, векторні структури дозволяють розробити ефективні паралельні алгоритми ітерування, у тому рахунку, під час оновлення індексів.

Результати третього розділу опубліковано у роботах [178, 182, 252].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилання [189, 207, 220, 224, 242, 245, 303, 315, 369, 445].

4 ДИСКРЕТНІ МОДЕЛІ МЕЖ ОБ'ЄКТІВ МАШИНОБУДУВАННЯ

Задачі побудови дискретних моделей меж двовимірних або тривимірних тіл, найчастіше, виникають під час розробки засобів їх візуалізації, тривимірного друку, а також чисельного аналізу на базі методу граничних елементів. Водночас основною вимогою, що висуваються до таких дискретних моделей (у додатках до чисельного аналізу), є точність апроксимації меж. Характерні точки та лінії (точки та лінії зламу поверхні) повинні бути подані вузлами та ребрами елементів дискретних моделей. Щільність елементів повинна бути адаптивною до кривини меж об'єкта.

Одним з перших та водночас найбільш ефективних (з точки зору швидкості роботи) алгоритмів побудови дискретних моделей меж форм об'єктів, поданих функціонально, є метод крокуючих кубиків (marching cubes) [345]. Цей метод заснований на ідеї сканування тривимірної області кубиками зі сталим або адаптивним кроком. У вершинах кожного кубика обчислюють значення функції, що подає форму об'єкта. Потім отримані значення використовують при визначенні відповідного шаблону для тріангуляції ділянки межі, пересіченої ребрами кубика. Всього таких шаблонів 256 (2^8), проте, унікальних (з точністю до обертання) з них тільки 16 (рис. 4.1, чорним кольором позначені внутрішні вершини кубиків). Для швидкого визначення ребер, які пересічені межею, та відповідного шаблону розроблені спеціальні таблиці. Кожному шаблону зіставлений двійковий код, що відповідає його позиції у цих таблицях. У оригінальному алгоритмі точки перетину ребер межами [345] запропоновано знаходити лінійною інтерполяцією.

Природним аналогом крокуючих кубів на площині є метод крокуючих квадратів (marching squares), який використовує квадрати для сканування площини [348]. У двовимірному випадку шаблонів перетину квадратів межею області буде 16 (2^4), з яких унікальними (з точністю до обертання) можна вважати 6 (рис. 4.2).

Програмні реалізації обох методів, зазвичай, засновані на присвоєнні кожному шаблону двійкового коду, який визначається для кожного осередку зміною біта, що відповідає номеру вузла у порядку обходу. Код визначає номери рядків у двох

таблицях: таблиці сторін (номерів пар вершин), пересічених границею, та таблиці межевих елементів (пар або трійок номерів сторін, на яких знайдені перетини з границею).

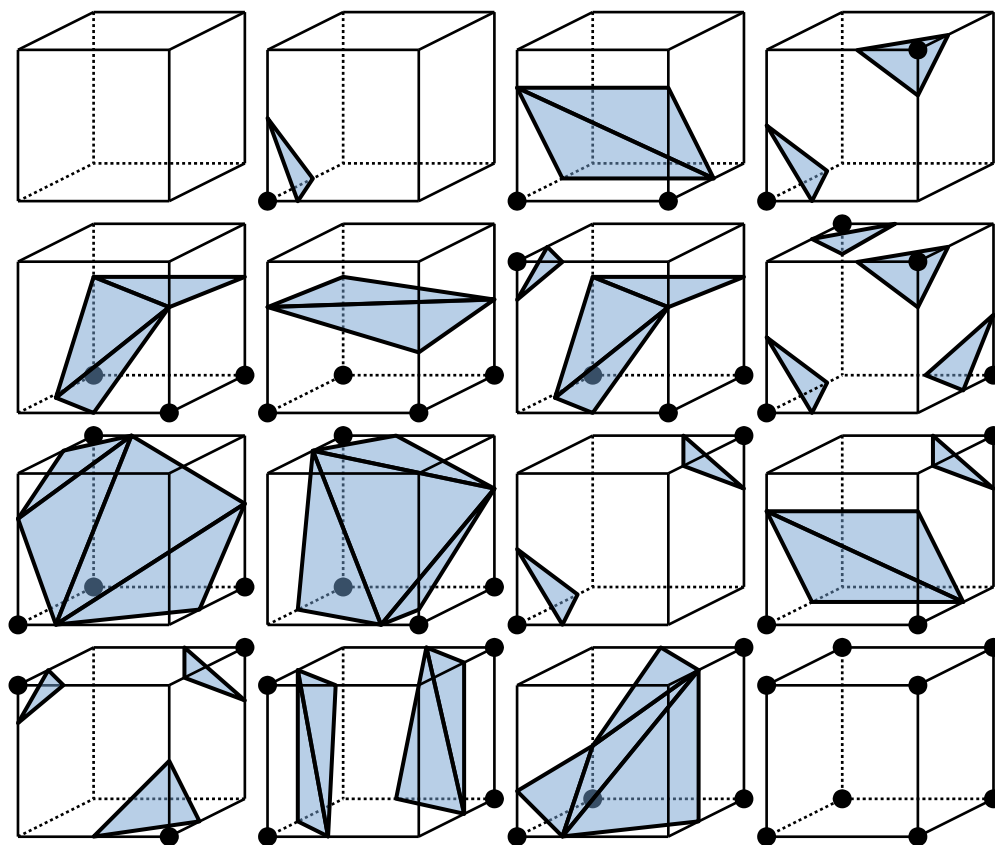


Рисунок 4.1 – Шаблони тріангуляції

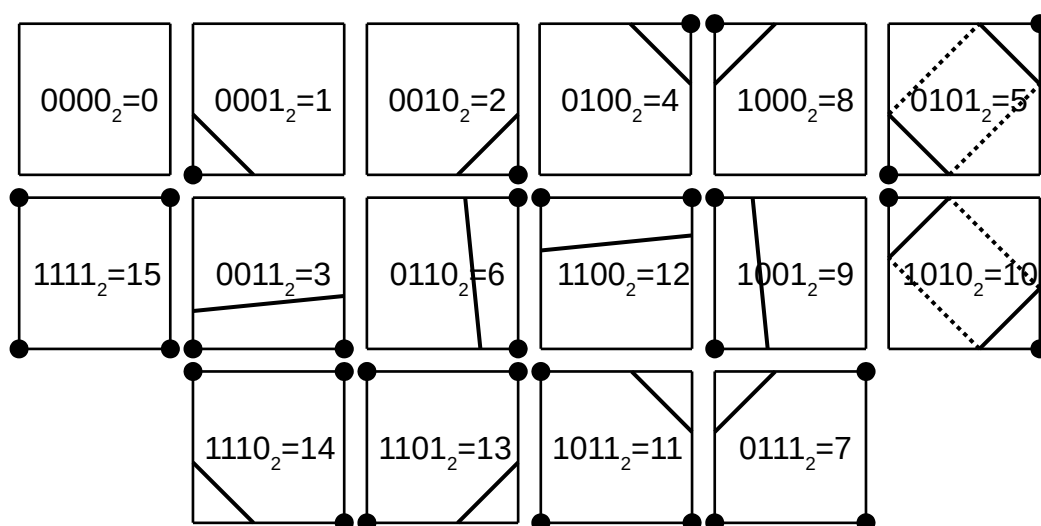


Рисунок 4.2 – Шаблони метода крокуючих квадратів

Основним недоліком цих методів є можливість отримання незв'язних дискретних моделей у результаті застосування неоднозначних шаблонів. Наприклад, двовимірних

шаблонів з кодами 5 та 10 (рис. 4.3) або тривимірних шаблонів, у яких є грані з двома внутрішніми вершинами на одній діагоналі та двома зовнішніми на іншій.

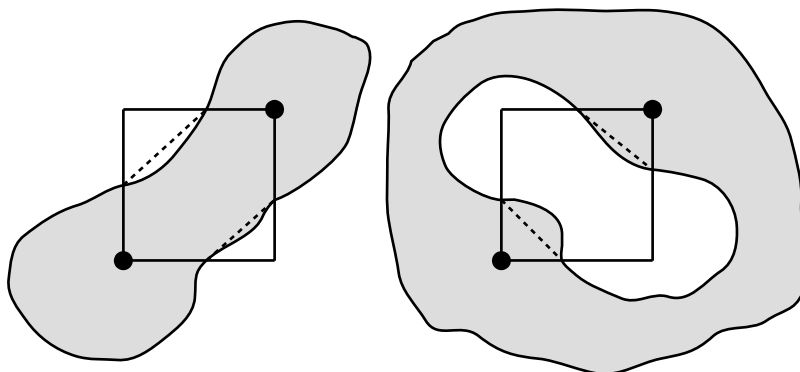


Рисунок 4.3 – Неоднозначність двовимірного шаблону з кодом 5

Найпростішим способом вирішення проблеми незв'язності дискретних моделей, отриманих методом крокуючих квадратів, є рекурсивне розбиття неоднозначних осередків на чотири рівні частини. Водночас для отримання топологічно коректних моделей у тривимірному випадку необхідно виконати додатковий аналіз. У роботі [375] запропоновано використовувати білінійну інтерполяцію для усунення неоднозначності випадків, подібних двовимірним з кодами 5 та 10, для граней кубів. В роботі Черняєва Є. В. [233] підхід, який ґрунтується на білінійній інтерполяції, узагальнено та виділено 33 основних шаблони, а також критерії їх використання (даний метод відомий під назвою «Marching Cubes 33»). Особливості програмної реалізації методу Marching Cubes 33 розглянуті у [259, 397].

Побудова дискретних моделей меж тривимірних форм об'єктів з використанням елементів, форма яких відмінна від трикутників (наприклад, чотирикутників), пов'язана з використанням більш складних алгоритмів. Використання багатьох бібліотек підпрограм для побудови дискретних моделей форм об'єктів виконується за принципом чорного ящика. У той же час, розробка складних програмних продуктів вимагає забезпечення максимальної універсальності рішень, у тому числі, за рахунок залучення патернів (шаблонів) проектування [190, 214, 251, 431]. Можливим способом підвищення технологічності та універсальності

програмного коду САПР машинобудування є розробка узагальнених методів побудови дискретних моделей форм об'єктів.

4.1 Проекційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально

Нехай форма деякого об'єкта Ω подана функцією $F(P)$, яка визначена у кожній точці P евклідового простору. Причому $F(P) > 0$, якщо $P \in \Omega \setminus \delta(\Omega)$, $F(P) = 0$, якщо $P \in \delta(\Omega)$ і, відповідно, $F(P) < 0$, якщо $P \notin \Omega$. Задача полягає у розробці метода побудови дискретної моделі межі об'єкта Ω на базі довільної форми скінченного елемента.

Для розв'язання поставленої задачі методи суперпозиції, які було запропоновано у [428–430] для побудови сіток чотирикутників (у двовимірному випадку) та шестигранників (у тривимірному випадку), можна узагальнити на випадок довільної розмірності та форми елемента. У основі цих методів лежить ідея використання деякої базової стратегії адаптації початкової сітки до меж форми деякого об'єкта.

4.1.1 Загальний алгоритм метода

Припустимо, що для частини простору, яка повністю включає об'єкт Ω , задана деяка фонові сітка. Якщо видалити з фонові сітки комірки, зовнішні відносно Ω або які перетинаються його межею, отримаємо початкову сітку – множини елементів, яку можна вважати образом цього об'єкта. Потім для кожної межевої грані початкової сітки можна знайти відповідну проекцію на межі об'єкта. В результаті отримаємо такий універсальний алгоритм:

algorithm background_grid_surface

input:

F – функція, яка подає форму об’єкта Ω

G – фонові сітка

ε – чисельний нуль

output:

M – дискретна модель $\delta(\Omega)$

begin

for each $n \in G.N$ **do**

if $F(n.v) < 0$ **then**

 remove_node($G, G.code(n)$)

end if

end for

$V_0 \leftarrow$ вузли дотику елементів G

while $V_0 \neq \emptyset$ **do**

$T \leftarrow \bigcup_{v \in V_0} G.incident(v)$

$k_{e0} \leftarrow \operatorname{argmin}_{k_e \in T} (G.power(G(k_e)))$

 remove_element(G, k_{e0})

$V_0 \leftarrow$ вузли дотику елементів G

end while

$M \leftarrow \delta(G)$

$h \leftarrow \min_{(n_0 \neq n_1) \in M.N} \|n_0.v - n_1.v\|$

for each $n \in M.N$ **do**

$n.v \leftarrow \text{find\_border}(n.v, F, h, \varepsilon)$

end for

return M

Отже, у алгоритмі можна виділити три основні етапи. Першим етапом є видалення всіх зовнішніх вузлів та інцидентних їм елементів фонові сітки. В

результаті залишаються вузли та елементи, які можна вважати початковим дискретним образом форми об'єкта Ω .

На другому етапі усуваються конфігурації, які можуть спричинити топологічно невірні результати. Такі конфігурації утворюються у разі дотику елементів у вузлі (приклад такої конфігурації для двовимірного випадку зображений на рис. 4.4). В даному випадку елементи вважаються сусідніми у вузлі, якщо у його оточенні простір ділиться на дві зв'язних підмножини, інакше вони вважаються дотичними.

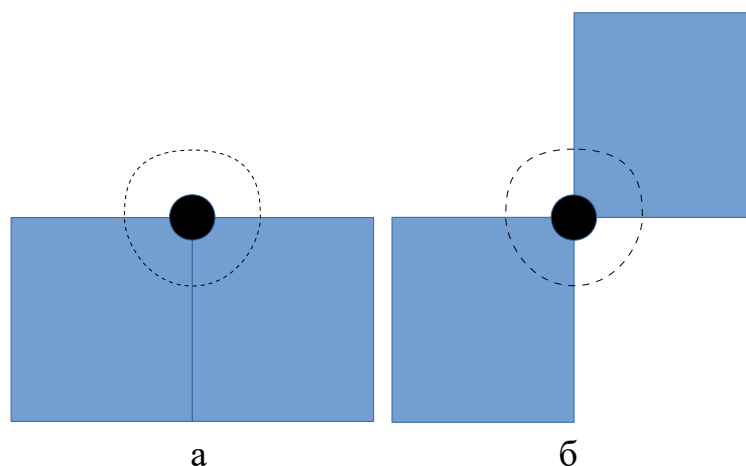


Рисунок 4.4 – Взаємне розташування елементів:

а – сусідні елементи

б – дотичні елементи

На третьому етапі формується підсумкова дискретна модель. Спочатку формується множина межових граней для елементів, які залишилися після другого етапу (ця дія позначена оператором δ). Потім для кожного межового вузла знаходиться проекція на межі об'єкта Ω .

4.1.2 Алгоритм пошуку проекцій вузлів на межу об'єкта

Для проектування вузла на межу використовується алгоритм `find_border( $v$ ,  $F$ ,  $h$ ,  $\epsilon$ )`. Він знаходить проекції точки v на границю функції F з кроком

h у напрямку градієнта функції F (ε – чисельний нуль). Крок h визначається в алгоритмі `background_grid_surface` як мінімальна відстань між парою сусідніх вузлів. Формально алгоритм `find_border` можна записати у такому вигляді:

algorithm `find_border`

input:

v – координати точки, для якої необхідно знайти проекцію на межу об'єкта Ω

F – функція, що подає форму об'єкта Ω

h – крок метода

ε – чисельний нуль

output:

точка на межі Ω

begin

$f \leftarrow F(v)$

while $|f| \geq \varepsilon$ **do**

$p \leftarrow (v - \text{sign}(f) \cdot h \cdot \hat{\text{grad}} F(v))$

if $\text{sign}(f) \neq \text{sign}(F(p))$ **then**

return `binary`(v, p, F, ε)

end if

$v \leftarrow p$

$f \leftarrow F(v)$

end while

return v

Алгоритм `find_border` здійснює пошук відрізка, на якому функція F змінює знак. Пошук у цьому алгоритмі виконується у напрямку антиградієнта для внутрішніх точок та градієнта для зовнішніх. Потім за допомогою двійкового пошуку (алгоритм `binary`) знаходиться точка на межі Ω .

Алгоритм двійкового пошуку `binary` знаходить корінь функції F на відрізку $[v_1; v_2]$. Його можна записати у такому вигляді:

algorithm binary**input:** v_1 – координати точки 1 v_2 – координати точки 2 F – функція, що змінює знак між точками v_1 і v_2 ε – чисельний нуль**output:** b – точка з відрізка $[v_1; v_2]$, яка є коренем F **begin** $f_1 \leftarrow F(v_1)$ $f_2 \leftarrow F(v_2)$ **do** $b \leftarrow 0,5 (v_1 + v_2)$ $f_b \leftarrow F(b)$ **if** $\text{sign}(f_1) \neq \text{sign}(f_b)$ **then** $v_2 \leftarrow b$ $f_2 = f_b$ **else** $v_1 \leftarrow b$ $f_1 \leftarrow f_b$ **end if****while** $|f_b| \geq \varepsilon$ **return** b

В результаті застосування алгоритму `background_grid_surface` буде отримана дискретна модель межі об'єкта. Структура граней дискретної моделі визначається початковою сіткою та функцією F . Отримані дискретні моделі меж об'єктів, зазвичай, вимагають додаткового згладжування та оптимізації положень вузлів для отримання найкращого результату.

4.1.3 Приклади застосування

Наприклад, розглянемо форму об'єкта, задану функцією

$$w(x, y, z) = \text{cuboid}(x, y, z, 2, 2, 2) \wedge \text{sphere}(x, y, z, 1) \wedge \neg \text{circle}(x; y; 0,5) \wedge \neg \text{circle}(x; z; 0,5) \wedge \neg \text{circle}(y; z; 0,5), \quad (4.1)$$

де $\text{cuboid}(x, y, z, w, h, d)$ – функція (2.24);

$\text{sphere}(x, y, z, r)$ – функція (2.20);

$\text{circle}(x, y, r)$ – функція (2.7).

В результаті використання рівномірної фонові сітки, що складається з 9261 вузла та 48000 тетраедрів, яка побудована на кубі $[-1; 1] \times [-1; 1] \times [-1; 1]$ (рис. 4.5, а), після видалення зовнішніх вузлів залишиться 2158 межових вузлів і 4332 елемента (рис. 4.5, б). Потім після знаходження відповідних проекцій на межу буде отримана дискретна модель межі у вигляді рис. 4.5, в.

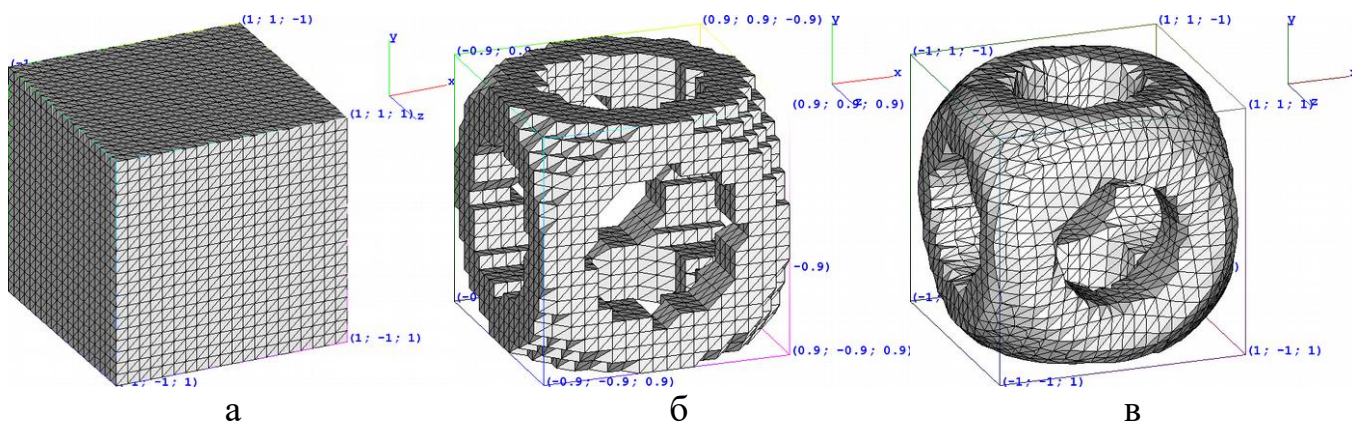


Рисунок 4.5 – Приклад побудови дискретної моделі межі з використанням фонові сітки тетраедрів для об'єкта, заданого формулою (4.1):

а – фонові сітка тетраедрів

б – фонові сітка тетраедрів після видалення зовнішніх вузлів

в – дискретна модель на базі трикутників на межі об'єкта

Для отримання дискретної моделі межі об'єкта, заданого формулою (4.1), також можна використовувати фонову сітку, яка, наприклад, складається з 9261 вузла і 8000 кубічних елементів (рис. 4.6, а). Після видалення зовнішніх вузлів з дискретної моделі залишиться 3064 граничних вузла і 3072 елементи (рис. 4.6, б), проекція яких на межу дозволить отримати результат (рис. 4.6, в).

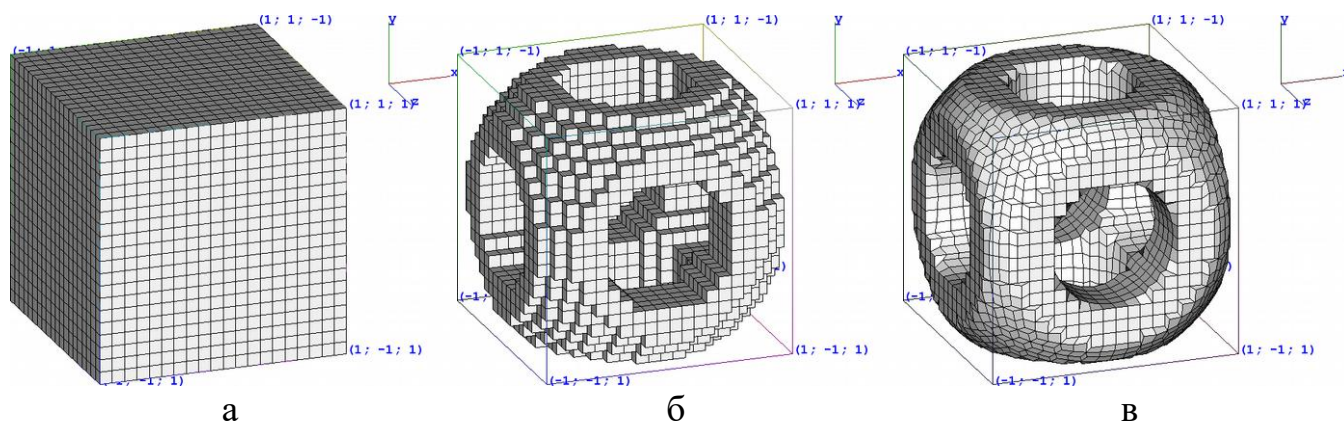


Рисунок 4.6 – Приклад побудови дискретної моделі межі з використанням фонові сітки кубів для геометричного об'єкта, заданого формулою (4.1):

а – фонові сітка кубів

б – фонові сітка кубів після видалення зовнішніх вузлів

в – дискретна модель на базі чотирикутників на межі об'єкта

Як фонові також можна використовувати неструктуровані дискретні моделі форм об'єктів. Так, наприклад, для моделі пробки, поданої формулою (2.29), фонові може виступати дискретна модель кругового циліндра з радіусом основи, що дорівнює 25, і висотою 50 (рис. 4.7, а для тетраедрів і рис. 4.7, б для шестигранників). В результаті застосування алгоритму `background_grid_surface` для фонових моделей циліндричної області будуть отримані дискретні моделі межі пробки (рис. 4.7).

Отже, густина вузлів та елементів у підсумкових дискретних моделях визначається наявністю відповідних областей у фонових сітках. Тобто, при використанні рівномірних фонових сіток будуть отримані близькі до рівномірних моделі меж, а для адаптивних фонових дискретних моделей буде більш помітною зміна щільності вузлів по поверхні.

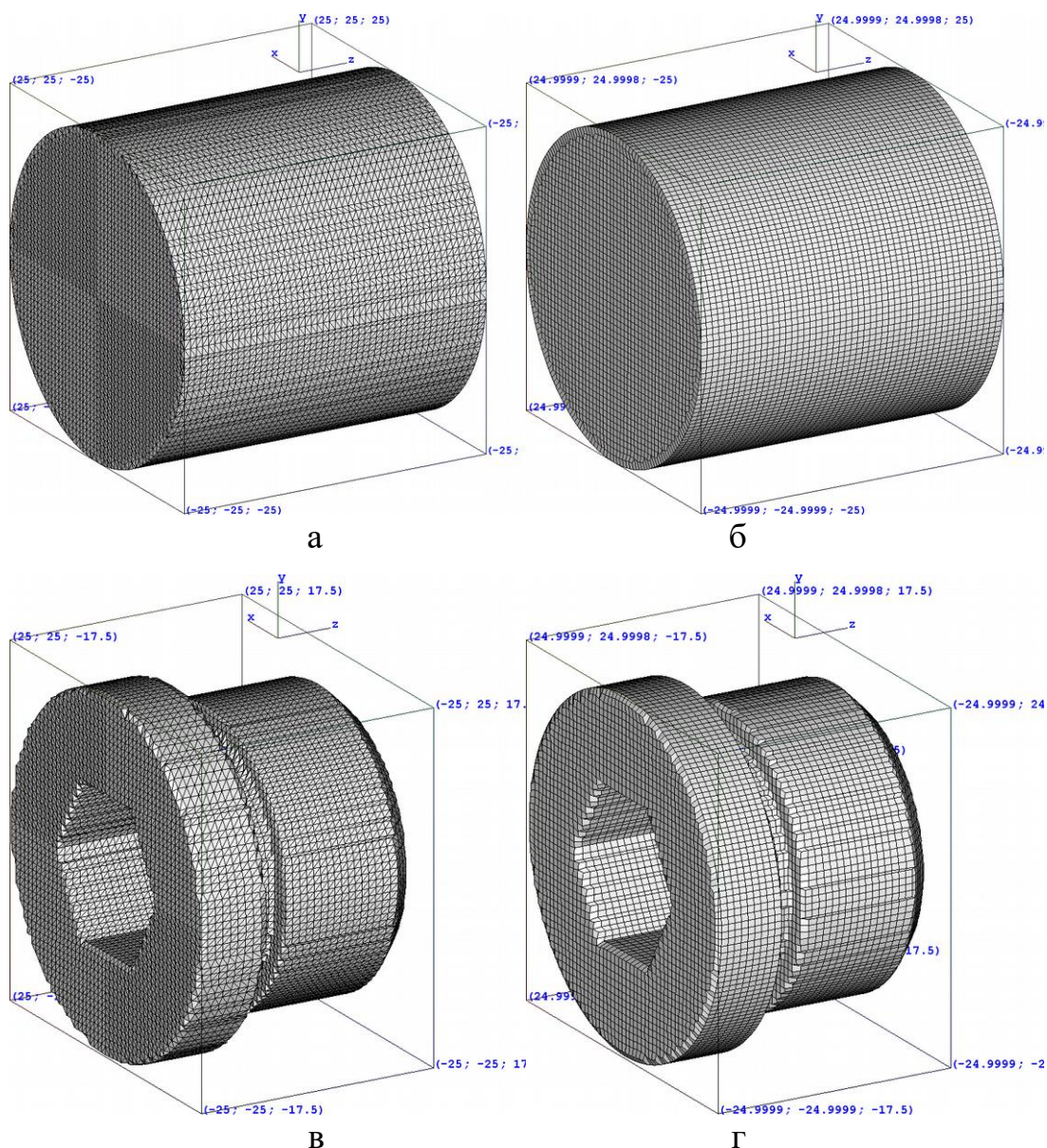


Рисунок 4.7 – Дискретні моделі межі пробки:

- а – фонова сітка тетраедрів (103275 вузлів і 586200 елементів)
- б – фонова сітка шестигранників (108783 вузла і 101800 елементів)
- в – дискретна модель на базі трикутників (11705 вузлів і 23406 елементів)
- г – дискретна модель на базі чотирикутників (13070 вузлів і 13068 елементів)

Наприклад, для побудови дискретної моделі межі важеля передньої підвіски, заданого формулою (2.28), як фонову можна використовувати модель профілю, адаптивну до його меж, зі збільшенням кількості елементів біля отворів. В результаті застосування фонової дискретної моделі, яка складається з 39675 вузлів і 197424 тетраедрів, можна отримати 7983 вузла і 15974 трикутника на межі (рис. 4.18).

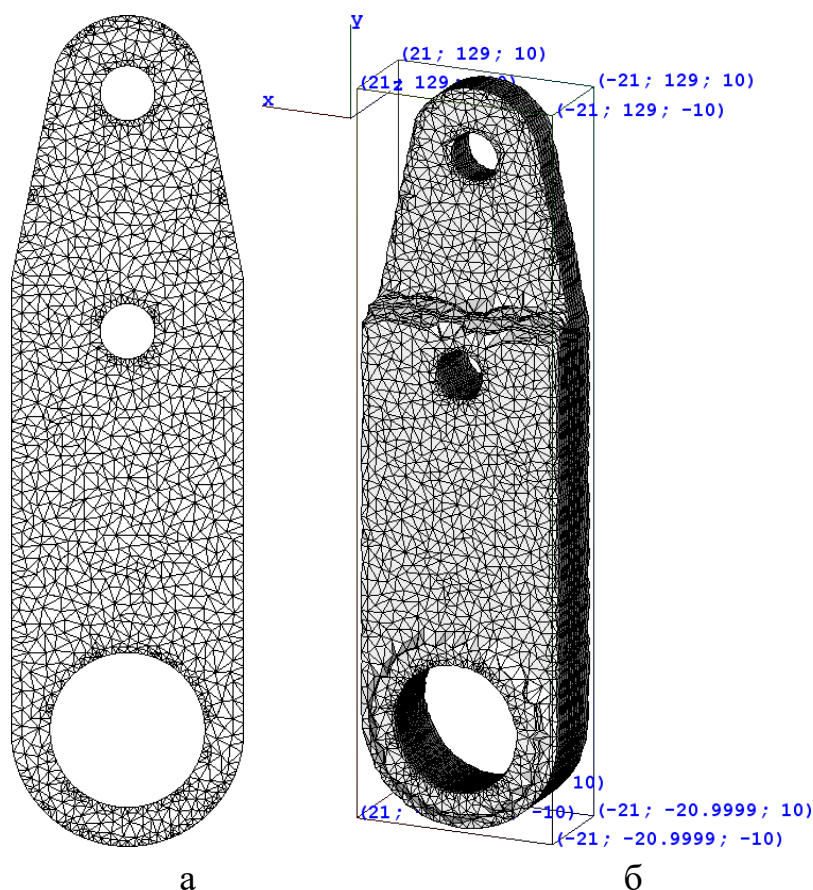


Рисунок 4.8 – Приклад використання адаптивної фонові дискретної моделі:

а – фонові сітка тетраедрів;

б – трикутники на межі моделі подовжувача важеля передньої підвіски.

4.1.4 Двовимірний випадок

Алгоритм `background_grid_surface`, у окремому випадку, може також застосовуватися для побудови контурів двовимірних областей. У цьому випадку на місці граней для пошуку проекцій виступатимуть сторони двовимірних елементів, а результатом завжди буде дискретна модель, яка подає контур відрізками прямої. Водночас описане вище зауваження про адаптивність залишається справедливим.

Наприклад, при використанні фонові сітки, яка складається з 676 вузлів і 1250 трикутників (рис. 4.9, а), для моделі пластини, заданої формулою (2.14), буде отримана модель межі, яка складається з 123 вузлів і 123 відрізків (рис. 4.9, б).

Водночас при використанні фонові сітки, яка складається з 676 вузлів і 625 чотирикутників (рис. 4.9, в), буде отримана модель, яка складається з 132 вузлів і 132 відрізків (рис. 4.9, г).

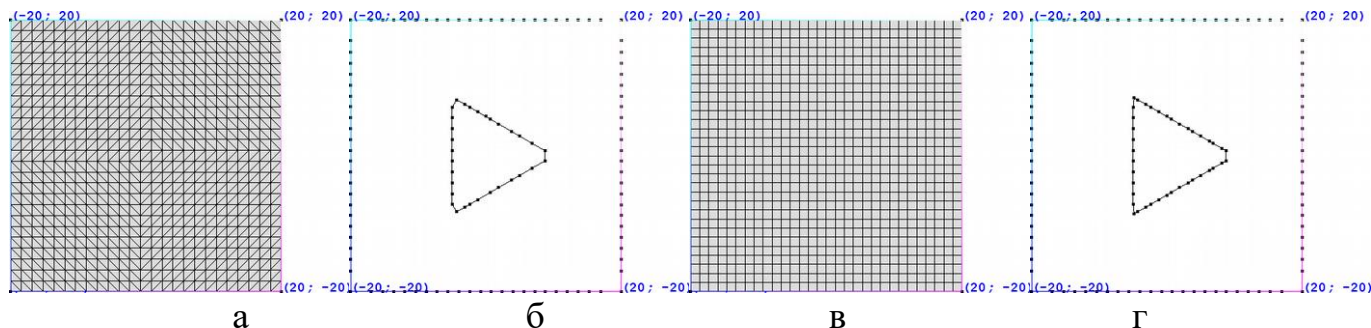


Рисунок 4.9 – Приклади побудови моделей меж двовимірної пластини:

а – фонові сітка трикутників

б – модель межі для фонові сітки трикутників

в – фонові сітка чотирикутників

г – модель межі для фонові сітки чотирикутників

За аналогією з тривимірним випадком у двовимірному випадку необхідно використовувати нерівномірні моделі на базі плоских елементів для забезпечення адаптивності підсумкових моделей меж.

4.1.5 Аналіз результатів

На відміну від методів побудови дискретних моделей меж об'єктів, які ґрунтуються на скануванні рівномірними комірками [397, 259, 345, 375], розроблений метод теоретично дозволяє генерувати моделі на базі комірок довільної форми.

Розроблений метод і пов'язані з ним алгоритми не залежать від форми елементів. Вони можуть застосовуватися для генерації дискретних моделей на базі

елементів різної форми. Єдина послідовність дій дозволяє проектувати програмне забезпечення на базі поведінкового шаблону проектування «Стратегія» [251].

Також запропонований метод можна використовувати для побудови картин ліній або поверхонь рівня для неявно заданих геометричних областей. Для цього у розроблених алгоритмах замість значення функції області необхідно аналізувати різницю її значення і константи, що відповідає необхідній лінії або поверхні рівня). Наприклад, для моделі пластини, заданої формулою (2.14), отримані лінії рівня, наведені на рис. 4.10.

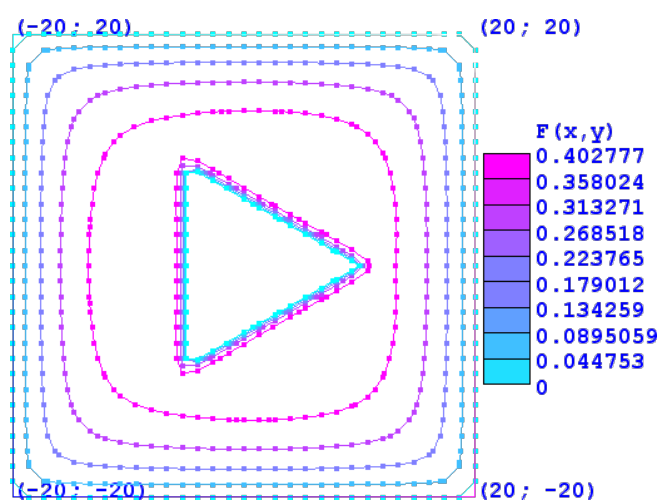


Рисунок 4.10 – Приклад ліній рівня для моделі двовимірної пластини

Аналіз отриманих дискретних моделей дозволяє зробити висновок про те, що застосування запропонованого методу може спричинити втрати точності апроксимації межі в околі її геометричних особливостей. Отже, до отриманих дискретних моделей необхідно застосовувати додаткові процедури їх оптимізації.

4.2 Метод згладжування координат вузлів дискретних моделей меж об'єктів, форма яких задана функціонально

Використання методів, що існують, [233, 345, 375] та описаного вище підходу до генерації дискретних моделей меж неявних функцій може спричинити неточності

апроксимації межі в околах її геометричних особливостей (наприклад, зламів поверхні). Збільшення числа вузлів та елементів у таких областях дозволяє досягти прийнятних результатів з точки зору візуалізації, але зумовлює значне зростання кількості невідомих, якщо моделі будуть використані у чисельному аналізі. В результаті актуальною є розробка методів підвищення точності апроксимації в околах зламів для дискретних моделей меж об'єктів, які не вимагають збільшення числа вузлів.

Аналіз результатів досліджень, присвячених оптимізації межових вузлів та елементів (пункт 1.4.2), дозволяє зробити висновок, що розроблені методи вирішують задачу точності апроксимації меж в околицях зламів для дискретних моделей, заснованих переважно на трикутниках. Підходи та методи, запропоновані у працях [188, 223, 234–236, 253, 263, 293, 297, 298, 302, 379–380], потребують додавання нових вузлів у дискретну модель, що спричиняє практичну неможливість їх застосування для чотирикутників. Використання ієрархічних структур за схемою, яка запропонована у роботах [218, 313, 358, 360, 436, 444, 487], дозволяє отримати прийнятну якість. Проте, водночас буде істотно зростати кількість вузлів та елементів, що буде спричиняти збільшення розмірності задачі (наприклад, у додатках до скінченно-елементного аналізу). Сегментація та кластеризація [300, 366] дозволяють виявити зони, у яких є геометричні особливості, але також вимагають вставки додаткових вузлів. Отже, у опублікованих на даний момент роботах не виявлено ефективні методи оптимізації дискретних моделей, які можна застосовувати одночасно і до трикутників, і до чотирикутників.

4.2.1 Функціонал відстані-довжини

Одним з найбільш вживаних та водночас простих методів покращення дискретних моделей є локальне згладжування Лапласа [266]. Воно пересуває кожен вузол у геометричний центр багатокутника, утвореного суміжними вузлами (рис. 4.11). Традиційно для досягнення кращого результату це згладжування застосовують кілька разів.

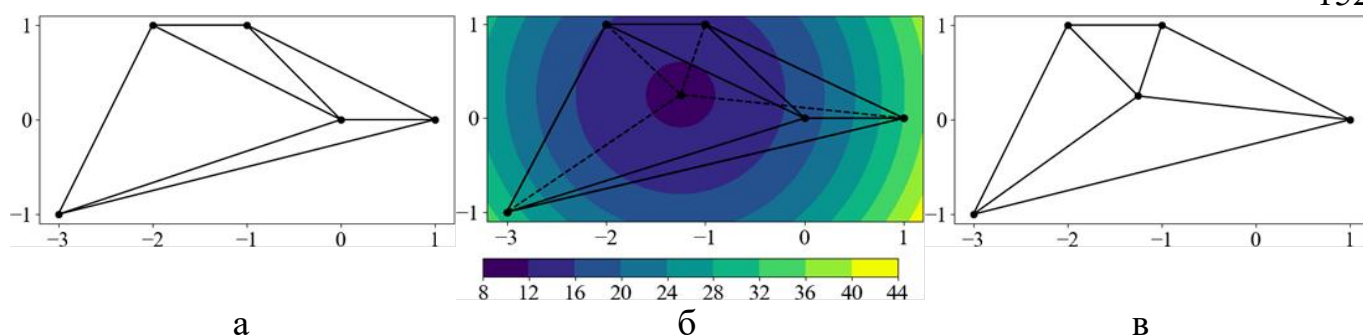


Рисунок 4.11 – Згладжування Лапласа:

а – вихідна сітка

б – розподіл значень функції (4.2)

в – результат згладжування

У згладжуванні Лапласа кожне ребро, яке сполучене з центральним вузлом, можна розглядати як пружину з початковою нульовою довжиною [298]. Сума сил, що діють у i -му вузлі сітки $\mathbf{M}$, визначається формулою:

$$F(\mathbf{M}, i, \kappa) = \kappa \sum_{k \in \mathbf{A}_i} (\mathbf{M}[k] - \mathbf{M}[i]),$$

де κ – жорсткість пружини;

$\mathbf{A}_i = \mathbf{M}.\text{adjacent}(i)$ – множина кодів суміжних з i -м вузлів.

Точка рівноваги системи пружин відповідає мінімуму потенційної енергії, яка визначається формулою:

$$E(\mathbf{M}, i, \kappa) = \frac{\kappa}{2} \sum_{k \in \mathbf{A}_i} \|\mathbf{M}[k] - \mathbf{M}[i]\|^2. \quad (4.2)$$

Формулу (4.2) у тривимірному випадку можна записана у такому вигляді:

$$E(\mathbf{M}, i, \kappa) = \frac{\kappa}{2} \sum_{k \in \mathbf{A}_i} \left((x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2 \right), \quad (4.3)$$

де x_i , y_i і z_i – відповідно, абсциса, ордината та апліката i -го вузла;

x_k, y_k і z_k – відповідно, абсциса, ордината та апліката k -го вузла.

Диференціюючи функцію (4.3) за x_i, y_i, z_i та порівнюючи до нуля отримаємо

$$\begin{aligned}x_i &= \frac{1}{|\mathbf{A}_i|} \sum_{k \in \mathbf{A}} x_k, \\y_i &= \frac{1}{|\mathbf{A}_i|} \sum_{k \in \mathbf{A}} y_k, \\z_i &= \frac{1}{|\mathbf{A}_i|} \sum_{k \in \mathbf{A}} z_k.\end{aligned}$$

де $|\mathbf{A}_i|$ – кількість суміжних з i -м вузлів.

В результаті виходить проста та дуже ефективна (з обчислювальної точки зору) процедура. Її застосування для дискретних моделей меж об'єктів пов'язане з пошуком проекції нової позиції точки на межу. Багаторазове ітераційне застосування такого згладжування прагне забезпечити рівну довжину сторонам елементів. Проте, відсутність інформації про точність апроксимації, зазвичай, зумовлює збільшення похибки в околах особливостей поверхні.

Наприклад, для дискретної моделі кулі одиничного радіуса з осьовим круговим отвором (радіус отвору дорівнює половині радіуса кулі) для вихідної дискретної моделі на базі трикутників (рис. 4.12, а) згладжена модель матиме вигляд, наведений на рис. 4.12, б. Аналогічно погіршується точність апроксимації межі для чотирикутників (рис. 4.12, в, г).

Розглянемо енергетичну функцію вигляду

$$E(\mathbf{M}, i, \kappa, F) = \kappa \sum_{k \in \mathbf{A}_i} \|\mathbf{M}[k] - \mathbf{M}[i]\|^2 + (1 - \kappa) \sum_{q \in \mathbf{I}_i} \|C_q - \text{projection}(C_q, F)\|^2, \quad (4.4)$$

де κ – параметр методу;

F – неявна функція, що описує форму об'єкта;

$\mathbf{A}_i$ – множина кодів суміжних з i -м вузлів;

$\mathbf{I}_i = \mathbf{M}.\text{incident}(i)$ – множина кодів інцидентних до i -го вузла елементів;

$$C_q = \frac{1}{|\mathbf{M}(q)|} \sum_{j \in \mathbf{M}(q)} \mathbf{M}[j] - \text{центр мас елемента з кодом } q;$$

$\text{projection}(C_q, F)$ – проекція точки C_q на поверхню форми об'єкта, яка задана функцією F .

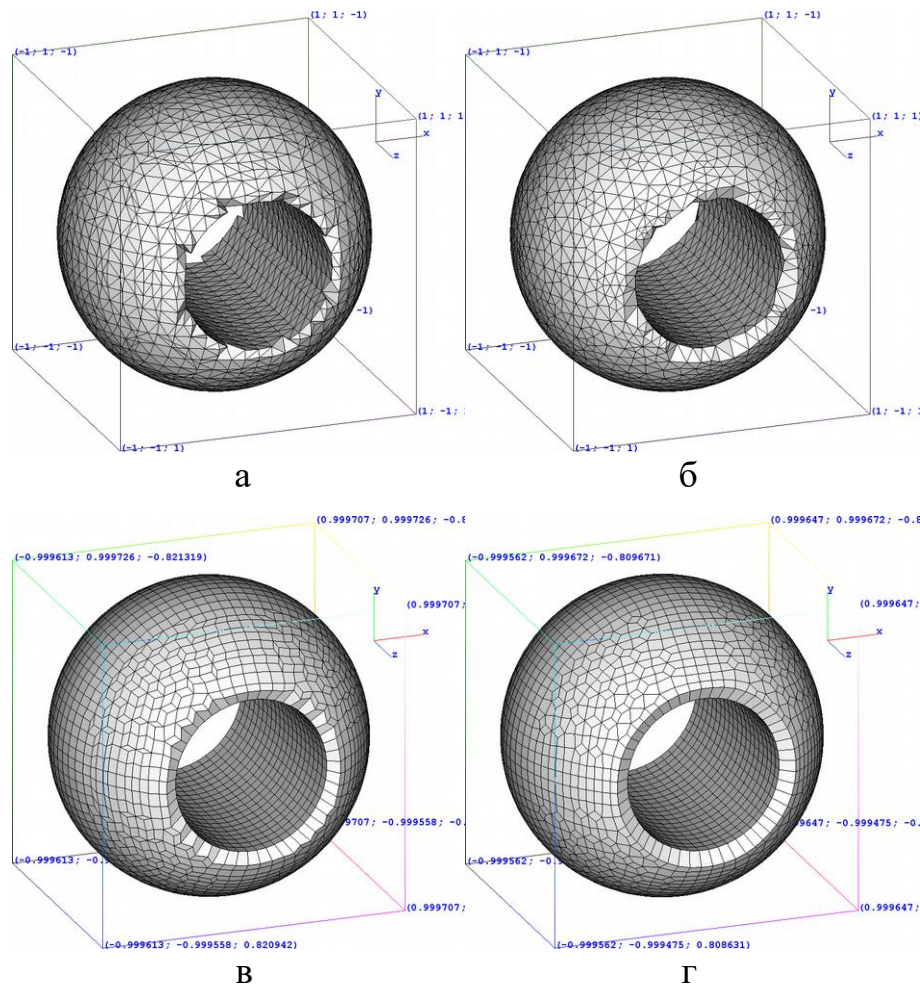


Рисунок 4.12 – Приклади застосування згладжування Лапласа:

а – вихідна дискретна модель трикутників

б – результат згладжування дискретної моделі трикутників

в – вихідна дискретна модель чотирикутників

г – результат згладжування дискретної моделі чотирикутників

Перший доданок у формулі (4.4) гарантує існування мінімуму [298]. Вибір другого доданка виконаний з наступного припущення. Кожен плоский елемент дискретної моделі межі об'єкта можна вважати апроксимацією відповідної частини поверхні (рис. 4.13), а відстань від центру елемента до поверхні – показником точності такої апроксимації.

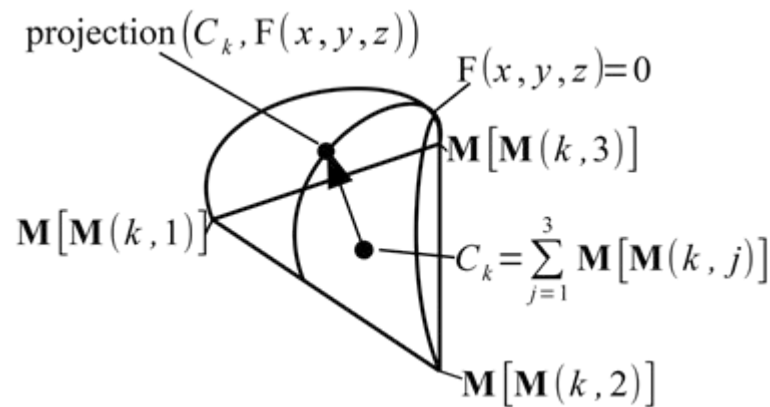


Рисунок 4.13 – Апроксимація частини поверхні трикутником

При обчисленні значення формули (4.4) пошук проекції точки на поверхню можна здійснювати у напрямку градієнта (для зовнішніх точок) або антиградієнта (для внутрішніх) за допомогою алгоритму `find_border`. Використання довжин сторін елементів та відстаней від центрів елементів до поверхні обумовлює назву «функціонал відстані-довжини».

4.2.2 Локальний алгоритм пошуку мінімуму функціоналу відстані-довжини

Знаходження аналітичного виразу для точки мінімуму функціоналу (4.4), подібно до того, як це зроблено для формули (4.3) не видається можливим через неявність і, зазвичай, нелінійність функції $F(x, y, z)$. Для чисельного пошуку мінімуму можна скористатися модифікацією методу Гауса, яка формалізована у вигляді такого алгоритму:

algorithm `local_refinement`

input:

F – функція, яка подає форму об'єкта

M – дискретна модель межі об'єкта

k_n – код вузла для оптимізації

κ – параметр жорсткості системи

ε – чисельний нуль

output:

$\mathbf{M}$ – дискретна модель межі об’єкта

begin

$\mathbf{A}_k \leftarrow \mathbf{M}.\text{adjacent}(k_n)$

$P_0 \leftarrow \mathbf{M}[k_n]$

$f_0 \leftarrow E(\mathbf{M}, k_n, \kappa, F)$

for each $j \in \mathbf{A}_k$ **do**

$l \leftarrow 0,1$

do

$P_1 \leftarrow (1 - l) \mathbf{M}[k_n] + l \mathbf{M}[j]$

$h \leftarrow \|P_0 - P_1\|$

$\mathbf{M}[k_n] \leftarrow \text{find\_border}(P_1, F, h, \varepsilon)$

$f_1 \leftarrow E(\mathbf{M}, k_n, \kappa, F)$

if $f_0 \leq f_1$ **then**

$\mathbf{M}[k_n] \leftarrow P_0$

$l \leftarrow \frac{l}{2}$

else

$f_0 \leftarrow f_1$

end if

while $l \geq \varepsilon$

end for

end

Пошук здійснюється у l -координатах багатокутника, який утворений сусідніми з i -м вузлами. Спуск здійснюється по кожній з l -координат, поки цільова функція спадає. Використовується адаптивний крок: у кожному напрямку, починаючи з кроку, значення якого емпірично прийнято рівним 0,1, якщо цільова функція перестає спадати, тоді крок зменшується вдвічі.

4.2.3 Глобальний алгоритм пошуку мінімуму функціоналу відстані-довжини

Застосування локального алгоритму послідовно до кожного вузла дискретної моделі межі об'єкта можна розглядати як ітераційне наближення до мінімуму функціоналу (4.4) у глобальній постановці. Відповідно, необхідно послідовно застосовувати локальний алгоритм для кожного вузла дискретної моделі межі об'єкта. В результаті отримаємо такий алгоритм оптимізації положень вузлів дискретної моделі межі об'єкта, форма якого задана функціонально:

algorithm boundary_refinement

input:

F – функція, яка подає форму об'єкта

M – дискретна модель межі об'єкта

κ – параметр жорсткості системи

ε – чисельний нуль

output:

M – дискретна модель межі об'єкта

begin

for each $n \in M.N$ **do**

 local_refinement($F, M, M.code(n), \kappa, \varepsilon$)

end for

end

За аналогією зі згладжуванням Лапласа алгоритм boundary_refinement необхідно застосовувати кілька разів. Водночас на кожній ітерації додатково можуть використовуватися методи оптимізації дискретних моделей, які ґрунтуються на локальних перебудовах елементів. Наприклад, при використанні на кожній ітерації операції «flip» [263, 379] для трикутників можна гарантувати, що дискретна модель буде відповідати критерію Делоне.

4.2.4 Приклади застосування методу

Розглянемо результати застосування алгоритму пошуку мінімуму функціоналу відстані-довжини, для оптимізації дискретної моделі, яка зображена на рис. 4.12, а. На рис. 4.14 наведено результати застосування алгоритму, якщо $\kappa = 10^{-3}$ і $\varepsilon = 10^{-6}$. Колір елемента показує значення найменшого кута у градусах (вважається, що чим більшим є мінімальний кут, тим якісніша модель). Можна помітити, що, починаючи з чотирьох ітерацій, злам поверхні біля отвору апроксимується вузлами та ребрами елементів, але при цьому залишаються гострі (менше 10 градусів) кути.

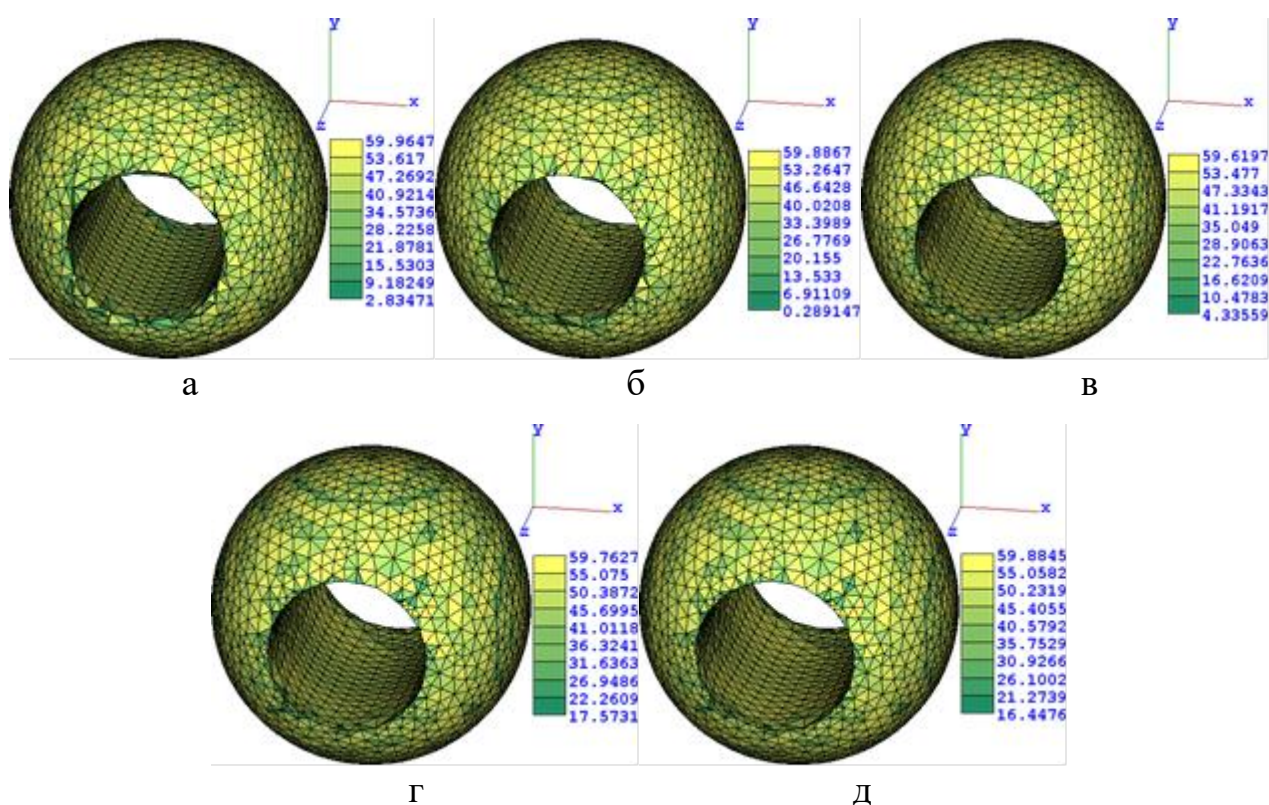


Рисунок 4.14 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у моделі кулі з круговим отвором (трикутники):

- а – 1 ітерація
- б – 2 ітерації
- в – 4 ітерації
- г – 8 ітерацій
- д – 16 ітерацій

Запропонований алгоритм може застосовуватися також для покращення дискретних моделей меж об'єктів на базі чотирикутних елементів. Наприклад, для дискретної моделі, яка зображена на рис. 4.12, в, результати застосування запропонованого алгоритму, якщо $\kappa = 10^{-3}$ і $\varepsilon = 10^{-6}$, матимуть вигляд, показаний на рис. 4.15. Колір елемента показує значення найбільшого кута у градусах (у випадку чотирикутників, чим більшим є максимальний кут, тим ближче він до виродженого елемента). Як й у випадку трикутників, починаючи з чотирьох ітерацій, злам поверхні біля отвору апроксимується вузлами та ребрами елементів, але залишаються елементи, які за формою близькі до трикутників.

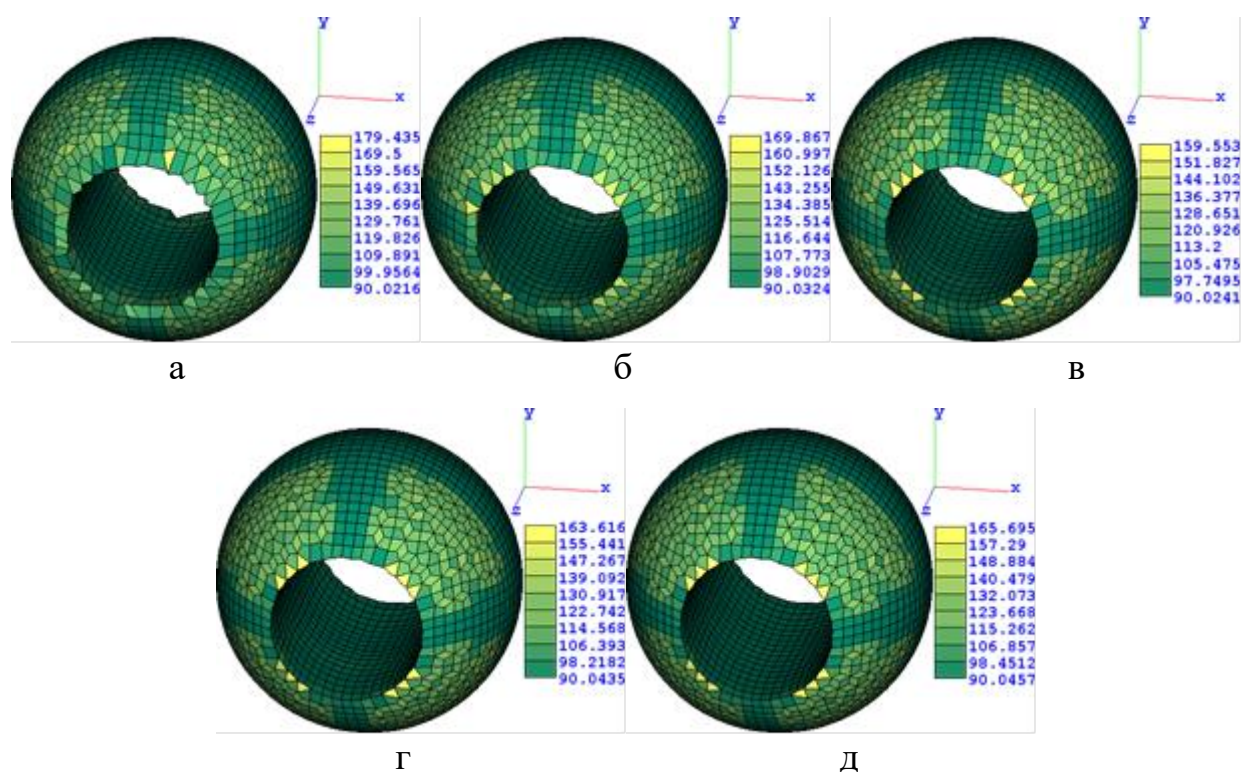


Рисунок 4.15 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у моделі кулі з круговим отвором (чотирикутники):

а – 1 ітерація

б – 2 ітерації

в – 4 ітерації

г – 8 ітерацій

д – 16 ітерацій

При дослідженні впливу параметра κ другий доданок у формулі (4.4) можна розглядати як силу тяжіння центрів елементів до поверхні. Тоді жорсткість κ визначає можливість «опору» системи пружин між вузлами руху вузлів по поверхні. Відповідно, чим більше κ , тим буде ближче результат мінімізації до згладжування Лапласа, а зі зменшенням значення κ більш точно будуть апроксимувати злами, але можливе отримання близько розташованих один до одного вузлів. На рис. 4.16 показані результати варіювання κ для 8 ітерацій покращення сітки.

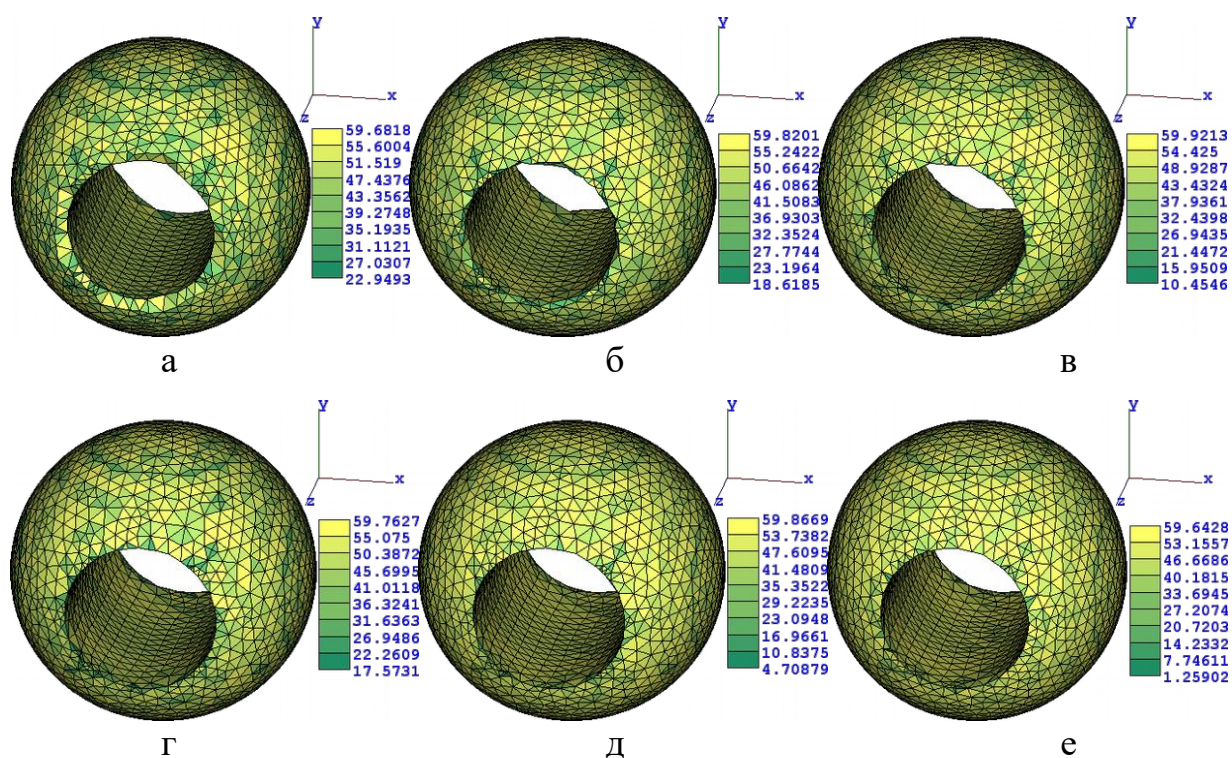


Рисунок 4.16 – Вплив параметра κ :

а – $\kappa = 0,1$

б – $\kappa = 0,01$

в – $\kappa = 0,005$

г – $\kappa = 0,001$

д – $\kappa = 0,0005$

е – $\kappa = 0,0001$

Модель кубика зі стороною рівною 2, у якому два кругових отвори (твірні, паралельні Ox і Oz) радіуса 0,5 і циліндричні ручки (твірні, паралельні Oz) того само радіуса й довжиною 1 (рис. 4.17), можна задати такою формулою:

$$w(x, y, z) = [\text{cuboid}(x, y, z, 2, 2, 2) \vee (\text{circle}(x, z; 0, 5) \wedge \text{stripe}(y, 4))] \wedge \neg \text{circle}(x, y; 0, 5) \wedge \neg \text{circle}(y, z; 0, 5), \quad (4.5)$$

де cuboid – функція (2.24);

circle – функція (2.7);

stripe – функція (2.11).

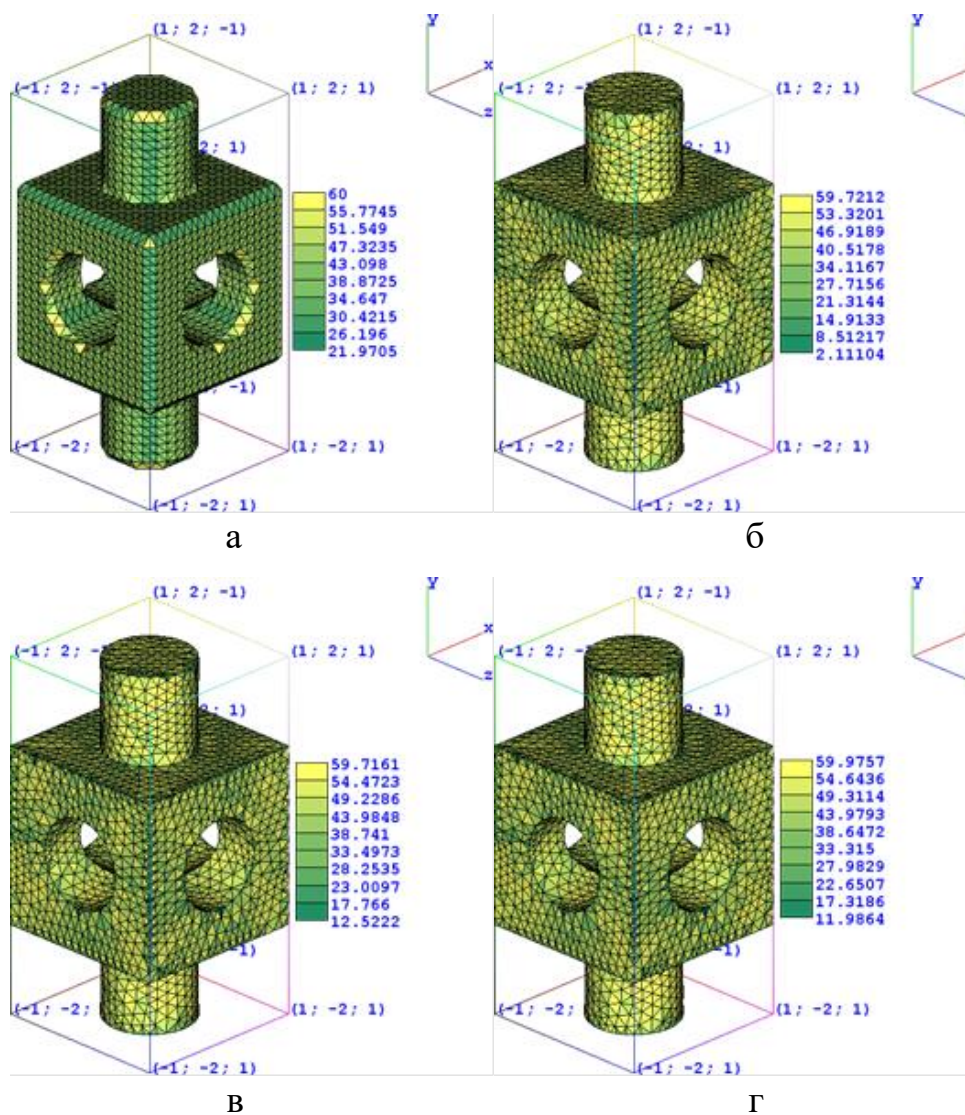


Рисунок 4.17 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у моделі «кубик з ручками» (трикутники):

- а – вихідна дискретна модель
- б – 4 ітерації оптимізації
- в – 8 ітерацій оптимізації
- г – 16 ітерацій оптимізації

На рис. 4.17, а показана вихідна дискретна модель на базі трикутних елементів, а також результати застосування алгоритму локальної мінімізації функціоналу відстані-довжини, якщо $\kappa = 10^{-3}$ і $\varepsilon = 10^{-6}$ (рис. 4.17, б–г). Аналогічно, на рис. 4.18, а–г показані результати для покращення сітки чотирикутних елементів.

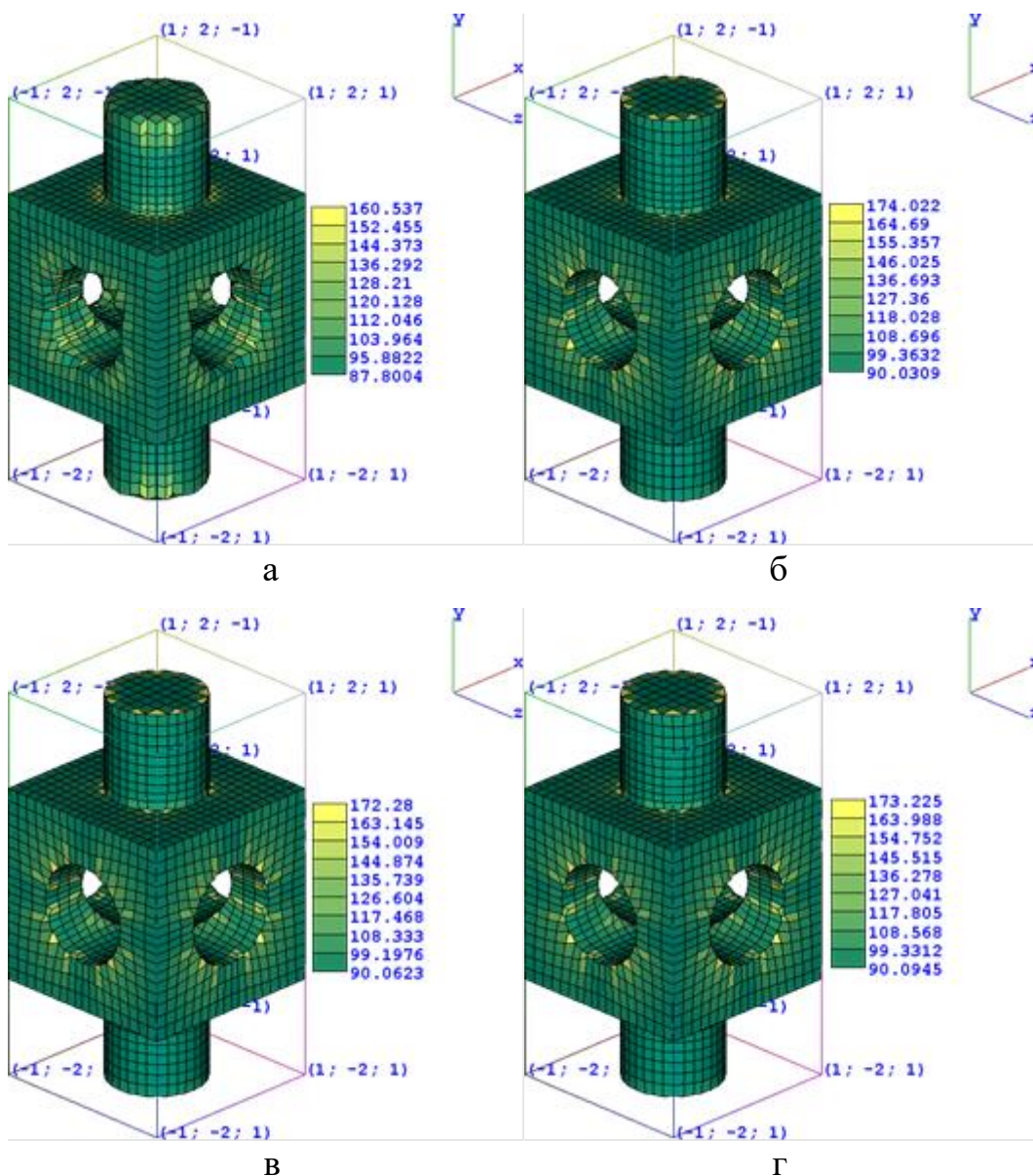


Рисунок 4.18 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у моделі «кубик з ручками» (чотирикутники):

а – вихідна дискретна модель

б – 4 ітерації оптимізації

в – 8 ітерацій оптимізації

г – 16 ітерацій оптимізації

Запропонований алгоритм можна застосувати й до покращення моделей гладких поверхонь. Наприклад, для поверхні «genus3» [223], яку можна подати такою формулою:

$$\text{genus3}(x, y, z) = -r_z^4 z^2 + \left(1 - \frac{x^2}{r_x^2} - \frac{y^2}{r_y^2}\right) \times \\ \times ((x - x_1)^2 + y^2 - r_1^2)((x + x_1)^2 + y^2 - r_1^2)(x^2 + y^2 - r_1^2),$$

при значеннях параметрів: $r_x = 6$, $r_y = 3,5$, $r_z = 4$, $r_1 = 1,2$ і $x_1 = 3,9$, результати покращення дискретних моделей ($\kappa = 10^{-3}$ і $\varepsilon = 10^{-6}$) набудуть вигляд, наведений на рис. 4.19.

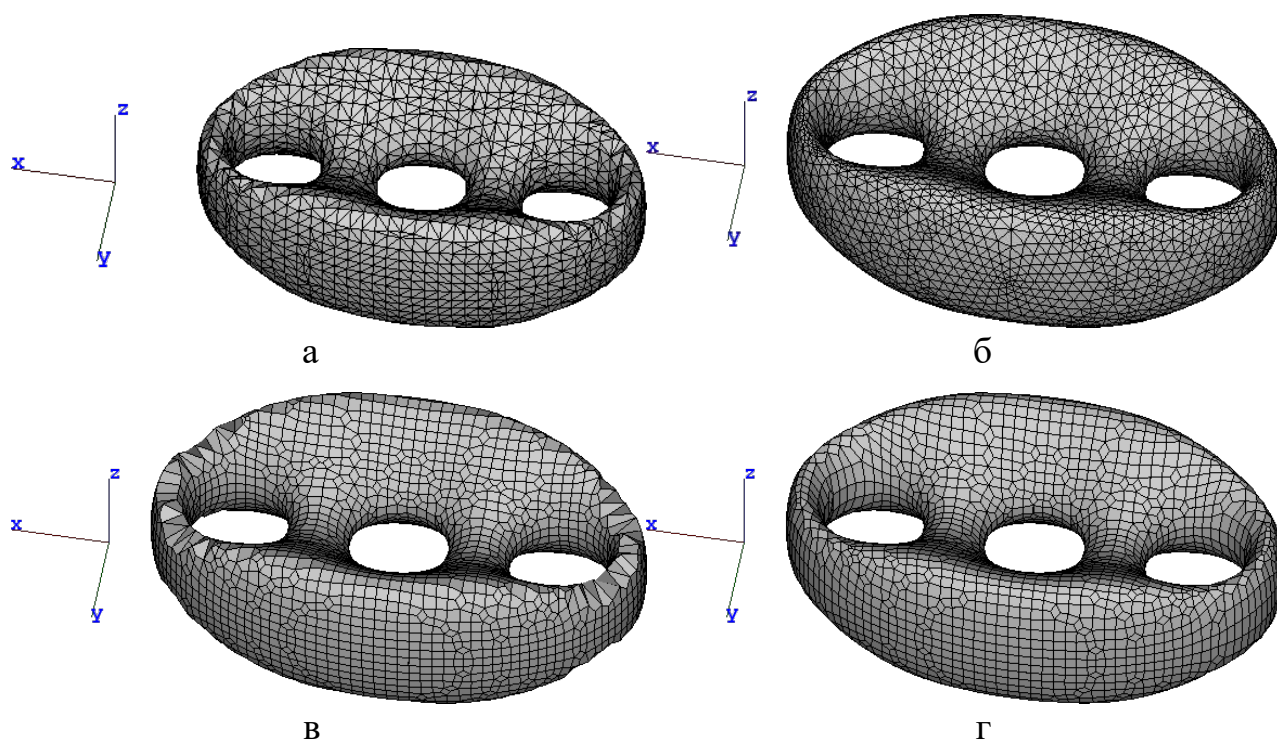


Рисунок 4.19 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у моделі «genus3»:

- а – вихідна дискретна модель на базі трикутників
- б – 8 ітерацій оптимізації для моделі на базі трикутників
- в – вихідна дискретна модель на базі чотирикутників
- г – 8 ітерацій оптимізації для моделі на базі чотирикутників

Отже, застосування запропонованого алгоритму до дискретних моделей гладких поверхонь буде спричиняти згущення числа вузлів у областях з найбільшою кривиною.

4.2.5 Двовимірний випадок

Описані вище функціонал і алгоритм у окремому випадку можуть застосовуватися для покращення меж форм двовимірних об'єктів. У такому випадку елементом межі виступає відрізок прямої, знаходження відстані до межі від якого дозволить обчислювати функціонал вигляду (4.4). Алгоритм мінімізації функціоналу не потребує змін, тому що пошук виконується у l -координатах суміжних вузлів.

Наприклад, для моделі пластини, заданої формулою (2.14), якщо як вихідні використовувати дискретні моделі ліній рівня (рис. 4.10), то у результаті застосування 8 ітерацій оптимізації для кожної моделі лінії рівня, результат набуде вигляд, наведений на рис. 4.20.

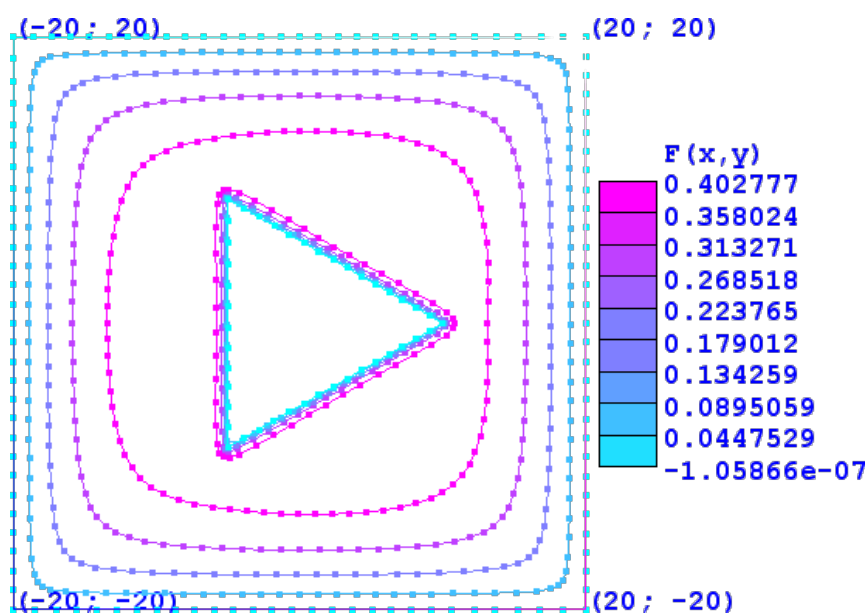


Рисунок 4.20 – Застосування локальної мінімізації функціоналу відстані-довжини для оптимізації положень вузлів у двовимірних дискретних моделях

4.2.6 Аналіз результатів

Отримані результати моделювання меж об'єктів (рис. 4.14–4.19) показують, що запропонований метод згладжування дискретних моделей меж об'єктів, який заснований на локальній мінімізації функціоналу відстані-довжини, дозволяє покращити точність дискретних моделей в околах особливостей (зламів поверхні). Водночас точність апроксимації особливостей буде залежати від параметра жорсткості ребер елементів вихідної дискретної моделі, що показано на рис. 4.16.

Перевага запропонованого метода полягає у тому, що (на відміну від результатів [298, 302, 358–359, 379–380]) він може застосовуватися і до трикутників, і до чотирикутників, а також він не вимагає вставки додаткових вузлів і елементів.

Необхідно зазначити, що даний метод покращення дискретних моделей не враховує метрик якості (мінімальних або максимальних кутів, співвідношень довжин сторін тощо). У результаті як недолік можна зазначити можливість одержання моделей з дуже гострими або тупими кутами. Також, як можна побачити з рис. 4.15 і 4.18 поблизу зламів поверхні можна отримати чотирикутники, за формою близькі до трикутників (один з кутів більше 175 градусів), усунути які без додаткового топологічного перебудування неможливо.

Запропонований метод можна використовувати спільно з методами локального згладжування для внутрішніх вузлів дискретних моделей форм об'єктів. У результаті можна сформувати єдину процедуру для всіх вузлів моделі.

4.3 Метод локальних перебудов елементів дискретних моделей меж об'єктів

Описані вище методи генерації та згладжування дискретних моделей меж об'єктів використовують тільки операції переміщення вузлів без зміни топології елементів (зв'язків між вузлами). Проте, у багатьох випадках зміна зв'язків між

вузлами дозволяє покращити характеристики дискретної моделі без втрати точності апроксимації геометричних особливостей межі об'єкта.

Наприклад, однією з найпростіших і водночас поширених операцій покращення дискретних моделей на базі трикутних елементів є «flip» [263, 379]. В основу цієї операції покладено ідею зміни спільного ребра (або діагоналі чотирикутника, утвореного вузлами) для пари сусідніх трикутників (рис. 4.21). «Flip» застосовують для приведення дискретних моделей у відповідність критерію Делоне [26], однією з важливих властивостей якого є максимізація мінімальних кутів у трикутниках моделі. У такому випадку діагональ змінюють тільки, якщо описане коло одного трикутника включає у себе всі вершини сусіднього.

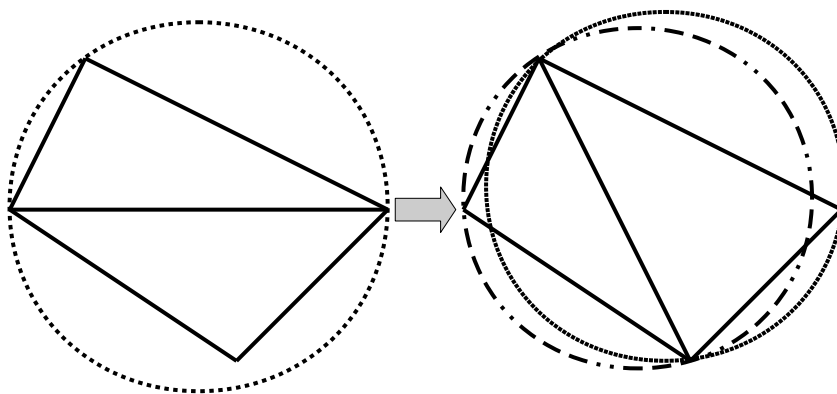


Рисунок 4.21 – Операція «flip» для трикутників

Для чотирикутників можна запропонувати аналогічну операцію зміни діагоналі (рис. 4.22). Для двох сусідніх чотирикутників існує два можливих варіанти зміни діагоналі. Отже, порівнюючи максимальні вузли у вихідній конфігурації з двох альтернативних необхідно обрати той, що з найменшим максимальним кутом (або з найбільшим мінімальним). Водночас необхідною умовою застосування цієї операції є опуклість фігури, утвореної вершинами чотирикутників.

У обох випадках алгоритм полягає у послідовній перевірці необхідності застосування операції «flip» та її виконанні за необхідністю. Елементи необхідно перевіряти на відповідність критерію поки були випадки зміни діагоналі (критерієм зупинки є відсутність перебудов на елементах за ітерацію).

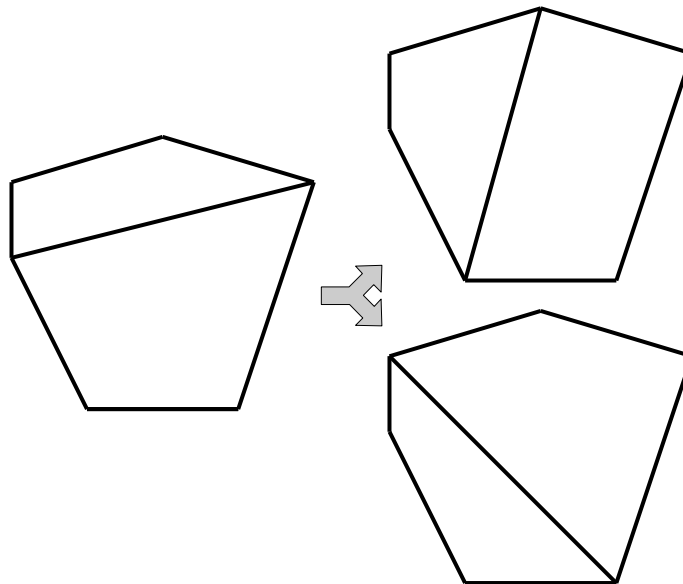


Рисунок 4.22 – Операція «flip» для чотирикутників

Підсумковий алгоритм застосування операції «flip» можна подати у такому вигляді:

algorithm flip

input:

$\mathbf{M}$ – дискретна модель межі об'єкта

output:

$\mathbf{M}$ – дискретна модель межі об'єкта

begin

do

$f \leftarrow \text{false}$

for each $e \in \mathbf{M.E}$ **do**

for $i \leftarrow 1$ **to** $|e|$ **do**

$k_e \leftarrow (\mathbf{M}.\text{incident}(e[i]) \cap \mathbf{M}.\text{incident}(e[i+1])) \setminus \mathbf{M}.\text{code}(e)$

if $\text{is\_flip}(\mathbf{M}.\text{code}(e), k_e)$ **then**

$\mathbf{M}.\text{flip}(\mathbf{M}.\text{code}(e), k_e)$

$f \leftarrow \text{true}$

end if

end for

```

end for
while  $f = \text{true}$ 
end

```

На рис. 4.23 порівнюються результати 8 ітерацій згладжування дискретної моделі межі об'єкта, заданого формулою (4.1), без використання процедури «flip» і з її використанням. Мінімальний кут в трикутних елементах дискретної моделі зростає з $7,83^\circ$ до $13,83^\circ$. Для чотирикутних елементів максимальний кут зменшується з $178,1^\circ$ до $174,89^\circ$.

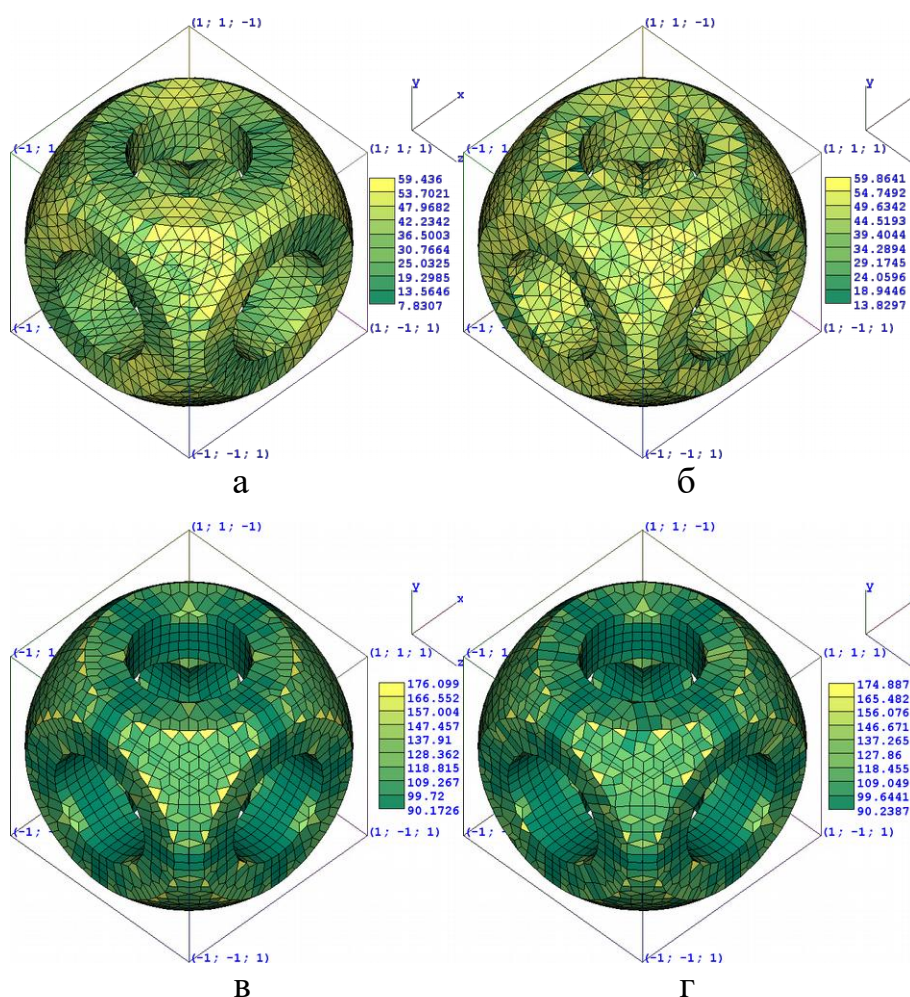


Рисунок 4.23 – Порівняння результатів застосування операції «flip»:

- а – результат згладжування для трикутників без використання операції «flip»
- б – результат згладжування для трикутників з використанням операції «flip»
- в – результат згладжування для чотирикутників без використання операції «flip»
- г – результат згладжування для чотирикутників з використанням операції «flip»

Для моделі пробки, заданої формулою (2.29), на рис. 4.24 порівняно результати оптимізації відповідної дискретної моделі межі.

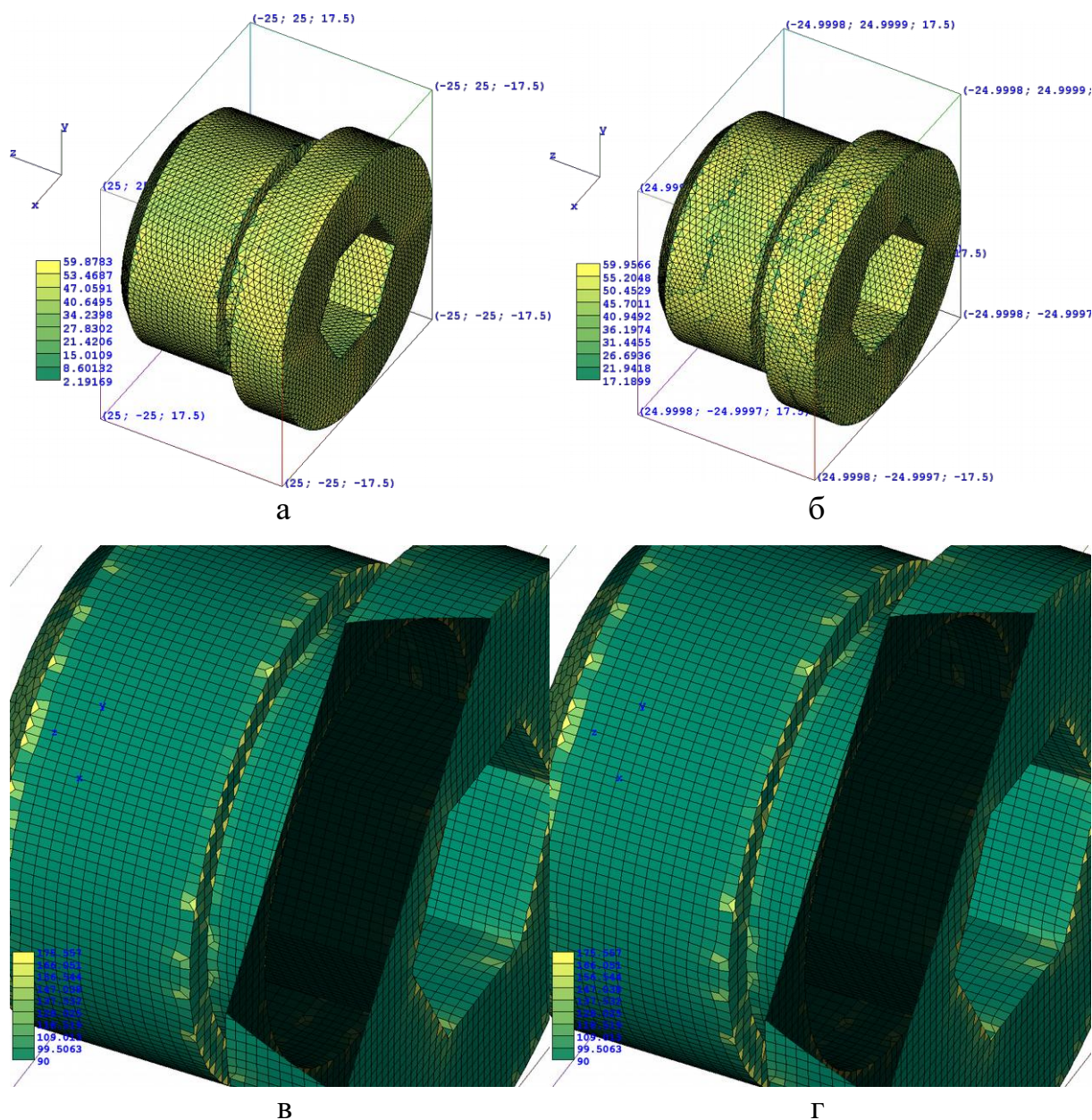


Рисунок 4.24 – Порівняння результатів застосування операції «flip»:

- а – результат згладжування для трикутників без використання операції «flip»
- б – результат згладжування для трикутників з використанням операції «flip»
- в – результат згладжування для чотирикутників без використання операції «flip»
- г – результат згладжування для чотирикутників з використанням операції «flip»

Найбільш гострий кут у дискретній моделі межі пробки, заснованій на трикутниках, якщо не використовувати операцію «flip», дорівнює $2,19^\circ$ (рис. 4.24, а).

Якщо «flip» була використана, то у цій моделі мінімальний кут дорівнює $17,19^\circ$ (рис. 4.24, б). У дискретній моделі межі пробки на базі чотирикутників максимальний кут не змінився і дорівнює $175,56^\circ$ (рис. 4.24, в–г), що обумовлено апроксимацією зламів межі парою суміжних ребер.

Аналіз отриманих результатів дозволяє зробити висновок, що застосування операції «flip» (яка не вимагає вставки додаткових вузлів) дозволяє поліпшити геометричні характеристики дискретної моделі. Водночас результат її застосування для трикутників більш помітний, ніж для чотирикутників. Це обумовлено тим, що вимога утворення опуклої фігури шістьма вузлами істотно знижує можливості її практичного застосування.

При використанні чотирикутників можливим способом зменшення великих кутів в елементах є розбиття їх на трикутники (проведенням ребра через найбільший кут) з подальшим перетворенням шляхом серединного розбиття. В такому випадку, після розбиття всіх необхідних чотирикутників на трикутники, формується нова множина чотирикутників шляхом з'єднання кожного вузла з центрами сусідніх у ньому сторін і геометричним центром елемента (рис. 4.25). Водночас для нових вузлів потрібно знайти їх проекції на межу об'єкта.

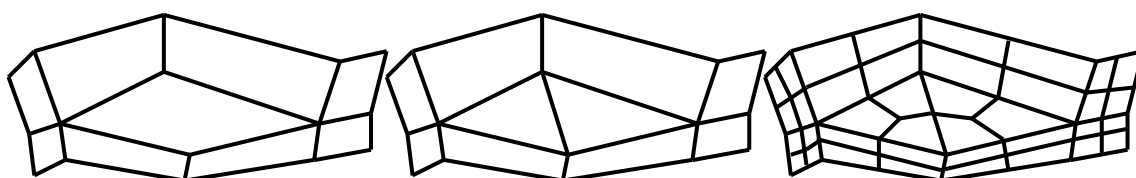


Рисунок 4.25 – Розбиття чотирикутників

Отже, розбиття елементів дозволяє зменшити значення максимального кута для дискретних моделей меж об'єктів на базі чотирикутників. Наприклад, для дискретної моделі межі «кубика», заданого формулою (4.1), (рис. 4.23, г) значення максимального кута в елементі зменшується з початкових $174,89^\circ$ до $155,13^\circ$ (рис. 4.26). Водночас зростає кількість вузлів і елементів. Наприклад, у вихідній дискретній моделі межі «кубика» було 3064 вузи та 3072 елементи, а у результаті розбиття – 13782 вузли та 13790 елементів.

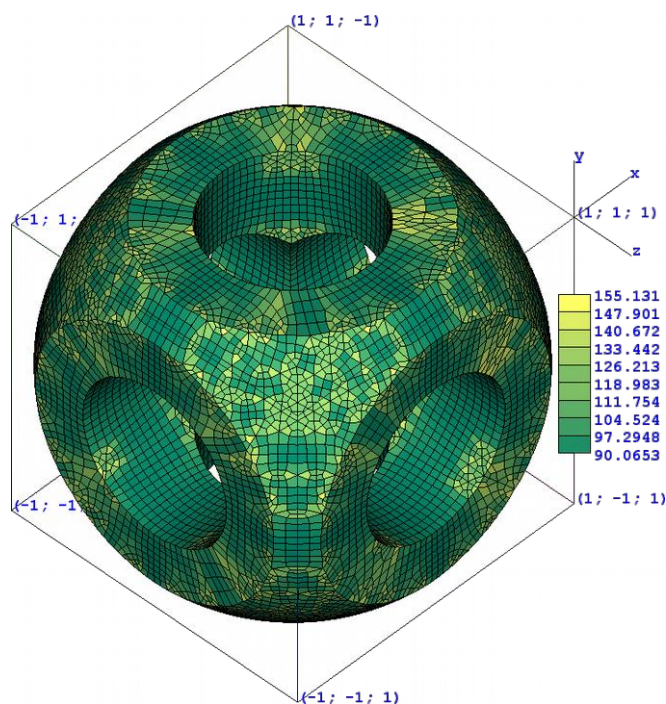


Рисунок 4.26 – Приклад використання розбиття чотирикутників

Висновки до розділу 4

Розроблений метод побудови дискретних моделей меж об'єктів теоретично дозволяє використання елементів довільної форми. Водночас щільність вузлів у підсумковій моделі залежить від щільності вузлів у вихідній. Проте, запропонований метод не враховує геометричних особливостей форми об'єкта і, отже, повинна застосовуватися додаткова оптимізація.

Запропонований метод згладжування дискретних моделей меж об'єктів ґрунтується на локальній мінімізації функціоналу відстані-довжини, що дозволяє точніше апроксимувати геометричні особливості. Водночас для чисельної реалізації пошуку мінімуму запропонована модифікація методу покоординатного спуску Гауса-Зейделя на випадок пошуку в l -координатах. Проведені дослідження показали, що за малої кількості ітерацій (менше чотирьох) можна отримати дискретні моделі форм об'єктів низької якості. Починаючи з четвертої ітерації, геометричні особливості форм об'єктів вже апроксимуються вузлами. Водночас виконання більше восьми ітерацій суттєво не змінює якості моделі. У результаті застосування цього методу

мінімальні кути для трикутників несуттєво збільшуються, а у чотирикутниках можливо отримати елементи, за формулю близькі до трикутників, отже, результати загладжування потребують додаткового оброблення топологічними перебудовами елементів.

Метод локальних перебудов елементів дискретних моделей меж об'єктів дозволяє збільшити значення мінімального кута в елементі для дискретних моделей на базі трикутників і зменшити значення максимального кута в елементі для у дискретних моделях на базі чотирикутників.

Незалежність методів генерації та згладжування дискретних моделей меж об'єктів дозволяє зробити висновок про можливість їх програмної реалізації на базі шаблону проектування «Стратегія». У результаті можна підвищити технологічність розробленої на їх базі САПР машинобудування.

Результати четвертого розділу опубліковано у роботах [112, 158, 177, 179, 182, 238].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилання [26, 188, 190, 214, 218, 223, 233–236, 251, 253, 259, 263, 266, 293, 297, 298, 300, 302, 313, 345, 358–360, 366, 375, 379–380, 397, 428–431, 436, 444, 487].

5 ДИСКРЕТНІ МОДЕЛІ ФОРМ ОБ'ЄКТІВ НА БАЗІ СОЛІДІВ

Розроблені у попередньому розділі методи генерації дискретних моделей меж об'єктів можна використовувати для розв'язання задач комп'ютерної графіки та чисельного дослідження поведінки оболонок. Проте, у практиці інженерного аналізу часто виникає необхідність використання дискретних моделей, які апроксимують всю геометричну область, а не тільки її межу. В такому випадку використовують моделі на базі солідів: трикутників або чотирикутників на площині та тетраедрів або шестигранників у просторі. Водночас дискретні моделі меж об'єктів можуть бути вихідними даними для методів генерації дискретних моделей на базі солідів.

5.1 Дво- і тривимірні триангуляції

Для трикутників і тетраедрів, які спільно називають симплексами (симплекс – це фігура, яка є n -вимірним узагальненням трикутника), розробляють спільні методи побудови дискретних моделей форм об'єктів. Зазвичай ці методи поєднують під спільною назвою «триангуляція».

Як показано у пункті 1.3.2, для множини точок на площині та планарних графів розроблені ефективні методи побудови триангуляції на основі критерію Делоне [26]. Вважають, що дискретна модель відповідає цьому критерію, якщо у коло, описане навколо довільного трикутника, потрапляють тільки вузли, які належать цьому трикутнику.

Дискретна модель форми об'єкта на базі трикутників, яка відповідає критерію Делоне, має такі властивості [135, 307]:

- єдиність, якщо довільні чотири точки не знаходяться на одному колі;
- максимальне значення мінімального кута в елементах дискретної моделі, які побудовані на даній множині точок;

– максимальна сума радіусів описаних кіл для трикутників, побудованих на даній множині точок.

У тривимірному випадку, коли елементами є тетраедри, а для перевірки критерію Делоне використовують описані навколо них сфери, властивість максимізації мінімального кута в елементах не зберігається. Проте, зважена за об'ємами інцидентних елементів сума квадратів довжин сторін буде мінімальною для заданої множини точок тільки при виконанні критерію Делоне [406].

5.1.1 Алгоритм метода тріангуляції

В обох випадках для побудови тріангуляції Делоне можна скористатися так званою суперобластю. Суперобласть – це елементарний опуклий багатогранник з трикутними гранями, який повністю включає всі точки межі форми об'єкта. У двовимірному випадку супер-областю може бути трикутник або квадрат, а в тривимірному – тетраедр або куб, грані якого розбиті на трикутники. Далі з'єднавши границю суперобласті з першим вузлом із дискретної моделі межі об'єкта отримаємо початкову тріангуляцію Делоне. Кожен наступний вузол із дискретної моделі межі об'єкта вставляється у тріангуляцію Делоне шляхом видалення всіх симплексів, в описані сфери яких він потрапляє, та з'єднанням з гранями отриманого після видалення багатогранника. Після видалення елементів, які з'єднані з вершинами супер області, буде отримана тріангуляція Делоне для опуклого многовиду, утвореного вузлами дискретної моделі межі об'єкта. Однак, межа об'єкта може бути неопуклою. У такому випадку в тріангуляцію потрібно додавати проекції середини грані або ребра елемента на межу об'єкта, якщо ця грань або ребро поза власних вершин перетинається межею. Отже, для побудови дискретної моделі виду (3.2) для форми об'єкта, яка задана функціонально, можна записати такий алгоритм генерації тріангуляції Делоне:

algorithm delaunay

input:

F – функція, яка подає форму об'єкта Ω

$\mathbf{B}$ – дискретна модель межі об'єкта Ω

ε – чисельний нуль

output:

T – дискретна модель форми об'єкта Ω на базі симплексів

begin

$T \leftarrow \text{initial\_triangulation}(\mathbf{B})$

for each $n \in \mathbf{B.N}$ **do**

$\text{insert\_delaunay}(T, n, \varepsilon)$

end for

do

$\text{is\_subdivided} \leftarrow \text{false}$

for each $e \in T.E$ **do**

for $i \leftarrow 1$ **to** $|e|$ **do**

for each $b \in \mathbf{B.E}$ **do**

if $\text{is\_crossed}(\mathbf{B.N}(b), e[i], e[i+1], \varepsilon) = \text{true}$ **then**

$$C \leftarrow \frac{1}{|b|} \sum_{j \in b} \mathbf{B}[j]$$

$$h \leftarrow \min_{(j \neq k) \in b} \|\mathbf{B}[j] - \mathbf{B}[k]\|$$

$\text{insert\_delaunay}(T, \text{find\_border}(C, F, h, \varepsilon), \varepsilon)$

$\text{is\_subdivided} \leftarrow \text{true}$

end if

end for

end for

end for

while $\text{is\_subdivided} = \text{true}$

for each $e \in T.E$ **do**

```


$$C \leftarrow \frac{1}{|e|} \sum_{j \in e} \mathbf{T}[j]$$

if  $F(C) < 0$  then
    remove_element( $\mathbf{T}$ ,  $\mathbf{T}.\text{code}(e)$ )
end if
end for
return  $\mathbf{T}$ 

```

Алгоритм побудови початкової триангуляції `initial_triangulation` потрібно реалізувати з урахуванням розмірності простору. У двовимірному випадку як супер-областю можна скористатися кубом, який повністю охоплює всі вузли моделі межі області. У результаті початкова триангуляція буде містити 4 трикутника.

algorithm `initial_triangulation`

input:

$\mathbf{B}$ – дискретна модель межі об'єкта Ω

output:

$\mathbf{T}$ – початкова триангуляція

begin

$$d \leftarrow 2 \max_{(n \neq m) \in \mathbf{B} \cdot \mathbf{N}} \|n - m\|$$

$$C \leftarrow \frac{1}{|\mathbf{B} \cdot \mathbf{N}|} \sum_{i=1}^{|\mathbf{B} \cdot \mathbf{N}|} \mathbf{B}[i]$$

$$\mathbf{T} \leftarrow (\emptyset, \emptyset)$$

$$P_1 \leftarrow C + (-d; -d)$$

$$P_2 \leftarrow C + (d; -d)$$

$$P_3 \leftarrow C + (d; d)$$

$$P_4 \leftarrow C + (-d; d)$$

insert_element($\mathbf{T}$, $\{P_1, P_2, \mathbf{B}[1]\}$)

insert_element($\mathbf{T}$, $\{P_2, P_3, \mathbf{B}[1]\}$)

insert_element($\mathbf{T}$, $\{P_3, P_4, \mathbf{B}[1]\}$)

```

insert_element(T, { $P_4$ ,  $P_1$ , B[1]})
return T

```

Наприклад, для дискретної моделі межі пластини (рис. 5.1, а), яка задана формулою (2.14), початкова тріангуляція прийме вигляд, наведений на рис. 5.1, б.

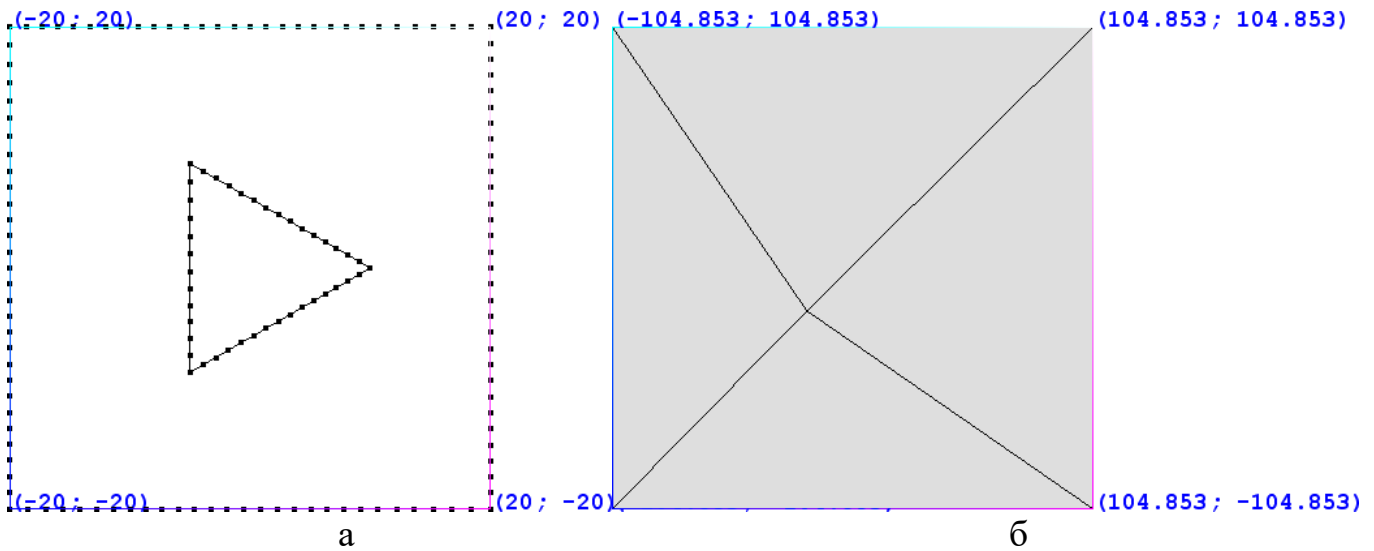


Рисунок 5.1 – Початкова тріангуляція:

а – вихідна дискретна модель межі об'єкта

б – початкова тріангуляція

Якщо у тривимірному випадку супер-областю буде куб, початкова тріангуляція буде мати 12 тетраедрів (по два на кожну грань куба). Отже, алгоритм `initial_triangulation` можна записати у такому вигляді:

algorithm `initial_triangulation`

input:

B – дискретна модель межі об'єкта Ω

output:

T – початкова тріангуляція

begin

$$d \leftarrow 2 \max_{(n \neq m) \in \mathbf{B.N}} \|n - m\|$$

$$C \leftarrow \frac{1}{|\mathbf{B} \cdot \mathbf{N}|} \sum_{i=1}^{|\mathbf{B} \cdot \mathbf{N}|} \mathbf{B}[i]$$

$$\mathbf{T} \leftarrow (\emptyset, \emptyset)$$

$$P_1 \leftarrow C + (-d; -d; -d)$$

$$P_2 \leftarrow C + (d; -d; -d)$$

$$P_3 \leftarrow C + (d; d; -d)$$

$$P_4 \leftarrow C + (-d; d; -d)$$

$$P_5 \leftarrow C + (-d; -d; d)$$

$$P_6 \leftarrow C + (d; -d; d)$$

$$P_7 \leftarrow C + (d; d; d)$$

$$P_8 \leftarrow C + (-d; d; d)$$

$$\text{insert\_element}(\mathbf{T}, \{P_1, P_2, P_3, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_1, P_3, P_4, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_4, P_3, P_8, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_8, P_3, P_7, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_5, P_7, P_6, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_5, P_8, P_7, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_1, P_4, P_8, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_1, P_8, P_5, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_2, P_6, P_3, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_3, P_6, P_7, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_8, P_3, P_7, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_1, P_5, P_2, \mathbf{B}[1]\})$$

$$\text{insert\_element}(\mathbf{T}, \{P_2, P_5, P_6, \mathbf{B}[1]\})$$

return T

Алгоритм `insert_delaunay(T, n, ε)` додає вузол n у дискретну модель $\mathbf{T}$. У цьому алгоритмі, після перевірки існування такого вузла у $\mathbf{T}$ видаляються всі елементи, в описане коло (або сферу тривимірному випадку) яких потрапляє вузол n . У результаті утворюється багатокутник (багатогранник), з'єднання ребер (граней)

якого з n дозволяє утворити множину елементів. Алгоритм можна формалізувати у такому вигляді:

algorithm insert_delaunay

input:

T – триангуляція

n – вузол

ε – чисельний нуль

output:

T – триангуляція

begin

$is\_exist \leftarrow \text{false}$

for each $n_t \in T.N$ **do**

if $\|n.v - n_t.v\| < \varepsilon$ **then**

$is\_exist \leftarrow \text{true}$

end if

end for

if $is\_exist = \text{false}$ **then**

$L \leftarrow (\emptyset, \emptyset)$

for each $e \in T.E$ **do**

$(C, r) \leftarrow \text{circum\_circle}(T.N(e))$

if $\|C - n.v\| < r$ **then**

insert_element($L, T.N(e), \varepsilon$)

remove_element($T, T.code(e)$)

end if

end for

$B \leftarrow \delta(L)$

for each $e \in B$ **do**

insert_element($T, \{B.N(e), n\}, \varepsilon$)

end for

end if

end

Алгоритм `circum_circle` необхідний для перевірки потрапляння точки в описане коло або сферу. Для цього необхідно знайти координати її центру та радіус. При розв'язанні цієї задачі у двовимірному випадку необхідно обчислити значення чотирьох визначників матриць 3×3 [395]. У результаті алгоритм можна подати у такому вигляді:

algorithm `circum_circle`

input:

V – список координат вершин двовимірного трикутного елемента

output:

C – координати центра описаного навколо V кола

r – радіус кола, описаного навколо V

begin

$$a \leftarrow \begin{vmatrix} V[1].x & V[1].y & 1 \\ V[2].x & V[2].y & 1 \\ V[3].x & V[3].y & 1 \end{vmatrix}$$

$$b_x \leftarrow - \begin{vmatrix} V[1].x^2 + V[1].y^2 & V[1].y & 1 \\ V[2].x^2 + V[2].y^2 & V[2].y & 1 \\ V[3].x^2 + V[3].y^2 & V[3].y & 1 \end{vmatrix}$$

$$b_y \leftarrow \begin{vmatrix} V[1].x^2 + V[1].y^2 & V[1].x & 1 \\ V[2].x^2 + V[2].y^2 & V[2].x & 1 \\ V[3].x^2 + V[3].y^2 & V[3].x & 1 \end{vmatrix}$$

$$c \leftarrow - \begin{vmatrix} V[1].x^2 + V[1].y^2 & V[1].x & V[1].y \\ V[2].x^2 + V[2].y^2 & V[2].x & V[2].y \\ V[3].x^2 + V[3].y^2 & V[3].x & V[3].y \end{vmatrix}$$

$$C \leftarrow \left(-\frac{b_x}{2a}; -\frac{b_y}{2a} \right)$$

$$r \leftarrow \frac{\sqrt{b_x^2 + b_y^2 - 4ac}}{2|a|}$$

return (C, r)

У тривимірному випадку для знаходження центру та радіусу описаної навколо тетраедра сфери необхідно знайти значення вже шести визначників матриць 4×4 , а алгоритм `circum_circle` матиме такий вигляд:

algorithm `circum_circle`

input:

V – список координат вершин тетраедра

output:

C – координати центра описаної навколо V сфери

r – радіус описаної навколо V сфери

begin

$$a \leftarrow \begin{vmatrix} V[1].x & V[1].y & V[1].z & 1 \\ V[2].x & V[2].y & V[2].z & 1 \\ V[3].x & V[3].y & V[3].z & 1 \\ V[4].x & V[4].y & V[4].z & 1 \end{vmatrix}$$

$$d_x \leftarrow \begin{vmatrix} V[1].x^2 + V[1].y^2 + V[1].z^2 & V[1].y & V[1].z & 1 \\ V[2].x^2 + V[2].y^2 + V[2].z^2 & V[2].y & V[2].z & 1 \\ V[3].x^2 + V[3].y^2 + V[3].z^2 & V[3].y & V[3].z & 1 \\ V[4].x^2 + V[4].y^2 + V[4].z^2 & V[4].y & V[4].z & 1 \end{vmatrix}$$

$$d_y \leftarrow \begin{vmatrix} V[1].x^2 + V[1].y^2 + V[1].z^2 & V[1].z & V[1].x & 1 \\ V[2].x^2 + V[2].y^2 + V[2].z^2 & V[2].z & V[2].x & 1 \\ V[3].x^2 + V[3].y^2 + V[3].z^2 & V[3].z & V[3].x & 1 \\ V[4].x^2 + V[4].y^2 + V[4].z^2 & V[4].z & V[4].x & 1 \end{vmatrix}$$

$$d_z \leftarrow \begin{vmatrix} V[1].x^2 + V[1].y^2 + V[1].z^2 & V[1].x & V[1].y & 1 \\ V[2].x^2 + V[2].y^2 + V[2].z^2 & V[2].x & V[2].y & 1 \\ V[3].x^2 + V[3].y^2 + V[3].z^2 & V[3].x & V[3].y & 1 \\ V[4].x^2 + V[4].y^2 + V[4].z^2 & V[4].x & V[4].y & 1 \end{vmatrix}$$

$$c \leftarrow \begin{vmatrix} \mathbf{V}[1].x^2 + \mathbf{V}[1].y^2 + \mathbf{V}[1].z^2 & \mathbf{V}[1].x & \mathbf{V}[1].y & \mathbf{V}[1].z \\ \mathbf{V}[2].x^2 + \mathbf{V}[2].y^2 + \mathbf{V}[2].z^2 & \mathbf{V}[2].x & \mathbf{V}[2].y & \mathbf{V}[2].z \\ \mathbf{V}[3].x^2 + \mathbf{V}[3].y^2 + \mathbf{V}[3].z^2 & \mathbf{V}[3].x & \mathbf{V}[3].y & \mathbf{V}[3].z \\ \mathbf{V}[4].x^2 + \mathbf{V}[4].y^2 + \mathbf{V}[4].z^2 & \mathbf{V}[4].x & \mathbf{V}[4].y & \mathbf{V}[4].z \end{vmatrix}$$

$$C \leftarrow \left(\frac{d_x}{2a}; \frac{d_y}{2a}; \frac{d_z}{2a} \right)$$

$$r \leftarrow \frac{\sqrt{d_x^2 + d_y^2 + d_z^2 - 4ac}}{2|a|}$$

return (C, r)

Алгоритм `is_crossed` використовується для перевірки перетину відрізком елемента дискретної моделі межі області. У двовимірному випадку цей алгоритм має такий вигляд:

algorithm `is_crossed`

input:

V – пара двовимірних точок, які відповідають елементу межі двовимірної області

P_1 – перша точка відрізка

P_2 – друга точка відрізка

ε – чисельний нуль

output:

true, якщо (P_1, P_2) перетинає **V**, false – інакше

begin

p $\leftarrow P_1$

r $\leftarrow P_2 - P_1$

q $\leftarrow \mathbf{V}[1]$

s $\leftarrow \mathbf{V}[2] - \mathbf{V}[1]$

d $\leftarrow \mathbf{r} \times \mathbf{s}$

if $|d| < \varepsilon$ **then**

return false

```

end if


$$t \leftarrow \frac{(\mathbf{q} - \mathbf{p}) \times \mathbf{s}}{d}$$



$$u \leftarrow \frac{(\mathbf{q} - \mathbf{p}) \times \mathbf{r}}{d}$$


if  $\varepsilon \leq t$  and  $t \leq 1 - \varepsilon$  and  $\varepsilon \leq u$  and  $u \leq 1 - \varepsilon$  then
    return true
end if
return false

```

У тривимірному випадку алгоритм `is_crossed` ґрунтується на перевірці перетину трикутника відрізком [372]. Його можна подати у такому вигляді:

algorithm `is_crossed`

input:

$\mathbf{V}$ – список тривимірних точок, які відповідають трикутному елементу

P_1 – перша точка відрізка

P_2 – друга точка відрізка

ε – чисельний нуль

output:

true, якщо (P_1, P_2) перетинає $\mathbf{V}$, false – інакше

begin

$\mathbf{o} \leftarrow P_1$

$\mathbf{d} \leftarrow P_2 - P_1$

$\mathbf{e}_1 \leftarrow \mathbf{V}[2] - \mathbf{V}[1]$

$\mathbf{e}_2 \leftarrow \mathbf{V}[3] - \mathbf{V}[1]$

$\mathbf{t} \leftarrow \mathbf{o} - \mathbf{V}[1]$

$\mathbf{p} \leftarrow \mathbf{d} \times \mathbf{e}_2$

$\mathbf{q} \leftarrow \mathbf{t} \times \mathbf{e}_1$

$$\begin{bmatrix} \tau \\ u \\ v \end{bmatrix} = \frac{1}{\mathbf{p} \cdot \mathbf{e}_1} \begin{bmatrix} \mathbf{q} \cdot \mathbf{e}_2 \\ \mathbf{p} \cdot \mathbf{t} \\ \mathbf{q} \cdot \mathbf{d} \end{bmatrix}$$

if $\tau < \varepsilon$ **or** $1 - \varepsilon < \tau$ **or** $u + v > 1 - \varepsilon$ **or** $u < \varepsilon$ **or** $v < \varepsilon$ **then**

return false

end if

return true

Останнім кроком алгоритму delaunay є видалення зовнішніх елементів. Для цього із результатів вставки вузлів у триангуляцію Делоне для суперобласті видаляються всі елементи, у середині яких функція області менша нуля (наприклад, супер-область – рис. 5.2, а, результат видалення зовнішніх елементів – рис. 5.2, б).

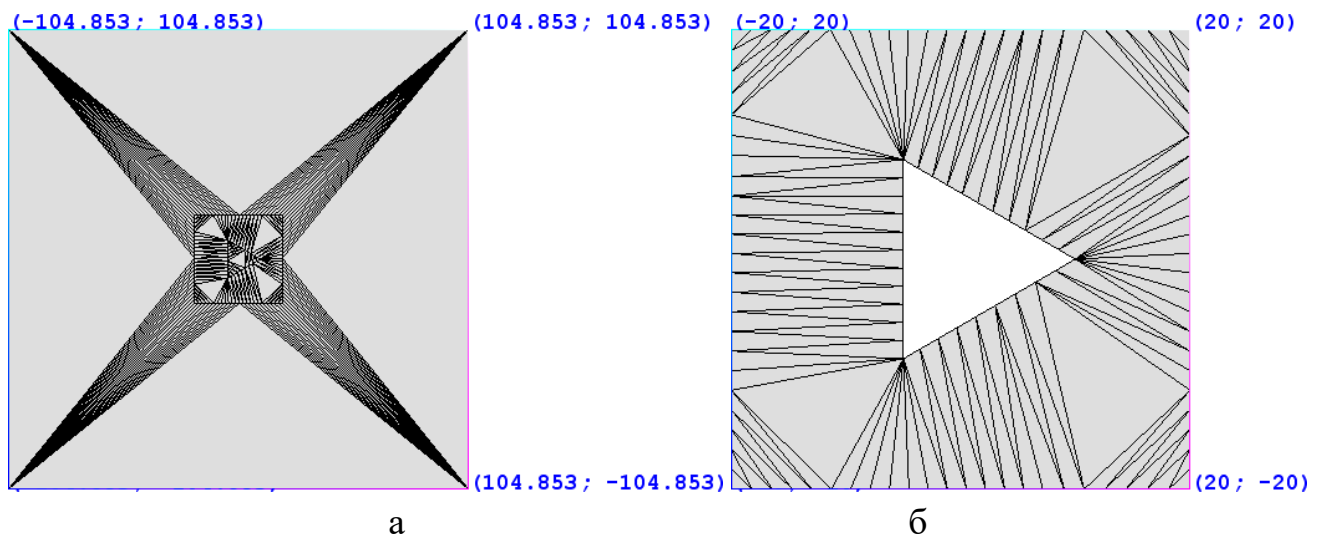


Рисунок 5.2 – Триангуляція Делоне:

а – побудована для суперобласті

б – результат видалення зовнішніх елементів для моделі пластинки

Необхідно відзначити, що у тривимірному випадку існують дискретні моделі меж геометричних об'єктів, для яких неможлива побудова множини тетраедрів без додавання нових вузлів [284]. Також у [417] доведено, що для вихідної дискретної моделі межі об'єкта перевірка існування множини тетраедрів, яка задовольняє критерію Делоне, є NP-повною.

У [439] доведена можливість побудови множини тетраедрів, що задовольняють критерію Делоне, якщо для кожного ребра вихідної триангуляції не існує вузлів, що належать до йому та потрапляють у сферу, діаметром якої є це ребро.

Алгоритм `delaunay` послідовно будує множину тетраедрів з перевіркою умови Делоне при вставці. Для забезпечення апроксимації меж об'єкта ребрами та гранями тетраедрів нові вузли додаються у точки перетину ребрами меж (якщо такі відрізняються від кінців ребра).

5.1.2 Алгоритм оптимізації чисельних характеристик дискретної моделі

З точки зору інженерного аналізу об'єктів машинобудування важливими є чисельні характеристики дискретних моделей форм. Наявність значних диспропорцій у елементах (дуже гострих кутів) зумовлює малі значення якобіана при скінченно-елементному аналізі з використанням місцевих координат. У роботі [418] доведено, що вставкою вузлів у центри трикутників, які не відповідають вимозі відносно гострого кута, і у середини граничних ребер, у діаметральні кола яких потрапляють вузли, що їм не належать, найбільш гострий кут буде не менший за $20,7^\circ$ (співвідношення сторін елементів буде не більше $\sqrt{2}$). Проте, ця техніка не можна застосовувати для тетраедрів, тому необхідно використовувати додаткове розбиття межових граней зі вставкою нового вузла у центри їх описаних кіл [440].

У результаті узагальнення методів, запропонованих у [418] і [440], на випадок використання дискретної моделі виду (3.2) при функціональному поданні форми об'єкта можна сформулювати такий алгоритм:

algorithm `refine_delaunay`

input:

D – дво- або тривимірна триангуляція Делоне форми об'єкта Ω

F – функція, яка подає форму об'єкта Ω

α – обмеження на відношення довжин сторін елемента

ε – чисельний нуль

output:

D – дво- або тривимірна триангуляція Делоне форми об'єкта Ω

begin

B $\leftarrow \delta(\mathbf{D})$

S $\leftarrow \text{encroached}(\mathbf{D}, \varepsilon) \cup \text{poor}(\mathbf{D}, \alpha)$

while **S** $\neq \emptyset$ **do**

if $s \in \mathbf{S}$ – сегмент **then**

$h \leftarrow 0,5 \|\mathbf{B}[s[1]] - \mathbf{B}[s[2]]\|$

$C \leftarrow \text{find\_border}(0,5 (\mathbf{B}[s[1]] + \mathbf{B}[s[2]]), F, h, \varepsilon)$

$\text{insert\_delaunay}(\mathbf{D}, C, \varepsilon)$

else if $s \in \mathbf{S}$ – тривимірний трикутник **then**

$C \leftarrow \text{circum\_circle}(\mathbf{D.N}(s))$

$\text{insert\_delaunay}(\mathbf{D}, C, \varepsilon)$

else if $s \in \mathbf{S}$ – двовимірний трикутник або тетраедр

$C \leftarrow \text{circum\_circle}(\mathbf{D.N}(s))$

$f \leftarrow \text{false}$

$\mathbf{B} \leftarrow \delta(\mathbf{D})$

if $\exists e \in \mathbf{B}: \text{is\_encroach}(\mathbf{B.N}(e), C, \varepsilon) = \text{true}$ **then**

$\mathbf{S} \leftarrow \mathbf{S} \cup e$

else

$\text{insert\_delaunay}(\mathbf{D}, C, \varepsilon)$

end if

end if

end while

return D

На першому етапі алгоритму формується множина, яка складається з сегментів, граней та елементів, що підлягають розбиттю. Для цього використано два

алгоритми: *encroached* і *poor*. Алгоритм *encroached* формує список ребер і граней (у тривимірному випадку). Для двовимірних дискретних моделей форм об'єктів його можна подати у такому вигляді:

algorithm *encroached*

input:

$\mathbf{D}$ – двовимірна триангуляція форми об'єкта Ω

ε – чисельний нуль

output:

$\mathbf{S}$ – множина межових ребер, які потрібно розбити

begin

$\mathbf{B} \leftarrow \delta(\mathbf{D})$

$\mathbf{S} \leftarrow \emptyset$

for each $b \in \mathbf{B}$ **do**

if $\exists i \notin b$: $\text{is\_encroach}(\{\mathbf{D}[b[1]], \mathbf{D}[b[2]]\}, \mathbf{D}[i], \varepsilon) = \text{true}$ **then**

$\mathbf{S} \leftarrow \mathbf{S} \cup b$

end if

end for

return $\mathbf{S}$

У тривимірному випадку результат алгоритму *encroached* повинен включати ребра та грані, які необхідно розбити. У цьому випадку алгоритм прийме такий вигляд:

algorithm *encroached*

input:

$\mathbf{D}$ – тривимірна триангуляція форми об'єкта Ω

ε – чисельний нуль

output:

$\mathbf{S}$ – множина межових ребер і граней, які потрібно розбити

```

begin
   $\mathbf{B} \leftarrow \delta(\mathbf{D})$ 
   $\mathbf{S} \leftarrow \emptyset$ 
  for each  $b \in \mathbf{B}$  do
    if  $\exists i \notin b: \text{is\_encroach}(\{\mathbf{D}[b[1]], \mathbf{D}[b[2]]\}, \mathbf{D}[i], \varepsilon) = \text{true}$  then
       $\mathbf{S} \leftarrow \mathbf{S} \cup (b[1], b[2])$ 
    else if  $\exists i \notin b: \text{is\_encroach}(\{\mathbf{D}[b[2]], \mathbf{D}[b[3]]\}, \mathbf{D}[i], \varepsilon) = \text{true}$  then
       $\mathbf{S} \leftarrow \mathbf{S} \cup (b[2], b[3])$ 
    else if  $\exists i \notin b: \text{is\_encroach}(\{\mathbf{D}[b[3]], \mathbf{D}[b[1]]\}, \mathbf{D}[i], \varepsilon) = \text{true}$  then
       $\mathbf{S} \leftarrow \mathbf{S} \cup (b[3], b[1])$ 
    end if
  end for
return  $\mathbf{S}$ 

```

Алгоритм перевірки необхідності розбиття `is_encroach` матиме дві реалізації: для ребер і трикутних граней. Для дво- і тривимірних ребер досить перевірити як співвідноситься відстань від центру ребра до точки з його довжиною. У результаті отримаємо такий алгоритм:

algorithm `is_encroach`

input:

$\mathbf{V}$ – список з координат двох вузлів, які належать одному ребру

P – координати точки для перевірки

ε – чисельний нуль

output:

`true`, якщо P потрапить у коло або сферу, діаметром якої є $\mathbf{V}$, `false` – інакше

begin

$C \leftarrow 0,5 (\mathbf{V}[1] + \mathbf{V}[2])$

$r \leftarrow 0,5 \|\mathbf{V}[2] - \mathbf{V}[1]\|$

return $\|P - C\| < r - \varepsilon$

Для трикутних граней перевірку попадання точки в описану екваторіальну сферу можна виконати у місцевих координатах. Для цього необхідно знайти координати вершин трикутника та заданої точки у системі координат, площина xOy якої збігається з площиною елемента, а вісь Oz з його нормаллю. У результаті отримаємо такий алгоритм:

algorithm is_encroach

input:

V – список з координат трьох вузлів, які належать одній грані

P – координати точки для перевірки

ε – чисельний нуль

output:

true, якщо P потрапить у екваторіальну сферу, описану навколо V , false – інакше

begin

$\mathbf{ab} \leftarrow V[2] - V[1]$

$\mathbf{ac} \leftarrow V[3] - V[1]$

$\mathbf{n} \leftarrow \mathbf{ab} \times \mathbf{ac}$

$\mathbf{v}_x \leftarrow \frac{\mathbf{ab}}{\|\mathbf{ab}\|}$

$\mathbf{v}_z \leftarrow \frac{\mathbf{n}}{\|\mathbf{n}\|}$

$\mathbf{v}_y \leftarrow \mathbf{v}_x \times \mathbf{v}_z$

$\lambda \leftarrow \begin{bmatrix} \mathbf{v}_{x,x} & \mathbf{v}_{x,y} & \mathbf{v}_{x,z} \\ \mathbf{v}_{y,x} & \mathbf{v}_{y,y} & \mathbf{v}_{y,z} \\ \mathbf{v}_{z,x} & \mathbf{v}_{z,y} & \mathbf{v}_{z,z} \end{bmatrix}$

$v_1 \leftarrow \lambda (V[1] - V[1])$

$v_2 \leftarrow \lambda (V[2] - V[1])$

$v_3 \leftarrow \lambda (V[3] - V[1])$

$p \leftarrow \lambda (P - V[1])$

$(C, r) \leftarrow \text{circum\_circle}(\{v_1, v_2, v_3\})$

return $\|p - C\| < r - \varepsilon$

Пошук центру описаної екваторіальній сфери для трикутної грані (як в наведений вище процедурі) можна виконати у місцевій системі координат. Для цього спочатку знаходиться центр описаного кола у місцевій системі координат, а потім за допомогою зворотного перетворення координат отримуються його координати у вихідній системі координат. У результаті отримуємо такий алгоритм:

algorithm `circum_circle`

input:

$\mathbf{V}$ – список з координат трьох вузлів, які належать одній грані

output:

C – координати центра описаної навколо $\mathbf{V}$ екваторіальної сфери

r – радіус описаної навколо $\mathbf{V}$ екваторіальної сфери

begin

$\mathbf{ab} \leftarrow \mathbf{V}[2] - \mathbf{V}[1]$

$\mathbf{ac} \leftarrow \mathbf{V}[3] - \mathbf{V}[1]$

$\mathbf{n} \leftarrow \mathbf{ab} \times \mathbf{ac}$

$\mathbf{v}_x \leftarrow \frac{\mathbf{ab}}{\|\mathbf{ab}\|}$

$\mathbf{v}_z \leftarrow \frac{\mathbf{n}}{\|\mathbf{n}\|}$

$\mathbf{v}_y \leftarrow \mathbf{v}_x \times \mathbf{v}_z$

$\lambda \leftarrow \begin{bmatrix} \mathbf{v}_{x,x} & \mathbf{v}_{x,y} & \mathbf{v}_{x,z} \\ \mathbf{v}_{y,x} & \mathbf{v}_{y,y} & \mathbf{v}_{y,z} \\ \mathbf{v}_{z,x} & \mathbf{v}_{z,y} & \mathbf{v}_{z,z} \end{bmatrix}$

$\mathbf{v}_1 \leftarrow \lambda (\mathbf{V}[1] - \mathbf{V}[1])$

$\mathbf{v}_2 \leftarrow \lambda (\mathbf{V}[2] - \mathbf{V}[1])$

$\mathbf{v}_3 \leftarrow \lambda (\mathbf{V}[3] - \mathbf{V}[1])$

$\mathbf{p} \leftarrow \lambda (\mathbf{P} - \mathbf{V}[1])$

$(C, r) \leftarrow \text{circum\_circle}(\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\})$

return $(\mathbf{V}[1] + \lambda^T C, r)$

Алгоритм формування списку «поганих» елементів `poor` ґрунтується на аналізі відношення найбільш довгої сторони елемента до найбільш короткої. Зрозуміло, чим ближче це співвідношення до одиниці, тим ближче елемент за формою до правильного. Цій алгоритм можна записати у загальному вигляді для дискретних моделей форм об'єктів у такому вигляді:

algorithm `poor`

input:

D – дво- або тривимірна триангуляція Делоне форми об'єкта Ω

α – обмеження на відношення довжин сторін елемента

output:

S – множина елементів, які необхідно розбити

begin

$S \leftarrow \emptyset$

for each $e \in D.E$ **do**

$l_{min} \leftarrow \min_{i \in [1; |e|]} \|D[i+1] - D[i]\|$

$l_{max} \leftarrow \max_{i \in [1; |e|]} \|D[i+1] - D[i]\|$

if $\frac{l_{max}}{l_{min}} > \alpha$ **then**

$S \leftarrow S \cup e$

end for

return S

У результаті застосування алгоритму покращення дво- або тривимірних триангуляцій будуть усунені елементи, у яких співвідношення довжин сторін буде не нижче заданого (якщо задане значення вище теоретичної оцінки). Наприклад, на рис. 5.3, а зображено результат покращення триангуляції Делоне для пластинки (рис. 5.2), якщо $\alpha = 2$, а на рис. 5.3, б – переріз площиною $x = 0$ результату покращення триангуляції для моделі об'єкта, який заданий формулою (4.1).

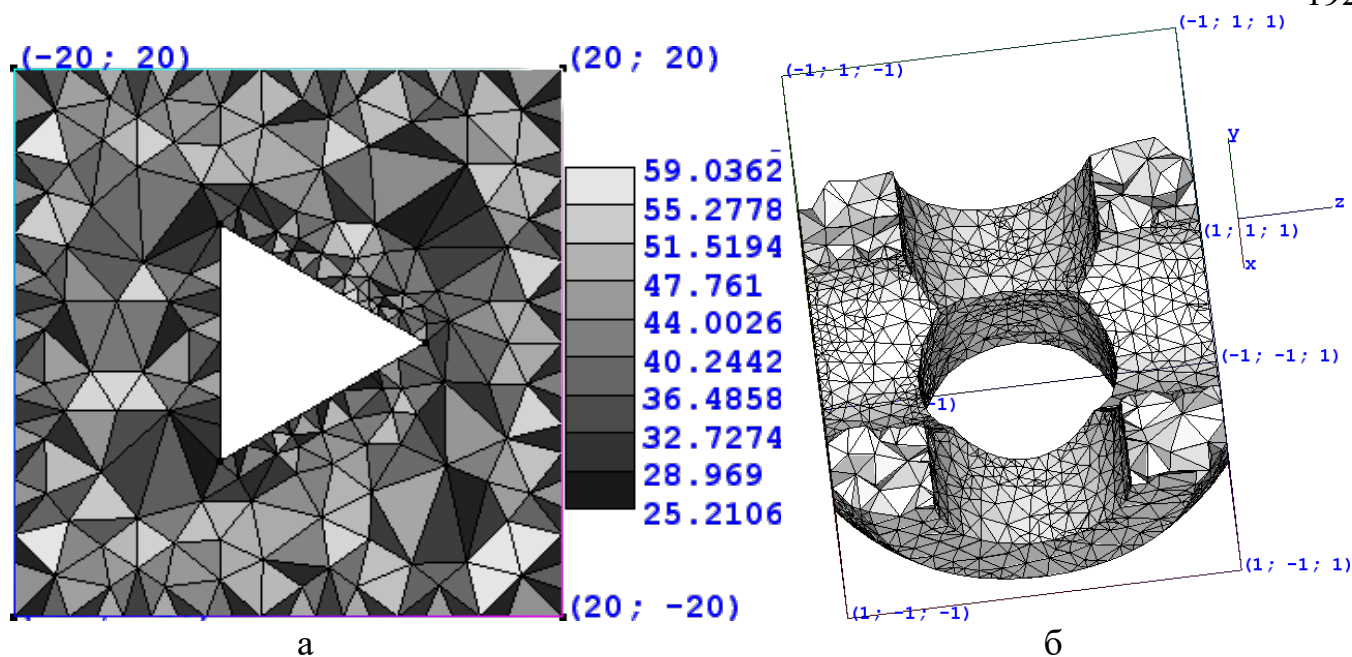


Рисунок 5.3 – Покращення триангуляції Делоне:

а – двовимірна триангуляція

б – тривимірна триангуляція

Таким чином, запропоновані узагальнені методи дозволяють будувати дискретні моделі на базі дво- або тривимірних симплексів, що задовольняють критерію Делоне. У основі цих алгоритмів ідея вставки нових вузлів з одночасною перевіркою виконання умови Делоне. Проте, техніка вставки нового вузла практично не може бути застосована для дискретних моделей форм об'єктів на базі чотирикутників або шестигранників.

5.2 Метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально

Проекційний метод побудови дискретних моделей меж об'єктів (підрозділ 4.1) можна використати як основу при розробці метода генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально. В алгоритмі `background_grid_surface` з фонові дискретної моделі видаляються всі зовнішні

вузли, а також вузли у яких існує неоднозначна інтерпретація напрямку нормалі. Потім генерується модель межі шляхом обчислення проєкцій межових граней елементів, що залишились, на межу об'єкта. Відповідно, для побудови дискретної моделі всього об'єкта потрібно заповнити елементами (солідами) шар простору, який залишився між елементами фонові дискретної моделі та елементами дискретної моделі межі об'єкта. Водночас необхідно враховувати, що між межовими гранями фонові дискретної моделі й елементами на межі існує взаємно-однозначна відповідність. Як результат можна сформулювати метод генерації дискретних моделей форм об'єктів, які задані функціонально, у вигляді такого алгоритму:

algorithm background-grid-mesh

input:

F – функція, яка подає форму об'єкта Ω

G – фонові дискретна модель

ε – чисельний нуль

output:

M – дискретна модель форми об'єкта Ω

begin

for each $n \in G.N$ **do**

if $F(n.v) < 0$ **then**

 remove_node(G , $G.code(n)$)

end if

end for

$V_0 \leftarrow$ вузли дотику елементів G

while $V_0 \neq \emptyset$ **do**

$T \leftarrow \bigcup_{v \in V_0} G.incident(v)$

$k_{e0} \leftarrow \operatorname{argmin}_{k_e \in T} (G.power(G(k_e)))$

 remove_element(G , k_{e0})

$V_0 \leftarrow$ вузли дотику елементів G

```

end while
 $\mathbf{B} \leftarrow \delta(\mathbf{G})$ 
 $\mathbf{M} \leftarrow \mathbf{G}$ 
 $h \leftarrow \min_{(n_0 \neq n_1) \in \mathbf{M.N}} \|n_0.v - n_1.v\|$ 
for each  $e \in \mathbf{B.E}$  do
     $\mathbf{V} \leftarrow \emptyset$ 
    for  $i \leftarrow |e|$  down-to 1 do
         $\mathbf{V} \leftarrow \mathbf{B.N}[e(i)]$ 
    end for
    for  $i \leftarrow 1$  to  $|e|$  do
         $\mathbf{V} \leftarrow \mathbf{V} \cup \{\text{find\_border}(\mathbf{V}[i], \mathbf{F}, h, \varepsilon)\}$ 
    end for
    add_elements( $\mathbf{M}, \mathbf{V}, \varepsilon$ )
end for
return  $\mathbf{M}$ 

```

На першому етапі алгоритму формується початкова дискретна модель шляхом видалення зовнішніх вузлів із фонової. Далі для усунення точок дотику з початкової моделі видаляються елементи, які суміжні у відповідних вузлах. Для кожної межевої грані початкової дискретної моделі обчислюється її проекція на межі об'єкта Ω . На заключному етапі формуються межеві елементи. Для цього вузли межевих граней та їх проекції об'єднуються у множину $\mathbf{V}$, для якої будуються елементи за допомогою алгоритму add_elements. Множина $\mathbf{V}$ будується з урахуванням того, що порядок обходу вершин у спільній грани сусідніх елементів протилежний. Якщо елементами фонові дискретної моделі є чотирикутники або шестигранники, то для кожної межевої грані формується тільки один елемент. У такому випадку алгоритм add_elements отримає такий вигляд:

```

algorithm add_elements
    input:

```

$\mathbf{M}$ – дискретна модель на базі двовимірних чотирикутників або шестигранників

$\mathbf{V}$ – список вершин

ε – чисельний нуль

begin

insert_element($\mathbf{M}$, $\mathbf{V}$, ε)

end

Якщо елементами фонові дискретної моделі є трикутники, то для кожного межового ребра початкової сітки можна двома способами побудувати елементи (змінюючи напрям діагоналі у чотирикутнику, який утворено суміжними межевими вершинами та їх проекціями). З двох можливих способів побудови трикутників у межевому шарі (рис. 5.4) раціонально обирати той, при якому сума довжин сторін буде мінімальною. Якщо вважати, що вершини збережені за порядком обходу супроти годинникової стрілки, то отримаємо такий алгоритм:

algorithm add_elements

input:

$\mathbf{M}$ – дискретна модель на базі двовимірних трикутників

$\mathbf{V}$ – список вершин

ε – чисельний нуль

begin

if $\|\mathbf{V}[3] - \mathbf{V}[1]\| < \|\mathbf{V}[4] - \mathbf{V}[2]\|$ **then**

insert_element($\mathbf{M}$, $\{\mathbf{V}[1], \mathbf{V}[2], \mathbf{V}[3]\}$, ε)

insert_element($\mathbf{M}$, $\{\mathbf{V}[1], \mathbf{V}[3], \mathbf{V}[4]\}$, ε)

else

insert_element($\mathbf{M}$, $\{\mathbf{V}[1], \mathbf{V}[2], \mathbf{V}[4]\}$, ε)

insert_element($\mathbf{M}$, $\{\mathbf{V}[2], \mathbf{V}[3], \mathbf{V}[4]\}$, ε)

end if

end

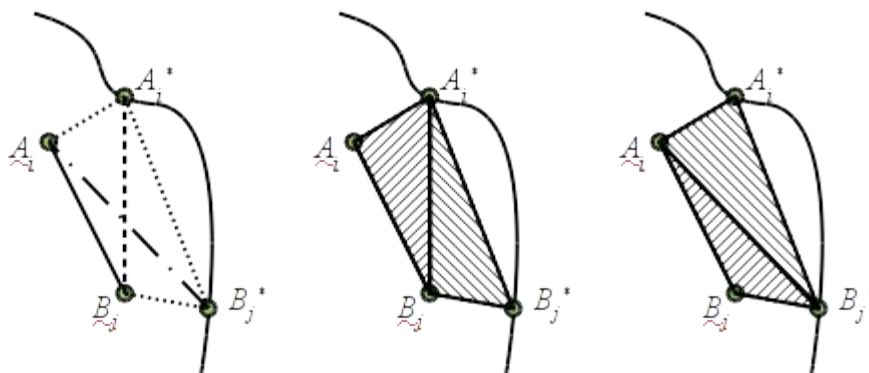


Рисунок 5.4 – Способи побудови трикутників у межевому шарі

У випадку, якщо елементами фонові дискретної моделі є тетраедри, при розбитті призми, яка утворена трикутною гранню та її проекцією, необхідно забезпечити топологічну коректність ребер та граней. Тобто, для кожної трійки вузлів, які утворюють внутрішню грань у одному елементі, повинна існувати грань з цих вузлів у іншому елементі.

Припустимо, що кодом кожного вузла є його номер у відповідній множині. Водночас будемо вважати, що у призматичних елементах порядок обходу вершин такий, як на рис. 5.5.

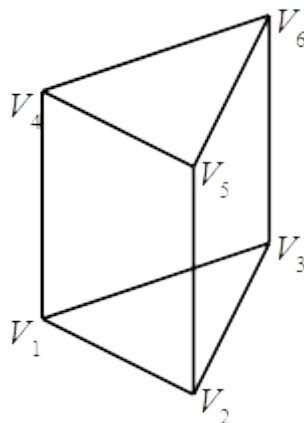


Рисунок 5.5 – Порядок обходу вершин призматичного елемента

Кожна чотирикутна грань призми може бути розбита на трикутники двома способами. У результаті існує вісім способів проведення діагоналей у чотирикутних гранях. Водночас у шести випадках можна розбити призму на три тетраедри (рис. 5.6). А у двох, що залишилися, за необхідне буде додавання нових вузлів [299].

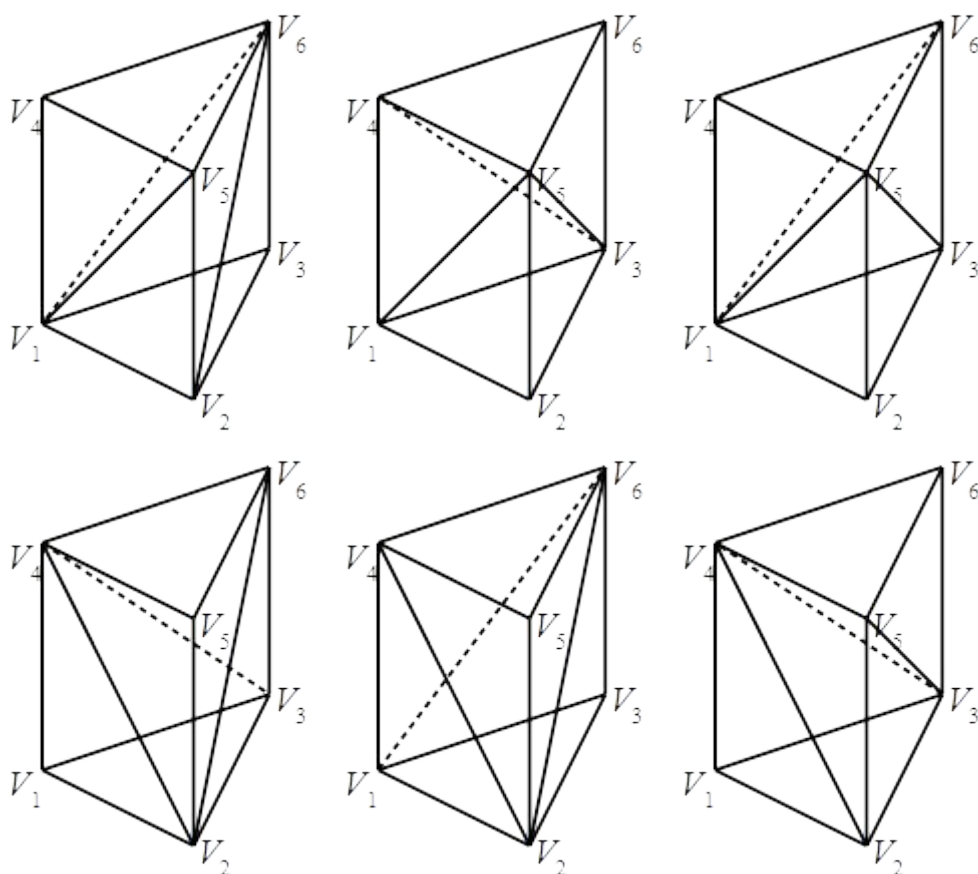


Рисунок 5.6 – Способи розбиття призми на тетраедри

Якщо прийняти вимогу, що діагоналі повинні проходити у елементі через вершину з найменшим кодом вузла, тоді у двох з трьох чотирикутних граней діагоналі будуть визначені. Щоб обрати напрям діагоналі на грані, яка залишилась, також необхідно знайти вершину з мінімальним кодом вузла та провести діагональ через неї [299]. У результаті можна отримати такий алгоритм додавання прикордонних тетраедричних елементів:

algorithm add_elements

input:

M – дискретна модель на базі тетраедрів

V – список вершин

ε – чисельний нуль

begin

I $\leftarrow$ ((1, 2, 3, 4, 5, 6), (2, 3, 1, 5, 6, 4), (3, 1, 2, 6, 4, 5), (4, 6, 5, 1, 3, 2), (5, 4, 6, 2, 1,

3), (6, 5, 4, 3, 2, 1))

$r \leftarrow \underset{i}{\operatorname{argmin}}(\mathbf{V}[i])$

if $\min(\mathbf{V}[\mathbf{I}[r, 2]], \mathbf{V}[\mathbf{I}[r, 6]]) < \min(\mathbf{V}[\mathbf{I}[r, 3]], \mathbf{V}[\mathbf{I}[r, 5]])$ **then**

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 2]], \mathbf{V}[\mathbf{I}[r, 3]], \mathbf{V}[\mathbf{I}[r, 6]]\}$, ϵ)

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 2]], \mathbf{V}[\mathbf{I}[r, 6]], \mathbf{V}[\mathbf{I}[r, 5]]\}$, ϵ)

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 5]], \mathbf{V}[\mathbf{I}[r, 6]], \mathbf{V}[\mathbf{I}[r, 4]]\}$, ϵ)

else

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 2]], \mathbf{V}[\mathbf{I}[r, 3]], \mathbf{V}[\mathbf{I}[r, 5]]\}$, ϵ)

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 5]], \mathbf{V}[\mathbf{I}[r, 3]], \mathbf{V}[\mathbf{I}[r, 6]]\}$, ϵ)

 insert_element($\mathbf{M}$, $\{\mathbf{V}[\mathbf{I}[r, 1]], \mathbf{V}[\mathbf{I}[r, 5]], \mathbf{V}[\mathbf{I}[r, 6]], \mathbf{V}[\mathbf{I}[r, 4]]\}$, ϵ)

end if

end

Отже, розроблено метод генерації дискретних моделей форм об'єктів на базі елементів типу солід. Наприклад, для 1-області (рис. 2.4), яка задана формулою (2.13), якщо використовувати як фонову рівномірну дискретну модель з 8×11 вузлів для прямокутника $[0; 30] \times [0; 40]$, то результати послідовного застосування алгоритмів background-grid-mesh і boundary_refinement (8 ітерацій згладжування дискретної моделі межі об'єкта) матимуть вигляд, наведений на рис. 5.7, а при використанні трикутників або вигляд, наведений на рис. 5.7, б, при використанні чотирикутників. В отриманих дискретних моделях мінімальний кут у елементі $11,7^\circ$ для трикутників і $19,8^\circ$ для чотирикутників (відповідні розподіли показані півтоном на рис. 5.7). Водночас у дискретній моделі на базі трикутників максимальний кут у діапазоні від $60,7^\circ$ до $155,5^\circ$, а відношення мінімальної до максимальної довжин сторін елемента – від 0,34 до 0,99. У дискретній моделі на базі чотирикутників максимальний кут змінюється від 90° до $160,9^\circ$, а відношення мінімальної до максимальної довжин сторін – від 0,4 до 0,93.

Аналіз значень мінімального кута ($\alpha_{\min}$), максимального кута ($\alpha_{\max}$) і мінімального значення відношення мінімальної до максимальної довжини сторін елемента ($\min(l_{\min} / l_{\max})$) при різній кількості елементів у фоновій дискретній моделі

показує, що можливим є отримання елементів з малими кутами як на базі трикутників, так і на базі чотирикутників (табл. 5.1).

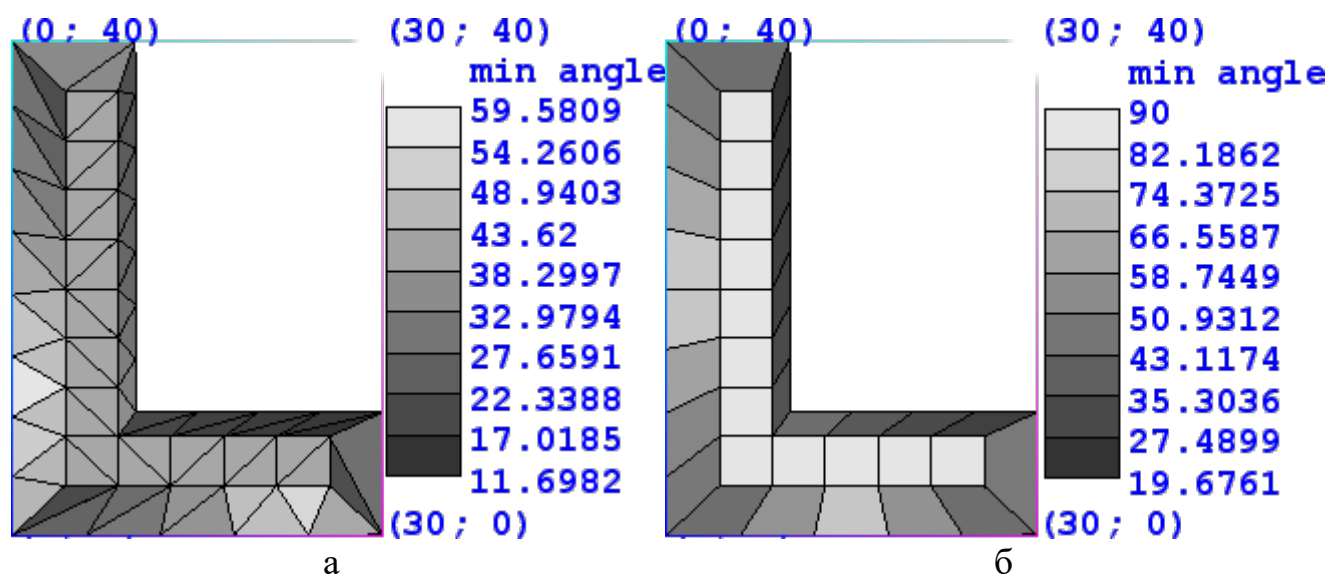


Рисунок 5.7 – Результати застосування узагальненого методу генерації дискретних моделей форм об’єктів на базі двовимірних елементів типу солід:

а – дискретна модель L-області на базі трикутників

б – дискретна модель L-області на базі чотирикутників

Таблиця 5.1 – Розподіл характеристик дискретних моделей L-області

Фонові дискретні моделі	8×11	9×12	10×13	11×14	11×15	12×16	13×17	14×18	14×19
Трикутні елементи									
Кількість елементів	76	88	100	144	152	168	184	242	252
$\alpha_{\min}, ^\circ$	11,7	15,9	17,7	6,9	11,9	16,0	17,5	6,9	12,3
$\alpha_{\max}, ^\circ$	155,5	144,4	135,9	166,0	154,9	144,0	135,2	166,0	154,2
$\min(l_{\min} / l_{\max})$	0,34	0,44	0,44	0,29	0,33	0,45	0,43	0,31	0,33
Чотирикутні елементи									
Кількість елементів	38	44	50	72	76	84	92	121	126
$\alpha_{\min}, ^\circ$	19,8	34,7	44,9	14,4	19,3	34,4	44,9	14,3	19,1
$\alpha_{\max}, ^\circ$	160,9	145,9	135,3	166,0	161,1	146,0	135,2	166,0	161,2
$\min(l_{\min} / l_{\max})$	0,4	0,37	0,34	0,38	0,5	0,54	0,50	0,34	0,44

При використанні чотирикутних (та шестигранних) фонових дискретних моделей можлива поява вироджених елементів (у яких є кути $\geq 180^\circ$). Наприклад, якщо використовувати як фонову рівномірну дискретну модель з 10×10 квадратних елементів для квадратної області $[-20; 20] \times [-20; 20]$, при побудові моделі пластинки, яка задана формулою (2.14), буде отримано три вироджених елементи (рис. 5.8).

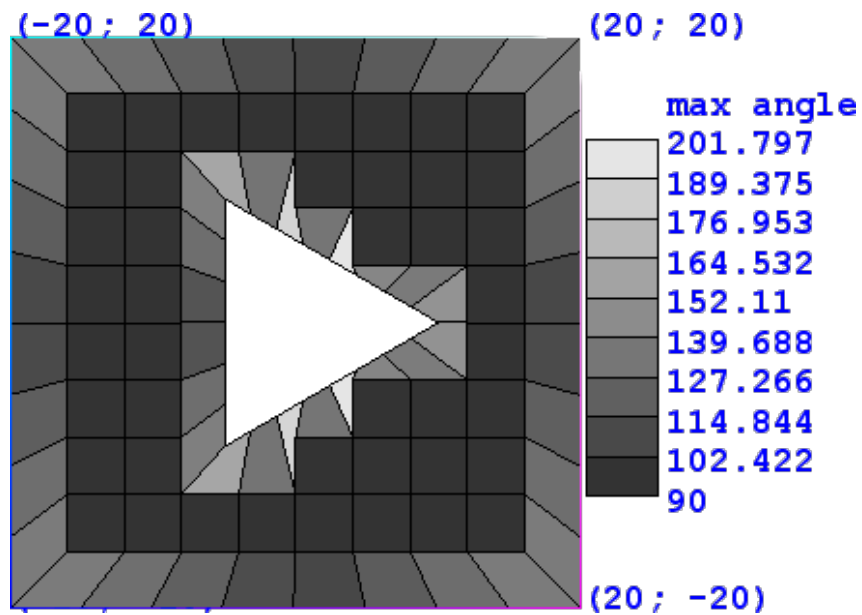


Рисунок 5.8 – Дискретна модель пластинки

У тривимірному випадку результати аналогічні двовимірним. Наприклад, для області, яка задана формулою (4.1), при використанні рівномірної дискретної моделі на базі тетраедрів ($21 \times 21 \times 21$ вузол) для куба $[-1; 1] \times [-1; 1] \times [-1; 1]$ мінімальний кут в елементі змінюється від $0,62^\circ$ до $50,1^\circ$, максимальний кут – від $68,9^\circ$ до $177,2^\circ$, відношення мінімальної до максимальної довжин ребер елемента – від 0,067 до 0,81 (рис. 5.9, а). Якщо використати кубічні фонові елементи, то мінімальний кут буде у діапазоні від $7,48^\circ$ до 90° , максимальний кут – від $68,9^\circ$ до $177,2^\circ$ й відношення мінімальної до максимальної довжин ребер елемента – від 0,12 до 1 (рис. 5.9, б).

В табл. 5.2 зіставляються значення мінімального кута ($\alpha_{\min}$), максимального кута ($\alpha_{\max}$) і мінімального значення відношення мінімальної до максимальної довжин сторін елемента ($\min(l_{\min} / l_{\max})$) при різній кількості елементів у фонових дискретних моделях на базі тетраедрів і шестигранників.

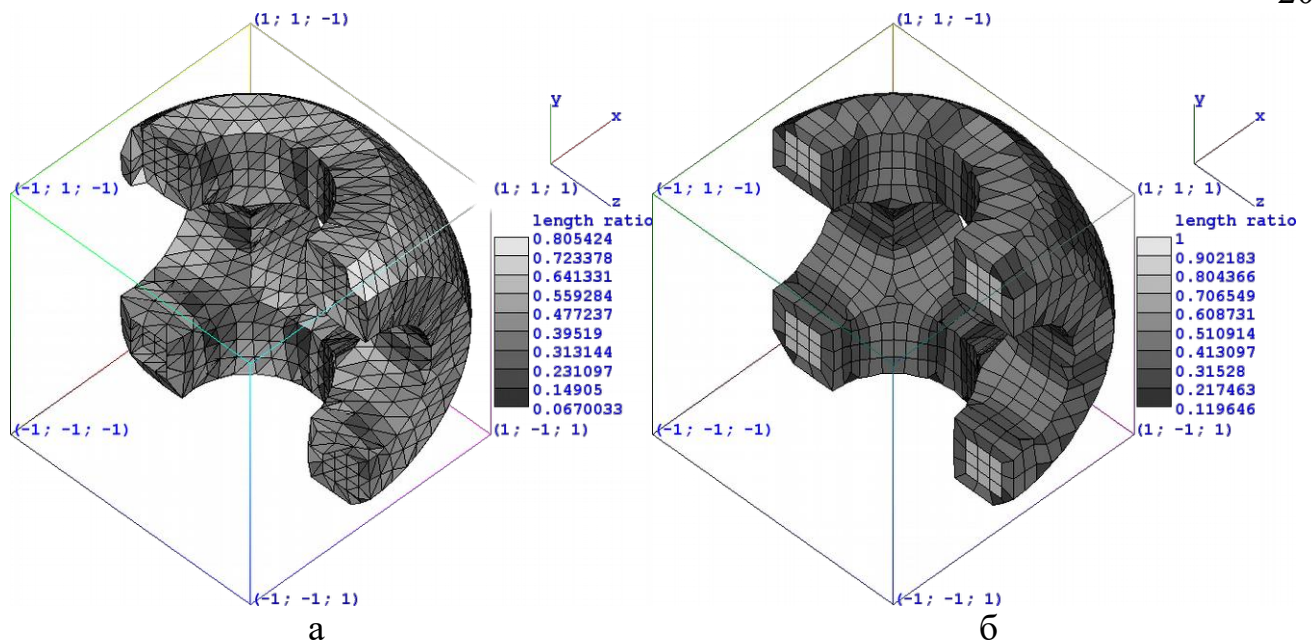


Рисунок 5.9 – Результати застосування узагальненого методу генерації дискретних моделей форм об’єктів на базі тривимірних елементів типу солід:

а – дискретна модель форми об’єкта на базі тетраедрів

б – дискретна модель форми об’єкта на базі шестигранників

Таблиця 5.2 – Розподіл характеристик дискретних моделей форми об’єкта, заданої формулою (4.1)

Фонові дискретна модель	21×21	22×22	23×23	24×24	25×25	26×26	27×27	28×28	29×29
Тетраедричні елементи									
Кількість елементів	25944	30048	34704	38364	45372	51798	56328	62718	73308
$\alpha_{\min}, ^\circ$	0,6	1,2	0,5	0,5	0,1	0,3	0,7	0,8	0,5
$\alpha_{\max}, ^\circ$	177,2	177,2	178,5	178,7	179,8	179,2	177,8	177,8	177,9
$\min(l_{\min} / l_{\max})$	0,06	0,1	0,01	0,03	0,02	0,06	0,06	0,1	0,04
Шестигранні елементи									
Кількість елементів	4416	5112	5912	6536	7736	8816	9584	10688	12496
$\alpha_{\min}, ^\circ$	7,5	14,1	8,8	1,2	2,2	5,7	6,5	12,5	2,3
$\alpha_{\max}, ^\circ$	177,2	177,6	179,8	179,7	179,3	179,5	176,9	178,5	178,1
$\min(l_{\min} / l_{\max})$	0,12	0,09	0,08	0,07	0,05	0,09	0,07	0,12	0,06

Так, можна зробити висновки про те, що отримані дискретні моделі потребують подальшої обробки (наприклад, згладжування) для покращення їх чисельних характеристик. Водночас дискретні моделі на базі чотирикутних або шестигранних елементів можуть містити вироджені елементи, для усунення яких необхідно виконати додаткову обробку.

5.3 Згладжування Лапласа для внутрішніх вузлів

Поширеним методом покращення характеристик дискретних моделей є згладжування Лапласа для внутрішніх вузлів. В підрозділі 4.2 викладено його теоретичне обґрунтування. Алгоритмічно при згладжуванні Лапласа кожен внутрішній вузол дискретної моделі форму об'єкта переноситься у позицію середнього арифметичного сусідніх вузлів. У результаті можна записати такий алгоритм:

algorithm laplacian-smoothing

input:

M – дискретна модель форми об'єкта

iter – кількість ітерацій

begin

for $i \leftarrow 1$ **to** *iter* **do**

for each $n \in A.N$ **do**

if $n \notin \delta(M)$ **then**

$A \leftarrow M.\text{adjacent}(M.\text{code}(n))$

$$n.v \leftarrow \frac{1}{|A|} \sum_{k \in A} M[k]$$

end if

end for

end for

end

На рис. 5.10, а–г показані результати, відповідно, однієї, двох, трьох і чотирьох ітерацій згладжування Лапласа для дискретної моделі L-області на базі трикутників (рис. 5.7, а). Водночас збільшення числа ітерацій згладжування Лапласа після чотирьох ітерацій слабо впливає на основні характеристики дискретної моделі форми об'єкта (табл. 5.3).

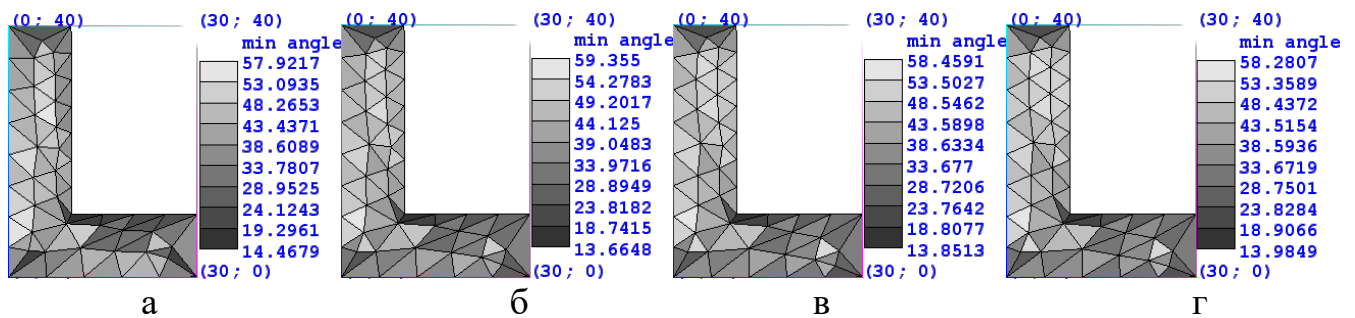


Рисунок 5.10 – Згладжування Лапласа дискретної моделі на базі трикутників:

а – 1 ітерація

б – 2 ітерації

в – 3 ітерації

г – 4 ітерації

Таблиця 5.3 – Розподіл характеристик дискретної моделі L-області на базі трикутних елементів

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	11,7	14,5	13,7	13,9	14,0	14,1	14,2	14,2	14,2	14,2
$\alpha_{\max}, ^\circ$	155,5	129,6	134,6	136,5	137,2	137,5	137,6	137,6	137,6	137,7
$\min(l_{\min} / l_{\max})$	0,34	0,28	0,29	0,29	0,29	0,29	0,29	0,29	0,29	0,29

На рис. 5.11, а–г показані результати, відповідно, однієї, двох, трьох і чотирьох ітерацій згладжування Лапласа для дискретної моделі L-області на базі чотирикутників (рис. 5.7, б). Водночас (як й у випадку з трикутниками) збільшення числа ітерацій згладжування Лапласа після чотирьох ітерацій слабо впливає на основні характеристики дискретної моделі форми об'єкта (табл. 5.4).

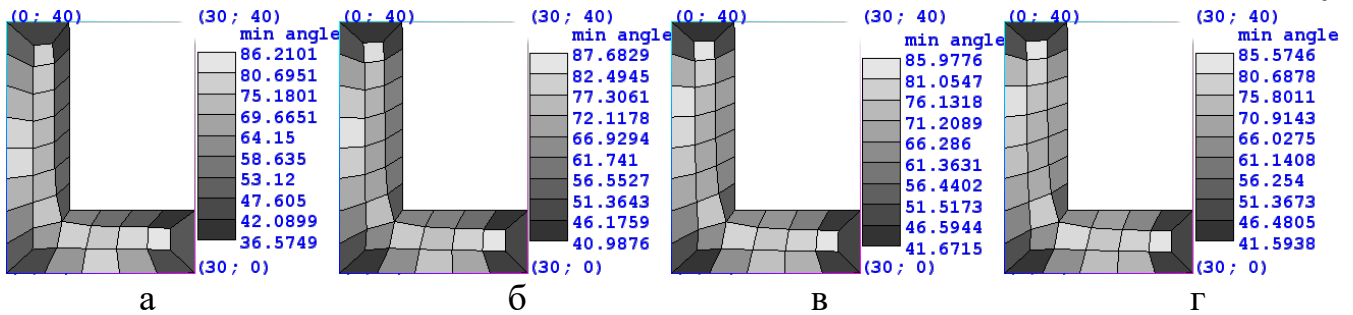


Рисунок 5.11 – Згладжування Лапласа дискретної моделі на базі чотирикутників:

а – 1 ітерація

б – 2 ітерації

в – 3 ітерації

г – 4 ітерації

Таблиця 5.4 – Розподіл характеристик дискретної моделі L-області на базі чотирикутних елементів

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	19,8	36,6	41,0	41,7	41,6	41,5	41,2	41	40,9	40,9
$\alpha_{\max}, ^\circ$	160,9	146,4	144,3	143,4	143,2	143,3	143,3	143,3	143,2	143,2
$\min(l_{\min} / l_{\max})$	0,4	0,3	0,32	0,33	0,33	0,33	0,33	0,33	0,33	0,33

У тривимірному випадку результати застосування згладжування Лапласа є аналогічними двовимірним. На рис. 5.12, а–г показані результати, відповідно, однієї, двох, трьох і чотирьох ітерацій згладжування Лапласа для дискретної моделі на базі тетраедрів, яка зображена на рис. 5.9, а.

При використанні більш, ніж чотирьох ітерацій згладжування Лапласа для дискретних моделей на базі тетраедрів характеристики мінімального і максимального кута, відношення довжин сторін змінюються слабо (табл. 5.5). Водночас характеристики дискретної моделі істотно залежать від відповідних значень для елементів на межі. Якщо застосувати, наприклад, чотири ітерації згладжування мезових вузлів разом зі згладжуванням внутрішніх вузлів, то мінімальне значення для відношення довжин сторін буде рівним 0,23 (рис. 5.13).

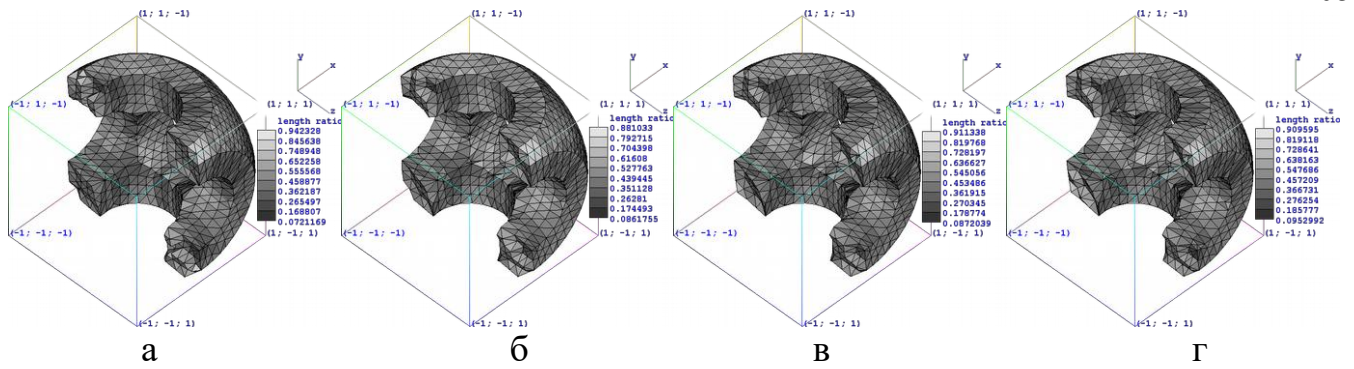


Рисунок 5.12 – Згладжування Лапласа дискретної моделі на базі тетраедрів:

а – 1 ітерація

б – 2 ітерації

в – 3 ітерації

г – 4 ітерації

Таблиця 5.5 – Розподіл значень характеристик дискретної моделі форми об'єкта на базі тетраедричних елементів

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	0,6	3,4	3,3	4,0	3,6	3,7	3,4	3,2	3,5	3,2
$\alpha_{\max}, ^\circ$	177,2	161,3	169,1	168,3	169,7	169,2	170,2	170,7	170,4	170,7
$\min(l_{\min} / l_{\max})$	0,06	0,07	0,09	0,09	0,11	0,11	0,11	0,11	0,11	0,12

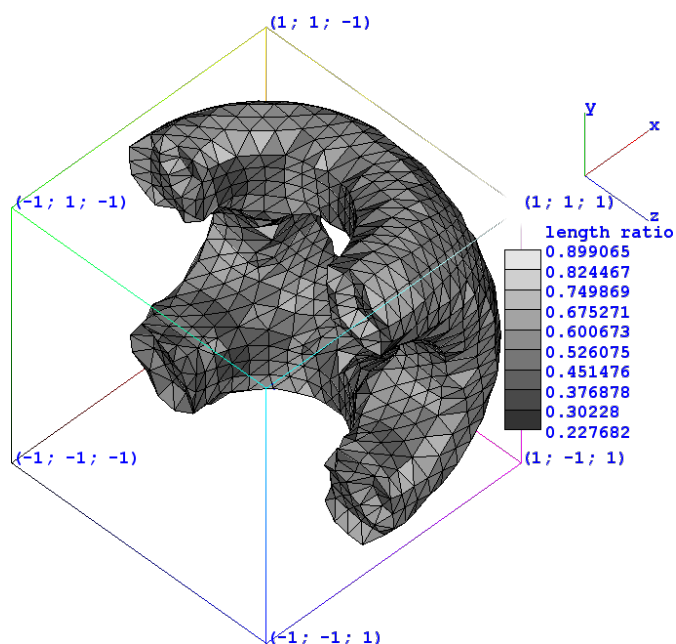


Рисунок 5.13 – Результат одночасного згладжування межових і внутрішніх вузлів

На рис. 5.14, а–г показані результати, відповідно, однієї, двох, трьох і чотирьох ітерацій згладжування Лапласа для дискретної моделі на базі шестигранників, зображеної на рис. 5.9, б. Як і в разі тетраедрів, при використанні більш чотирьох ітерацій згладжування Лапласа для дискретних моделей на базі шестигранників характеристики мінімального і максимального кута, відношення довжин сторін змінюються слабо (табл. 5.6): починаючи з п'ятої ітерації мінімальний кут знаходиться в інтервалі $23\text{--}24^\circ$, а мінімальне співвідношення довжин сторін в елементі в межах $0,12\text{--}0,13$.

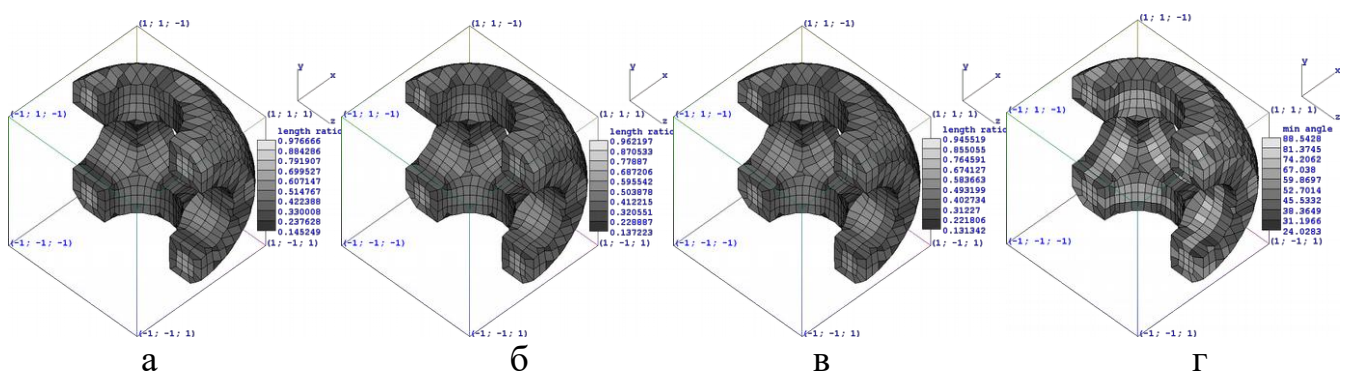


Рисунок 5.14 – Згладжування Лапласа дискретної моделі на базі шестигранників:

а – 1 ітерація

б – 2 ітерації

в – 3 ітерації

г – 4 ітерації

Таблиця 5.6 – Розподіл значень характеристик дискретної моделі форми об'єкта на базі шестигранних елементів

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	7,5	15,8	24,0	24,2	24,0	23,8	23,7	23,5	23,4	23,3
$\alpha_{\max}, ^\circ$	177,2	178,4	178,4	178,4	179,5	179,2	179,1	179,0	179,0	178,9
$\min(l_{\min} / l_{\max})$	0,12	0,15	0,15	0,15	0,13	0,13	0,12	0,12	0,12	0,12

Як і для тетраедрів, характеристики дискретної моделі на базі шестигранників залежать від положень вузлів на межі. Якщо застосувати чотири ітерації

згладжування і внутрішніх, і межових вузлів, тоді мінімальне значення для відношення довжин сторін буде приблизно рівним 0,21 (рис. 5.15).

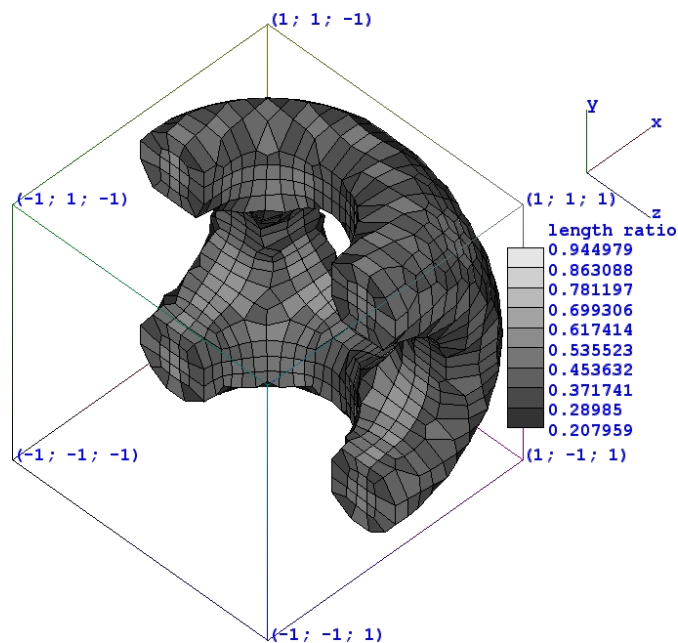


Рисунок 5.15 – Результат одночасного згладжування межових і внутрішніх вузлів дискретної моделі форми об’єкта на базі шестигранників

5.4 Метод згладжування координат внутрішніх вузлів дискретних моделей форм об’єктів

Практичне застосування згладжування Лапласа зумовлює поступове вирівнювання довжин внутрішніх ребер елементів. Проте, переміщення вузлів в координати геометричних центрів опуклих фігур, які утворюються сусідніми вузлами, може не тільки не покращувати характеристики дискретної моделі форми об’єкта, але й спричинити появу вироджених (у разі використання чотирикутних або шестигранних елементів). У такому випадку можна використовувати згладжування, засноване на мінімізації деякої цільової функції, визначеної на вузлах дискретної моделі форми об’єкта.

Розглянемо плоский чотирикутний елемент. Аплікати векторних добутків пар векторів, направлених з кожної вершини уздовж суміжних у ній ребер (рис. 5.16),

дорівнюють подвоєним площам трикутників, визначених вершиною, суміжними ребрами і діагоналлю. Причому знак аплікати буде додатним, якщо кут між векторами набуває значення більше 0 і менше π ; від'ємним, якщо кут між векторами від π до 2π . Апліката дорівнюватиме нулю, якщо вектори лежать на одній прямій. Отже, у неопуклого чотирикутника хоча б в одному вузлі такий добуток буде меншим або дорівнювати нулю.

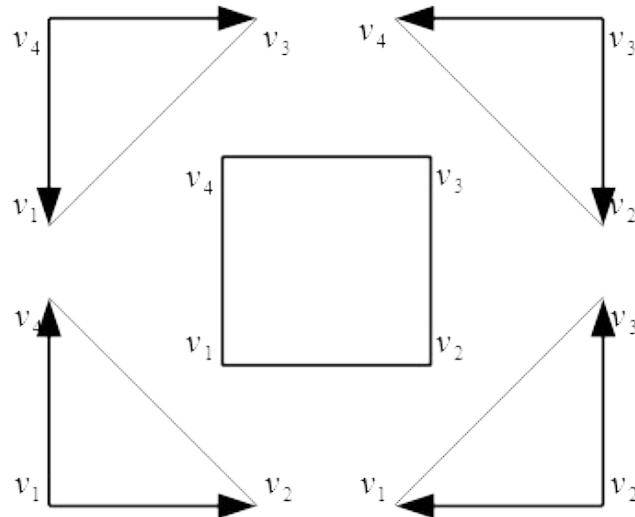


Рисунок 5.16 – Подання чотирикутника чотирма трикутниками

Для кожного елемента дискретної моделі $\mathbf{M}$, код якого k_e ($n_e = |\mathbf{M}(k_e)|$ – кількість вершин у елементі), можна визначити функціонал вигляду

$$L(\mathbf{M}, k_e) = \sum_{j=1}^{n_e} e^{-t\alpha(\mathbf{M}(k_e, j-1), \mathbf{M}(k_e, j), \mathbf{M}(k_e, j+1))}, \quad (5.1)$$

де α – площа трикутника зі знаком, яка визначена такою формулою:

$$\begin{aligned} \alpha(\mathbf{M}(k_e, j-1), \mathbf{M}(k_e, j), \mathbf{M}(k_e, j+1)) = \\ = (\mathbf{M}(k_e, j-1) - \mathbf{M}(k_e, j)) \times (\mathbf{M}(k_e, j+1) - \mathbf{M}(k_e, j)). \end{aligned} \quad (5.2)$$

Доданки у формулі (5.1) будуть приймати значення від 0 до 1, якщо відповідний кут більший 0 і менший π , дорівнювати 1, якщо кут, рівний 0 або π , і є більшими 1, якщо кут від π до 2π . У результаті функціонал, визначений на множині

елементів $\mathbf{E}$ дискретної моделі $\mathbf{M}$, мінімізація якого дозволить усунути елементи з кутом, більшим або рівним π , матиме такий вигляд ($\mathbf{M}.\text{code}(e)$ – код елемента e):

$$F(\mathbf{M}) = \sum_{e \in \mathbf{M} \cdot \mathbf{E}} L(\mathbf{M}, \mathbf{M}.\text{code}(e)). \quad (5.3)$$

Значення параметру t обирається з врахуванням розмірів об'єкту та кількості елементів у дискретній моделі. Можна вважати, що

$$t = \frac{1}{\bar{\alpha}},$$

де $\bar{\alpha}$ – середнє значення функції (5.2) для елементів дискретної моделі.

Мінімізація функціоналу (5.3) для всіх вузлів сітки одночасно є вельми екстенсивною процедурою, тому що кількість незалежних змінних буде в два рази більше кількості вузлів у сітці. Зменшити розмірність розв'язуваної задачі можна за допомогою послідовних локальних рішень для кожного вузла.

Для кожного вузла з кодом k_i можна визначити функціонал вигляду

$$W(\mathbf{M}, k_i) = \sum_{k_e \in \mathbf{I}_k} L(\mathbf{M}, k_e), \quad (5.4)$$

де $\mathbf{I}_k = \mathbf{M}.\text{incident}(k_i)$ – множина елементів, інцидентних у вузлі з кодом k_i .

Функціонал (5.4) містить дві незалежні змінні (координати вузла), що суттєво спрощує пошук мінімуму. Послідовну мінімізацію функціоналів (5.4) можна розглядати як ітераційне наближення до мінімуму функціоналу (5.3) (наприклад, за допомогою градієнтного спуску). За аналогією зі згладжуванням Лапласа для досягнення найкращого результату дану процедуру необхідно повторювати кілька разів. У результаті отримано такий алгоритм:

algorithm local-functional-smoothing

input:

$\mathbf{M}$ – дискретна модель форми об’єкта

W – цільова функція

$iter$ – кількість ітерацій

begin

for $i \leftarrow 1$ **to** $iter$ **do**

for each $n \in \mathbf{M.N}$ **do**

if $n \notin \delta(\mathbf{M})$ **then**

$n.v \leftarrow \underset{n.v}{\operatorname{argmin}} W(\mathbf{M}, \mathbf{M.code}(n))$

end if

end for

end for

end

Розглянемо роботу описаного вище алгоритму на прикладі дискретної моделі квадратної пластинки з трикутним вирізом, форма якої задана формулою (2.14). Вихідна модель (рис. 5.17, а), отримана за допомогою алгоритму background-grid-mesh при використанні рівномірного фоновієї сітки з 15×15 квадратних елементів, містить вироджені елементи біля вершин вирізу. У результаті згладжування Лапласа (4 ітерації, рис. 5.17, б) загальне розташування вузлів стало краще, але залишилися три неопуклих елемента (близько вершин вирізу). Після однієї ітерації запропонованим алгоритмом (рис. 5.17, в) модель вже не містить неопуклих елементів, але близько отвору є елемент з кутом, близьким до 180° . На рис. 5.17, г, д, є і ж, відповідно, показані результати застосування 2, 4, 8 і 16 ітерацій по вузлам з локальною мінімізацією. Можна побачити, що вже на 8 ітераціях ребро біля правого кута вирізу практично збігається з його бісектрисою.

Для вихідної дискретної моделі функціонал (5.3) дорівнює 58,217. У результаті мінімізації (5.3) за допомогою методу градієнтного спуску в глобальній постановці (одночасно для всіх внутрішніх вузлів) отримана дискретна модель, для якої функціонал (5.3) дорівнює 1,451 (рис. 5.17, з). Мінімальний кут в елементах

дискретної моделі буде в межах від $27,3^\circ$ до $89,6^\circ$, максимальний кут – від $90,7^\circ$ до $151,6^\circ$, а відношення довжин сторін – від 0,33 до 0,98.

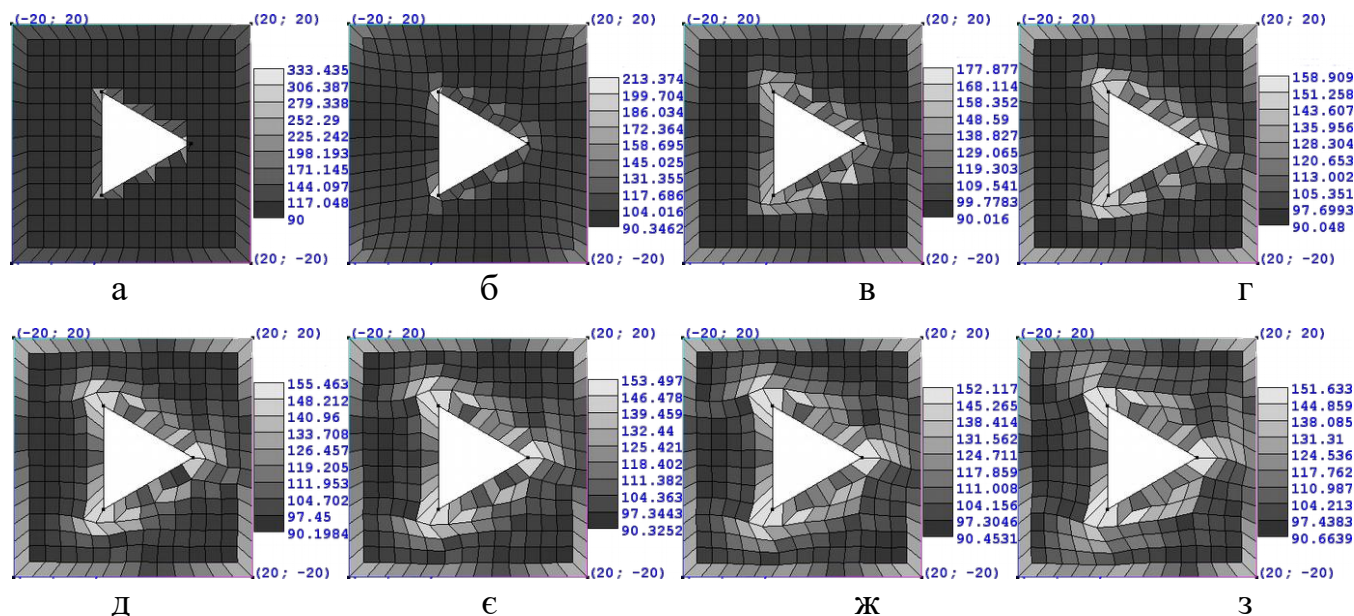


Рисунок 5.17 – Згладжування внутрішніх вузлів дискретної моделі пластинки на базі чотирикутних елементів:

а – вихідна дискретна модель пластинки

б – 4 ітерації згладжування Лапласа

в – 1 ітерація згладжування на основі локальної мінімізації функціоналу

г – 2 ітерації згладжування на основі локальної мінімізації функціоналу

д – 4 ітерації згладжування на основі локальної мінімізації функціоналу

е – 8 ітерацій згладжування на основі локальної мінімізації функціоналу

ж – 16 ітерацій згладжування на основі локальної мінімізації функціоналу

з – результат глобальної мінімізації функціоналу (5.3)

Необхідно відзначити, що зовнішній кут біля вершин трикутного отвору ділиться на два одним ребром, отже, мінімально можливою оцінкою для характеристики максимального кута буде 150° . У табл. 5.7 порівнюються значення функціоналу (5.3), а також характеристики мінімального і максимального кутів, відношення довжин сторін у залежності від кількості ітерацій згладжування внутрішніх вузлів дискретної моделі.

Таблиця 5.7 – Розподіл характеристик у дискретній моделі пластинки

Кількість ітерацій	1	2	3	4	5	6	7	8	9	10
$F(\mathbf{M})$	5,423	2,796	2,233	1,965	1,809	1,710	1,643	1,598	1,565	1,542
$\alpha_{\min}, ^\circ$	33,7	34,1	33,0	32,2	31,5	30,9	30,4	29,9	29,6	29,3
$\alpha_{\max}, ^\circ$	177,9	158,9	156,6	155,5	154,7	154,2	153,8	153,5	153,2	153,0
$\min(l_{\min} / l_{\max})$	0,28	0,37	0,41	0,40	0,38	0,37	0,37	0,36	0,36	0,35

При використанні фонової дискретної моделі на базі 50×50 квадратних елементів для моделі фрикційного диска трансмісії, яка задана формулою (2.16), результат буде містити вироджені чотирикутники (рис. 5.18, а). Після чотирьох ітерацій згладжування все ще залишаються з максимальним кутом більшим 180° (рис. 5.18, б). У результаті шістнадцяти ітерацій згладжування (рис. 5.18, в) вироджених елементів не буде (максимальний кут елемента знаходиться у діапазоні від $90,16^\circ$ до $167,38^\circ$). Водночас у результаті мінімізації (5.3) одночасно для всіх внутрішніх вузлів результати будуть близькими до 16 ітерацій локальної мінімізації (максимальний кут елемента знаходиться у діапазоні від $90,20^\circ$ до $167,25^\circ$).

Згладжування на основі локальної мінімізації можна застосовувати у тривимірному випадку. Для цього у формулах (5.1–5.2) замість площі необхідно використовувати вирази для об'єму зі знаком. Для кожної вершини об'єм з урахуванням знака визначається як змішаний добуток трьох векторів, визначених суміжними ребрами. При використанні тетраедричних елементів таких добутоків буде чотири, а при шестигранних – вісім.

У тривимірному випадку для кожного елемента дискретної моделі $\mathbf{M}$, код якого k_e (n_e – кількість вершин у елементі), функція (5.1) матиме такий вигляд:

$$L(\mathbf{M}, k_e) = \sum_{j=1}^{n_e} e^{-t \alpha(\mathbf{M}, k_e, j)},$$

де функція α для тетраедрів визначається виразом (за умови порядку обходу вершин, зображеного на рис. 3.1, б)

$$\alpha(\mathbf{M}, k_e, j) = \begin{cases} [(\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 1)) \times (\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 1))] \cdot ((\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 1))), & j=1, \\ [(\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 2)) \times (\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 2))] \cdot ((\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 2))), & j=2, \\ [(\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 3)) \times (\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 3))] \cdot ((\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 3))), & j=3, \\ [(\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 4)) \times (\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 4))] \cdot ((\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 4))), & j=4. \end{cases}$$

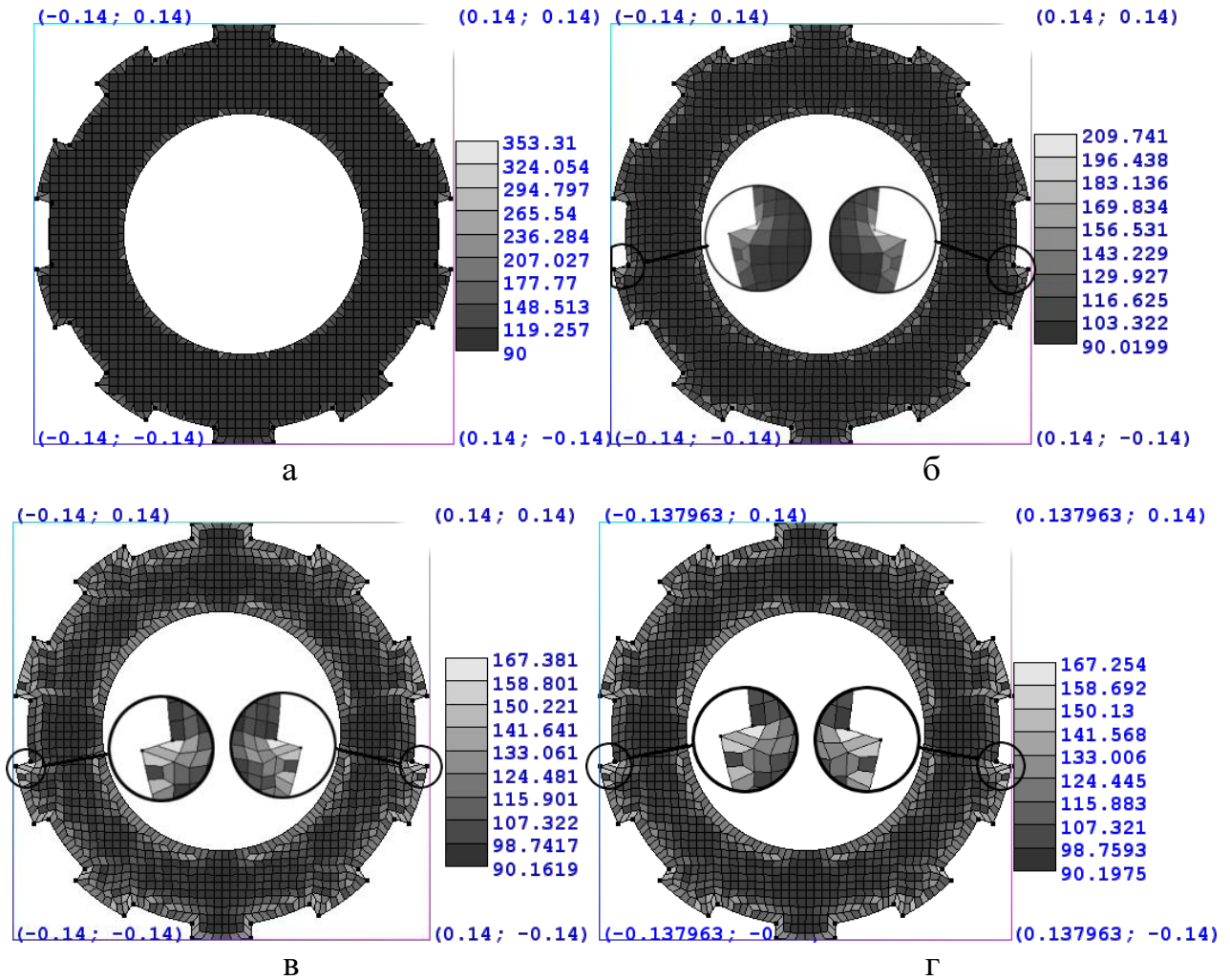


Рисунок 5.18 – Згладжування дискретної моделі форми фрикційного диска:

а – вихідна дискретна модель

б – 4 ітерації згладжування Лапласа

в – 16 ітерацій згладжування на основі локальної мінімізації функціоналу

г – результат глобальної мінімізації функціоналу (5.3)

Для шестигранників, якщо порядок обходу вершин відповідає зображеному на рис. 3.1, г, функція α отримає такий вигляд:

$$\alpha(\mathbf{M}, k_e, j) = \begin{cases} [(\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 1)) \times (\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 1))] \cdot ((\mathbf{M}(k_e, 5) - \mathbf{M}(k_e, 1))), & j=1, \\ [(\mathbf{M}(k_e, 6) - \mathbf{M}(k_e, 2)) \times (\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 2))] \cdot ((\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 2))), & j=2, \\ [(\mathbf{M}(k_e, 7) - \mathbf{M}(k_e, 3)) \times (\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 3))] \cdot ((\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 3))), & j=3, \\ [(\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 4)) \times (\mathbf{M}(k_e, 8) - \mathbf{M}(k_e, 4))] \cdot ((\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 4))), & j=4, \\ [(\mathbf{M}(k_e, 6) - \mathbf{M}(k_e, 5)) \times (\mathbf{M}(k_e, 1) - \mathbf{M}(k_e, 5))] \cdot ((\mathbf{M}(k_e, 8) - \mathbf{M}(k_e, 5))), & j=5, \\ [(\mathbf{M}(k_e, 7) - \mathbf{M}(k_e, 6)) \times (\mathbf{M}(k_e, 2) - \mathbf{M}(k_e, 6))] \cdot ((\mathbf{M}(k_e, 5) - \mathbf{M}(k_e, 6))), & j=6, \\ [(\mathbf{M}(k_e, 6) - \mathbf{M}(k_e, 7)) \times (\mathbf{M}(k_e, 8) - \mathbf{M}(k_e, 7))] \cdot ((\mathbf{M}(k_e, 3) - \mathbf{M}(k_e, 7))), & j=7, \\ [(\mathbf{M}(k_e, 7) - \mathbf{M}(k_e, 8)) \times (\mathbf{M}(k_e, 5) - \mathbf{M}(k_e, 8))] \cdot ((\mathbf{M}(k_e, 4) - \mathbf{M}(k_e, 8))), & j=8. \end{cases}$$

Наприклад, якщо як вихідну використовувати дискретну модель на базі тетраедрів (рис. 5.9, а) для форми об'єкта, заданого формулою (4.1), то у результаті чотирьох ітерацій згладжування внутрішніх вузлів на основі локальної мінімізації функціоналу максимальний кут у елементах буде знаходитись у діапазоні від $66,7^\circ$ до $151,2^\circ$ (рис. 5.19, а), в результаті восьми ітерацій – від $65,0^\circ$ до $149,6^\circ$ (рис. 5.19, б), а в результаті шістнадцяти ітерацій – від $63,1^\circ$ до $149,6^\circ$ (рис. 5.19, в). Значення основних характеристик дискретної моделі у залежності від кількості ітерацій такого згладжування показані у табл. 5.8.

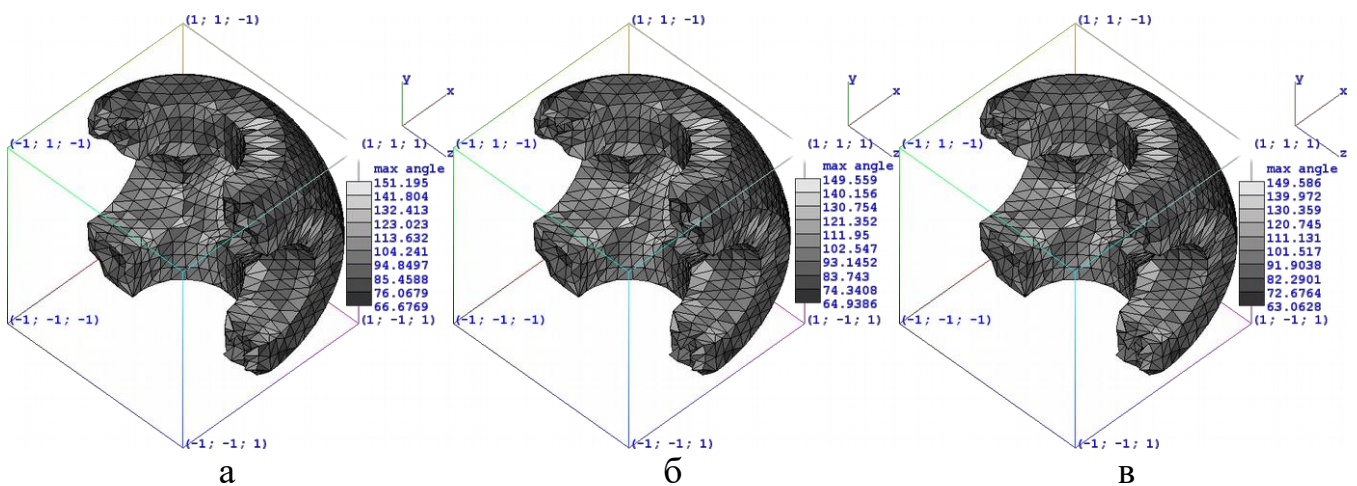


Рисунок 5.19 – Згладжування дискретної моделі на базі тетраедрів:

а – 4 ітерації згладжування на основі локальної мінімізації функціоналу

б – 8 ітерацій згладжування на основі локальної мінімізації функціоналу

в – 16 ітерацій згладжування на основі локальної мінімізації функціоналу

Таблиця 5.8 – Розподіл характеристик дискретної моделі на базі тетраедрів при згладжуванні з використанням локальної мінімізації функціоналу

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	0,6	4,7	4,3	4,2	4,1	4,1	4,1	4,1	4,0	4,0
$\alpha_{\max}, ^\circ$	177,2	157,7	152,4	151,4	151,2	150,7	150,2	149,8	149,6	149,6
$\min(l_{\min} / l_{\max})$	0,06	0,085	0,078	0,076	0,075	0,075	0,074	0,074	0,073	0,073

Якщо як вихідну використовувати дискретну модель на базі шестигранників (рис. 5.9, б) для форми об'єкта, заданого формулою (4.1), то у результаті чотирьох ітерацій згладжування внутрішніх вузлів на основі локальної мінімізації функціоналу максимальний кут в елементах буде знаходитися в діапазоні від $92,7^\circ$ до $162,2^\circ$ (рис. 5.20, а), в результаті восьми ітерацій – від $93,2^\circ$ до $162,2^\circ$ (рис. 5.20, б), а у результаті шістнадцяти ітерацій – від $93,7^\circ$ до $162,2^\circ$ (рис. 5.20, в). Зазначимо, що значення $162,2^\circ$ відповідає найбільшому кутку межевої грані. Значення основних характеристик дискретної моделі у залежності від кількості ітерацій такого згладжування показані у табл. 5.9.

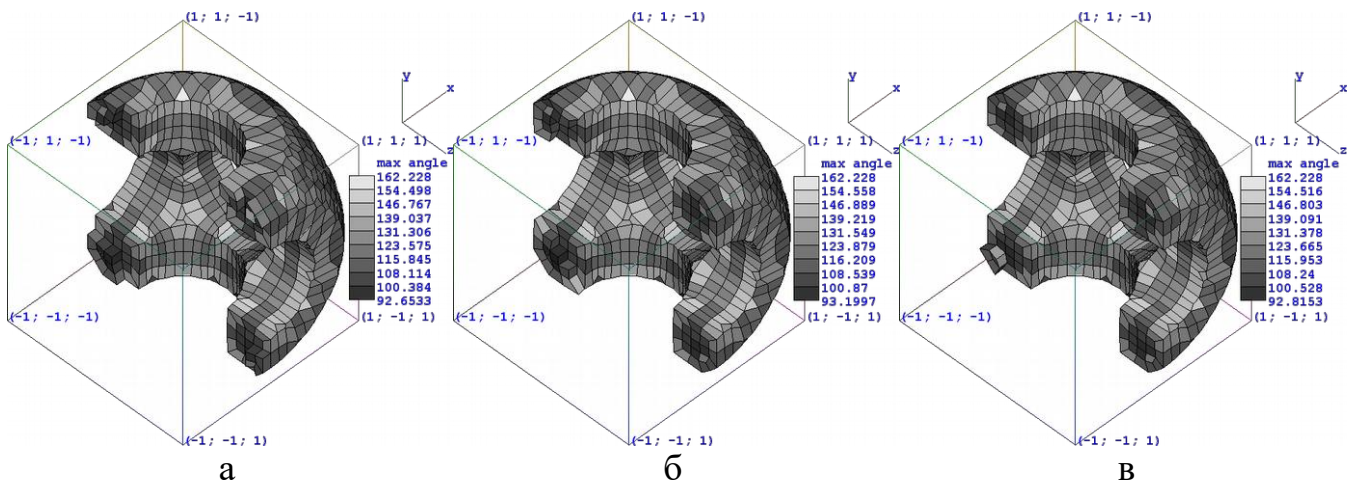


Рисунок 5.20 – Згладжування дискретної моделі на базі шестигранників:

а – 4 ітерації згладжування на основі локальної мінімізації функціоналу

б – 8 ітерацій згладжування на основі локальної мінімізації функціоналу

в – 16 ітерацій згладжування на основі локальної мінімізації функціоналу

Таблиця 5.9 – Розподіл характеристик дискретної моделі на базі шестигранників при згладжуванні з використанням локальної мінімізації функціоналу

Кількість ітерацій	0	1	2	3	4	5	6	7	8	9
$\alpha_{\min}, ^\circ$	7,5	15,4	24,1	27,2	28,3	28,3	28,3	28,3	28,3	28,3
$\alpha_{\max}, ^\circ$	177,2	170,6	162,2	162,2	162,2	162,2	162,2	162,2	162,2	162,2
$\min(l_{\min} / l_{\max})$	0,12	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13	0,13

Для порівняльної оцінки розглянемо час, необхідний на згладжування Лапласа, локальну і глобальну мінімізацію функціоналу (5.3), у залежності від числа вузлів у дискретної моделі пластинки з трикутним отвором (рис. 5.17). Для обчислювального експерименту використовувався комп'ютер з процесором Intel Core i5-3330 3.0GHz (4 ядра) і 8 гігабайтами оперативної пам'яті (результати показані у табл. 5.10). Під час обчислювального експерименту процедури згладжування Лапласа і локальної мінімізації з використанням запропонованого підходу повторювалися у чотирьох ітераціях. Можна побачити, що найменший час необхідний для виконання згладжування Лапласа. Локальна мінімізація у середньому на три порядки довше згладжування Лапласа і на три порядки швидше пошуку мінімуму у глобальній постановці.

Таблиця 5.10 – Залежність часу згладжування від кількості вузлів у дискретної моделі пластинки з трикутним отвором

Метод згладжування	Кількість вузлів						
	828	1138	1496	1912	2367	2874	3427
	Час, с						
Згладжування Лапласа	0,002	0,003	0,004	0,005	0,006	0,007	0,009
Локальна мінімізація	0,382	0,469	0,880	1,108	1,420	1,710	2,029
Глобальна мінімізація	63,620	102,300	271,000	428,316	662,700	947,434	1338,041

Отже, розроблений метод згладжування внутрішніх вузлів дискретних моделей на основі локальної мінімізації функціоналу дозволяє не тільки покращити їх

характеристики, а й усунути вироджені елементи (при використанні чотирикутників або шестигранників). Як видно з рис. 5.17–5.20, а також з табл. 5.7–5.9, для досягнення прийнятних результатів достатньо 8-10 ітерацій. Водночас дослідження витрат часу, дозволяє зробити, що локальний алгоритм мінімізації функціоналу в середньому на три порядки довший згладжування Лапласа і на три порядки швидший пошуку його мінімуму за допомогою глобального алгоритму.

5.5 Використання методу для побудови дискретних моделей поверхонь об'єктів, поданих гібридно

При моделюванні оболонкових конструкцій у таких галузях як, наприклад, ракетобудування, зручним у використанні може бути подання поверхні у вигляді (2.49). У такому випадку вихідними даними будуть:

- $\text{surface}(u, v)$ – параметрична поверхня, визначена на прямокутнику $\mathbf{P} = \{(u, v) | a \leq u \leq b, c \leq v \leq d\}$;
- $F(x, y, z)$ – неявна функція, яка більша нуля у внутрішніх точках поверхні об'єкта, що моделюється.

Структурно метод побудови тріангуляції такої поверхні можна подати трьома послідовними етапами:

- I – побудова дискретної моделі меж поверхні;
- II – побудова початкової тріангуляції;
- III – оптимізація тріангуляції.

Визначимо операцію обчислення відстані між двома точками як функцію

$$d(p, q) = \|\text{surface}(p) - \text{surface}(q)\|,$$

де p і q – координати точок у параметричному просторі ($p, q \in \mathbf{P}$).

Також визначимо функцію

$$W(u, v) = F(R(u, v)) \wedge \left[\frac{(b-a)^2}{4} - \left(u - \frac{a+b}{2} \right)^2 \right] \wedge \left[\frac{(d-c)^2}{4} - \left(v - \frac{d+c}{2} \right)^2 \right],$$

яка більше або дорівнює нулю тільки у внутрішніх точках для функції F прямокутника P .

Для подальшої побудови дискретної моделі використовуватимемо цю функцію W .

I. У параметричному просторі побудуємо дискретну модель меж об'єкта з використанням алгоритму `background-grid-mesh`. Для цього прямокутник P розіб'ємо на рівномірні фонові комірки (у загальному випадку можливо використовувати не рівномірне розбиття), а як функцію, яка описує форму поверхні, використаємо W .

II. Побудуємо початкову тріангуляцію для моделі меж об'єкта, отриманої на попередньому кроці, за допомогою алгоритму `delaunay` з перевіркою умови Делоне у параметричному просторі. Природно, що двовимірну тріангуляцію Делоне, побудована у параметричному просторі, не обов'язково буде зберігати відповідність критерію Делоне, після відображення сітки на поверхню.

III. Приведемо дискретну модель у відповідність критерію Делоне за допомогою алгоритму `refine_delaunay`, у якому для перевірки потрапляння в описане коло використовується алгоритм `circum-circle` для тривимірних трикутних граней (пошук центру і радіусу здійснюється у місцевій системі координат). Водночас координати центру описаного кола у параметричному просторі визначаються формулою

$$c = \operatorname{argmin}_{p \in P} d(\operatorname{surface}(p), C),$$

де C – центр описаного екваторіального кола у тривимірних координатах.

Наприклад, для поверхні Безье з круглим отвором, яка задана формулою (2.50), початкова тріангуляція у параметричному просторі та результат її покращення наведені на рис. 5.21.

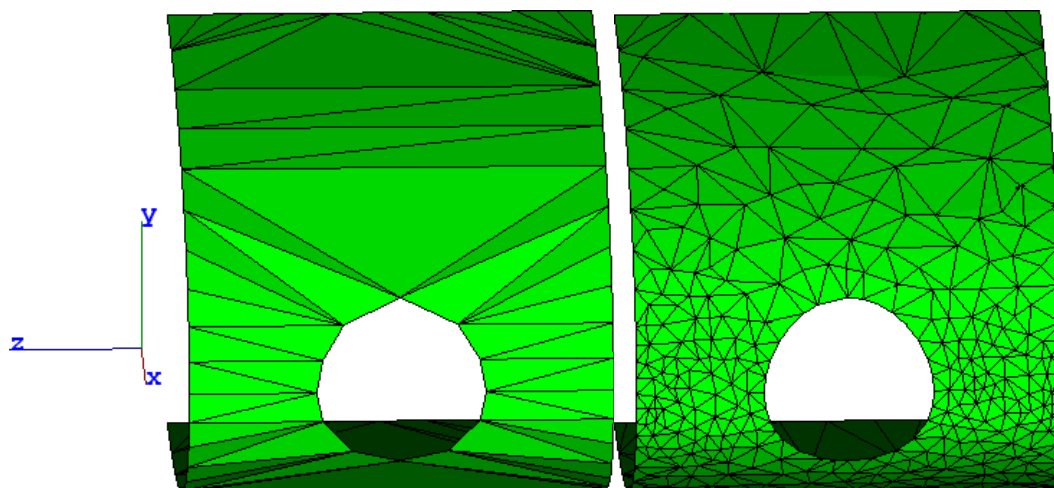


Рисунок 5.21 – Триангуляція поверхні Безьє з круглим отвором

Для моделі конічної оболонки, яка задана конічною параметричною поверхнею і функцією (2.51), що описує технологічні отвори, можна отримати дискретну модель у вигляді, наведеному рис. 5.22.

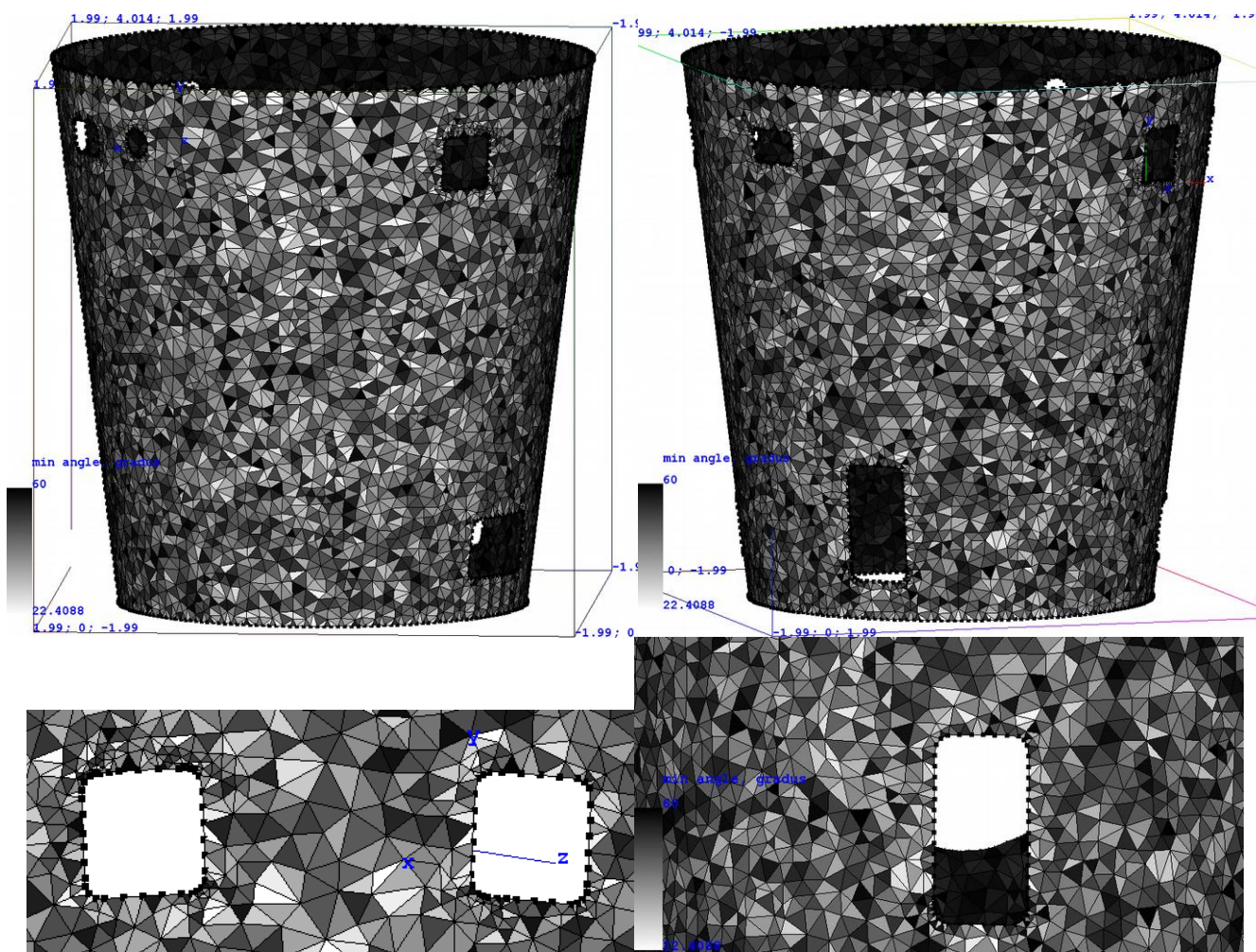


Рисунок 5.22 – Конічна оболонка: розподіл значень мінімальних кутів у трикутниках

Висновки до розділу 5

Отже, основним науковим результатом цього розділу є метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально. Цей метод дозволяє використовувати спільний алгоритм для двовимірних і тривимірних моделей на базі симплексів (трикутників і тетраедрів) і топологічних кубиків (чотирикутників і шестигранників). Підсумкові дискретні моделі, зазвичай, потребують додаткової обробки.

Для покращення характеристик дискретних моделей розроблено новий метод локального згладжування дискретних моделей форм об'єктів, який дозволяє усунути вироджені елементи (у випадку чотирикутників або шестигранників). Такий метод дозволяє отримати якість дискретних моделей, порівнянну з результатами глобальної оптимізації координат вузлів, але водночас використовує менше обчислювальних ресурсів.

Спільне використання узагальненого методу генерації дискретних моделей, методу згладжування координат вузлів на межі (розробленого у четвертому розділі) та згладжування внутрішніх вузлів дозволить генерувати дискретні моделі високої якості з адаптацією до зламів форм об'єктів.

Метод генерації дискретних моделей можна застосовувати для моделей оболонкових конструкцій з отворами при гібридному поданні.

Результати п'ятого розділу опубліковано у роботах [112, 158, 168, 170–171, 152, 176–177, 179, 181, 182, 238, 239].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилення [26, 135, 284, 299, 307, 395, 406, 417, 418, 439, 440].

6 АДАПТИВНІ ДИСКРЕТНІ МОДЕЛІ ФОРМ ОБ'ЄКТІВ МАШИНОБУДУВАННЯ

Описані вище методи генерації та згладжування дискретних моделей форм виробів дозволяють врахувати їх геометричні особливості при виборі координат вузлів. В результаті отримуємо дискретні моделі, адаптивні до геометрії виробів (характерних точок, зламів тощо). Проте, для застосування дискретних моделей форм виробів у скінченно-елементному аналізі необхідно враховувати точність чисельного рішення задачі.

У методі скінченних елементів шукана величина в кожному елементі апроксимується деякою елементарною функцією (наприклад, лінійною або білінійною). Відповідно, при збільшенні кількості вузлів і елементів буде збільшуватися точність такої апроксимації. Проте, пропорційно числу вузлів буде зростати розмірність розв'язуваної задачі. Тому раціонально збільшувати щільність дискретної моделі в областях, де найгірше апроксимовано рішення.

6.1 Дослідження впливу форми скінченного елемента на точність рішення

В основу методу скінченних елементів [503–505] покладено ідею апроксимації шуканої неперервної величини (наприклад, температури, переміщення тощо) за допомогою кусково неперервних функцій, які визначені на елементарних геометричних областях – скінченних елементах, що утворюють дискретну модель задачі. Для визначення значень шуканої неперервної величини у вузлах дискретної моделі будується система лінійних алгебраїчних рівнянь, матриця якої носить назву глобальної матриці жорсткості. Водночас вибір форми елемента і якість дискретизації вихідної області впливають на точність розв'язку. Так від форми скінченного елемента залежить степінь полінома, який використовується для апроксимації, а від якості дискретної моделі – значення якобіанів.

Типова схема дослідження поведінки об'єкта з використанням методу скінченних елементів ґрунтується на заміні варіаційної задачі системою лінійних алгебраїчних рівнянь виду:

$$[\mathbf{K}]\{\mathbf{X}\}=\{\mathbf{F}\}, \quad (6.1)$$

де $[\mathbf{K}]=\sum_e [\mathbf{K}_e]$ – глобальна матриця, яка є спеціальною сумою локальних матриць,

що апроксимують розв'язок на кожному скінченному елементі;

$\{\mathbf{X}\}$ – вектор, який апроксимує рішення задачі у вузлах скінченних елементів;

$\{\mathbf{F}\}$ – глобальний вектор, який є спеціальною сумою діючих сил.

6.1.1 Двовимірні задачі дослідження напружено-деформованого стану

Для дослідження поведінки призматичних тіл або пластин, товщина яких є сталою, а зовнішні сили діють перпендикулярно до поздовжньої вісі та не змінюються уздовж товщини, часто використовують скінченно-елементні моделі плоско-напруженого або плоско-деформованого стану. У обох випадках поле переміщень однозначно визначається переміщеннями u і v , які відповідають осям координат x і y .

У результаті мінімізації енергії деформацій об'єкта, який знаходиться під дією зовнішніх сил, для локальних матриць жорсткості справедливим буде такий вираз:

$$[\mathbf{K}_e]=h \iint [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dx dy, \quad (6.2)$$

де h – товщина об'єкта;

$[\mathbf{B}]$ – матриця, яка пов'язує переміщення та деформації;

$[\mathbf{D}]$ – матриця, яка пов'язує деформації та напруження.

Матрицю $[\mathbf{B}]$ у (6.2) можна подати у такому вигляді:

$$[\mathbf{B}] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \dots & \frac{\partial N_m}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \dots & 0 & \frac{\partial N_m}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \dots & \frac{\partial N_m}{\partial y} & \frac{\partial N_m}{\partial x} \end{bmatrix}, \quad (6.3)$$

де $N_i = N_i(x, y)$ ($i = 1 \dots m$) – функції форми скінченного елемента;

m – кількість вузлів елемента.

Матриця $[\mathbf{D}]$ у (6.2) для ізотропного матеріалу з модулем Юнга E і коефіцієнтом Пуассона ν у випадку плоско-напруженого стану має такий вигляд [503]:

$$[\mathbf{D}] = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1 - \nu}{2} \end{bmatrix},$$

У випадку плоско-деформованого стану:

$$[\mathbf{D}] = \frac{E(1 - \nu)}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1 - \nu} & 0 \\ \frac{\nu}{1 - \nu} & 1 & 0 \\ 0 & 0 & \frac{1 - 2\nu}{2(1 - \nu)} \end{bmatrix}.$$

Функції форми у (6.3) для лінійного трикутного скінченного елемента мають такий вигляд:

$$N_i(x, y) = \alpha_{i,0} + \alpha_{i,1}x + \alpha_{i,2}y, \quad i = \overline{1, 3},$$

а для білінійного чотирикутного скінченного елемента такий:

$$N_i(x, y) = \alpha_{i,0} + \alpha_{i,1}x + \alpha_{i,2}y + \alpha_{i,3}xy, \quad i \in \overline{1, 4},$$

де $\alpha_{i,j}$ – коефіцієнти, які обирають таким чином, щоб функція форми i -го вузла була рівною одиниці у цьому вузлі та нулю в інших.

1. Згин консолі, навантаженої на кінці. Консоль, яка має вузький (щодо довжини) поперечний переріз одиничної товщини, згинається силою P , що прикладена на кінці (рис. 6.1). Ширина консолі дорівнює l , а висота $2c$.

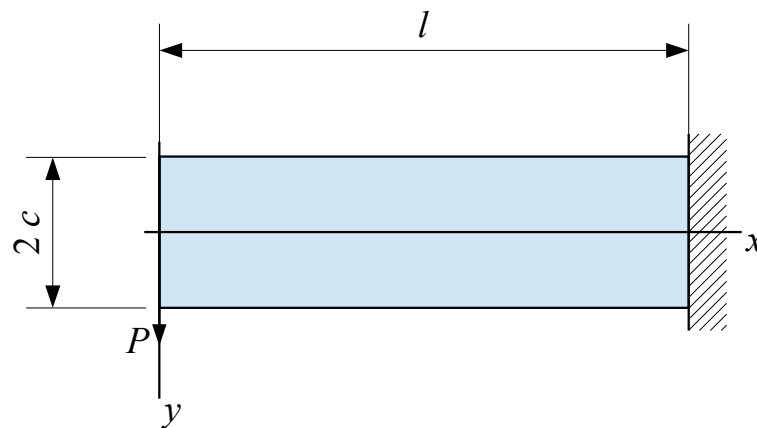


Рисунок 6.1 – Згин консолі: схема закріплення та дії сили

Якщо використовувати рівномірні дискретні моделі області, побудовані на 51×21 вузлі, то у результаті обчислювального експерименту при $P = 200$ МН, $E = 203200$ МПа, $\nu = 0,27$, $l = 10$ м, $c = 2$ м отримаємо, що при використанні чотирикутних елементів значення середини прогину навантаженого кінця консолі для середини консолі дорівнює $0,0635$ м (плоско-деформований стан). При використанні трикутних скінченних елементів – $0,0629$ м. Водночас відомо, що рівняння зігнутої осі консолі має такий вигляд [142]:

$$v(x)_{y=0} = \frac{P x^3}{6EI} - \frac{P l^2 x}{2EI} + \frac{P l^3}{3EI} + \frac{P c^2}{2IG}, \quad (6.4)$$

де $I = \frac{2}{3} c^3$ – момент інерції;

E – модуль пружності;

$G = \frac{E}{2(1+\nu)}$ – модуль зсуву;

ν – коефіцієнт Пуассона.

На кінці консолі при $x=0$ значення прогину, обчислене за формулою (6.4), дорівнює 0,0709 м. Отже, похибка для максимального значення прогину осі консолі складає приблизно 10% при використанні чотирикутників і 11% – трикутників. На рис. 6.2 порівнюється чисельні та аналітичне значення прогину осі консолі.

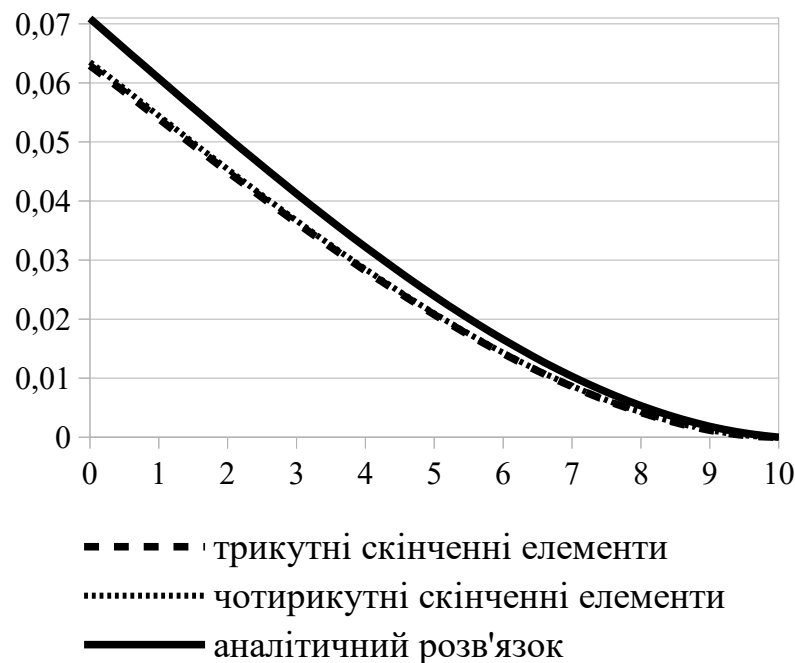


Рисунок 6.2 – Згин консолі: прогин осі консолі

Розподіл горизонтальних напружень у консолі описується формулою

$$\sigma_x = -\frac{P}{I} x y,$$

яка справедлива на значному віддаленні від кінців консолі [142]. Тому порівняємо значення напружень вздовж вертикальної осі консолі (при $x = \frac{l}{2}$). У табл. 6.1 наведено отримані результати тільки для інтервалу значень y від -2 до 0 (розподіл напружень по площині консолі є дзеркальним відносно початку координат).

Таблиця 6.1 – Порівняння розподілу напружень σ_x (МПа) вздовж вертикальної осі консолі

у, м	-2	-1,8	-1,6	-1,4	-1,2	-1	-0,8	-0,6	-0,4	-0,2	0
Трикутники	355,8	321,5	283,5	246,2	209,1	171,9	134,9	97,9	61,4	29,6	0
Чотирикутники	377,5	337,0	299,6	262,2	224,7	187,3	149,8	112,4	74,9	37,46	0
Аналітичний розв'язок	375	337,5	300	262,5	225	187,5	150	112,5	75	37,5	0
Трикутники (похибка, %)	5,13	4,75	5,50	6,21	7,09	8,30	10,09	12,97	18,20	20,96	–
Чотирикутники (похибка, %)	0,65	0,14	0,13	0,13	0,13	0,12	0,12	0,12	0,12	0,12	–

Отже, у результаті скінченно-елементного аналізу отримано, що чисельні значення для прогину осі консолі для трикутників і чотирикутників майже збігається, але останні дозволяють більш точно апроксимувати розподіл напружень, що обумовлено існуванням нелінійного члена у відповідних функціях форми.

2. Згин балки рівномірним навантаженням. Балка вузького прямокутного перерізу одиничної ширини, оперта на кінцях, згинається рівномірно розподіленим навантаженням q (рис. 6.3).

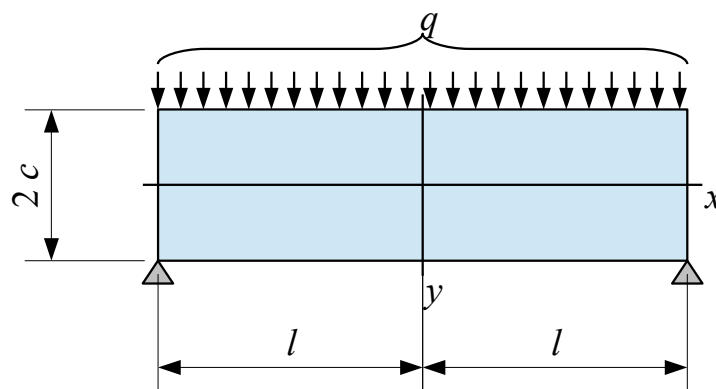


Рисунок 6.3 – Згин балки

Якщо довжина балки дорівнює $2l$, її висота – $2c$, модуль пружності – E , а коефіцієнт Пуассона – ν , то крива прогинів її горизонтальної осі має вигляд [142]:

$$v(x)_{y=0} = \frac{5}{24} \frac{q l^4}{E I} \left[1 + \frac{12}{5} \frac{c^2}{l^2} \left(\frac{4}{5} + \nu \right) \right] - \frac{q}{2 E I} \left[\frac{l^2 x^2}{2} - \frac{x^4}{12} - \frac{1}{5} c^2 x^2 + \left(1 + \frac{1}{2} \nu \right) c^2 x^2 \right].$$

Якщо для скінченно-елементного дослідження плоско-деформованого стану використовувати рівномірну структуровану дискретну модель, яка побудована на 51×11 вузлі для прямокутника $[-10; 10] \times [-2; 2]$ ($l=10$ м, $c=2$ м), то при інтенсивності навантаження $q=200$ МН/м, модулі пружності $E=203200$ МПа та коефіцієнті Пуассона $\nu=0,27$ похибка значень прогину складатиме у середньому 10% для трикутників і 7% для чотирикутників (рис. 6.4, а). Водночас на віддалені від кінців балки розподіл напружень σ_x визначається такою формулою [142]:

$$\sigma_x(x, y) = \frac{q}{2 I} (l^2 - x^2) y + \frac{q}{2 I} \left(\frac{2}{3} y^3 - \frac{2}{5} c^2 y \right).$$

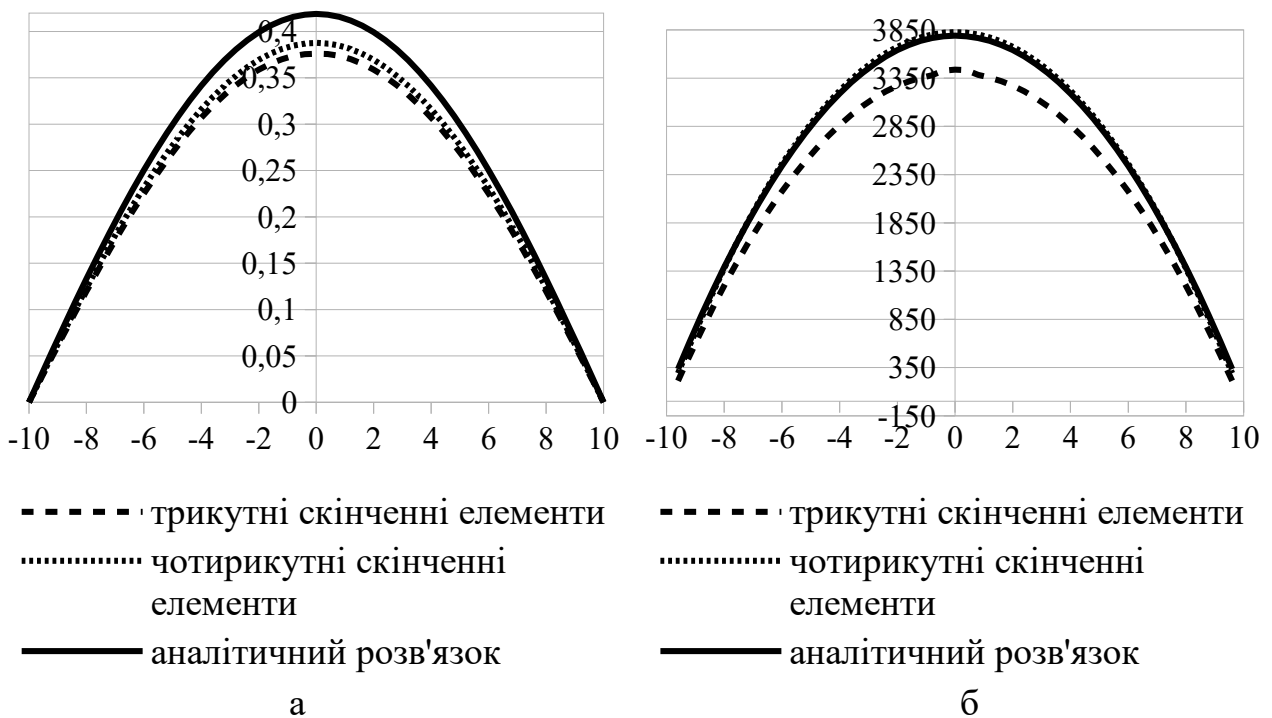


Рисунок 6.4 – Напружено-деформований стан балки:

а – прогин горизонтальної осі (м)

б – розподіл напружень σ_x вдовж горизонтальної осі (МПа)

На рис. 6.4, б можна побачити, що чотирикутні елементи більш точно апроксимують розподіл напружень уздовж горизонтальної осі балки.

6.1.2 Тривимірні задачі дослідження напружено-деформованого стану

У результаті мінімізації енергії деформацій об'єкта, який знаходиться під дією зовнішніх сил, у тривимірному випадку для локальних матриць жорсткості справедливим буде такий вираз:

$$[\mathbf{K}_e] = \iiint [\mathbf{B}]^T [\mathbf{D}] [\mathbf{B}] dx dy dz, \quad (6.5)$$

де $[\mathbf{B}]$ – матриця, яка пов'язує переміщення та деформації;

$[\mathbf{D}]$ – матриця, яка пов'язує деформації та напруження.

Матрицю $[\mathbf{B}]$ у (6.5) можна подати у такому вигляді:

$$[\mathbf{B}] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \frac{\partial N_2}{\partial x} & 0 & 0 & \dots & \frac{\partial N_m}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & 0 & \frac{\partial N_2}{\partial y} & 0 & \dots & 0 & \frac{\partial N_m}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & 0 & 0 & \frac{\partial N_2}{\partial z} & \dots & 0 & 0 & \frac{\partial N_m}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & 0 & \dots & \frac{\partial N_m}{\partial y} & \frac{\partial N_m}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial y} & \dots & 0 & \frac{\partial N_m}{\partial z} & \frac{\partial N_m}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_2}{\partial x} & \dots & \frac{\partial N_m}{\partial z} & 0 & \frac{\partial N_m}{\partial x} \end{bmatrix}, \quad (6.6)$$

де $N_i = N_i(x, y, z)$ ($i = 1 \dots m$) – функції форми скінченного елемента;

m – кількість вузлів елемента.

Матриця $[\mathbf{D}]$ у (6.5) для ізотропного матеріалу з модулем Юнга E і коефіцієнтом Пуассона ν у випадку тривимірного напружено-деформованого стану має такий вигляд [503]:

$$[\mathbf{D}] = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}.$$

Функції форми у (6.6) для лінійного тетраедричного скінченного елемента мають вигляд:

$$N_i(x, y, z) = \alpha_{i,0} + \alpha_{i,1}x + \alpha_{i,2}y + \alpha_{i,3}z, \quad i = \overline{1, 4},$$

Для трилінійного шестигранного скінченного елемента:

$$N_i(x, y, z) = \alpha_{i,0} + \alpha_{i,1}x + \alpha_{i,2}y + \alpha_{i,3}z + \alpha_{i,4}xy + \alpha_{i,5}xz + \alpha_{i,6}yz + \alpha_{i,7}xyz, \quad i \in \overline{1, 8}.$$

де $\alpha_{i,j}$ – коефіцієнти, які обирають таким чином, щоб функція форми i -го вузла дорівнювала одиниці у цьому вузлі та нулю в інших.

Розглянемо напружено-деформований стан призматичного стрижня під дією власної ваги. Якщо вертикальна вісь стрижня збігається з Oz, об'ємні сили діють у додатному напрямі цієї осі, $\gamma = \rho g$ – це питома вага стрижня, l – його довжина, E – модуль Юнга, ν – коефіцієнт Пуассона, то переміщення вздовж його вісі визначаються такою формулою [142]:

$$w(x, y, z) = \frac{\gamma z^2}{2E} + \frac{\nu \gamma}{2E} (x^2 + y^2) - \frac{\gamma l^2}{2E}. \quad (6.7)$$

Водночас у стрижні діють напруження

$$\sigma_z = \gamma z. \quad (6.8)$$

Для скінченно-елементного дослідження поведінки стрижня використаємо рівномірні структуровані дискретні моделі призматичної області $1 \times 1 \times 20$ м (довжина $l=20$ м), які побудовані на $3 \times 3 \times 41$ вузлі (рис. 6.5). Питома вага $\gamma=0,0765$ МН/м³, модуль Юнга $E=210000$ МПа, коефіцієнт Пуассона $\nu=0,3$. Вважаємо, що для кінця стрижня $z=l$ заборонені переміщення у третьому напрямі (вздовж осі Oz), а об'ємна сила, яка дорівнює γ , діє додатному напрямку.

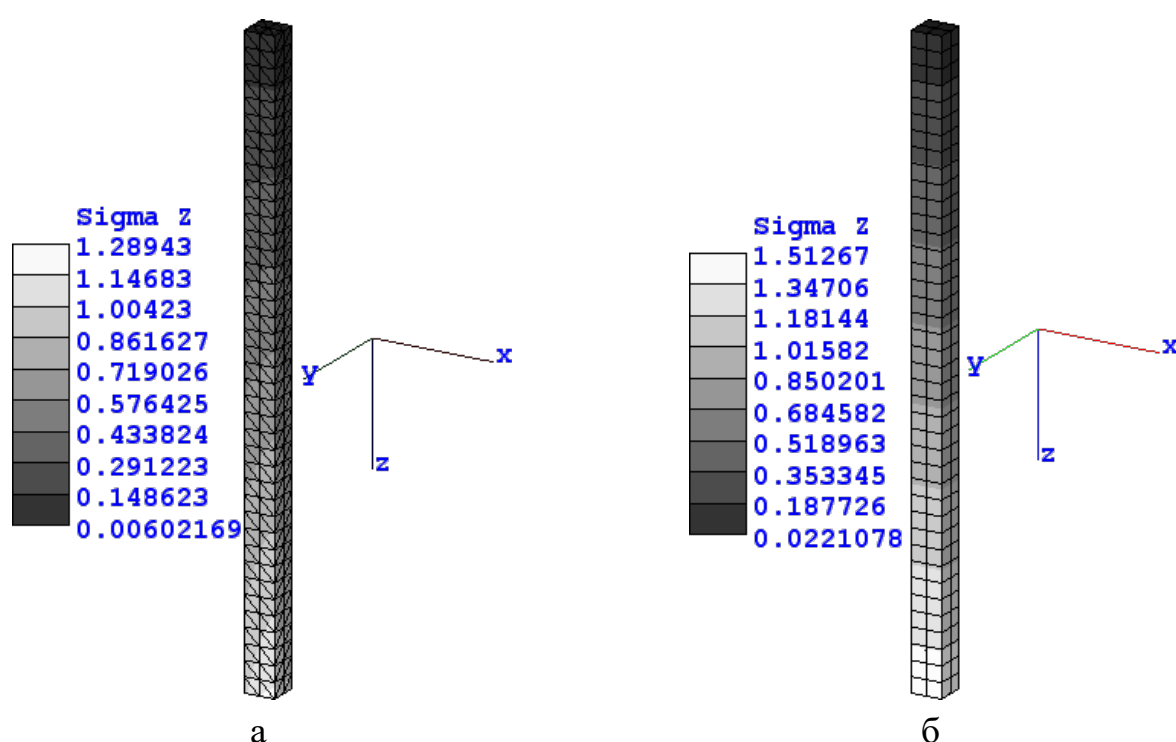


Рисунок 6.5 – Розподіл напружень σ_z (МПа) у дискретній моделі стрижня:

а – тетраедричні скінченні елементи

б – шестигранні скінченні елементи

У результаті скінченно-елементного аналізу отримано, що при використанні тетраедрів, максимальне подовження осі стрижня під дією власної ваги становить $7,2867 \times 10^{-5}$ м, при використанні кубиків – $7,2866 \times 10^{-5}$ м. Водночас для осі стрижня (якщо $x=0$ і $y=0$) за формулою (6.7) максимальне подовження дорівнює $7,2857 \times 10^{-5}$ м. Отже, різниця між розв'язками складає менше одного відсотку

(порівняння похибки відносно аналітичного розв'язку зображене на рис. 6.6, а). На рис. 6.6, б порівнюються розподіли значень напружень уздовж осі стрижня, отримані на базі кубічних і тетраедричних елементів, аналітичного розв'язку.

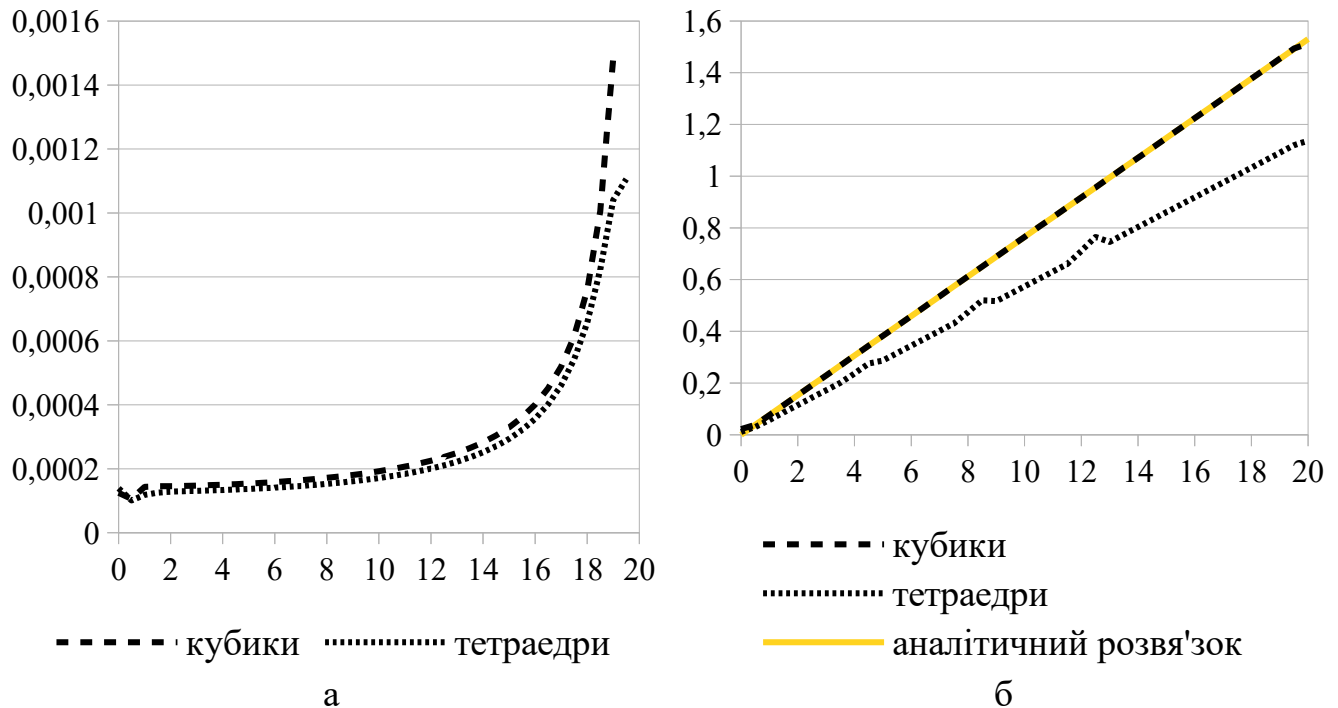


Рисунок 6.6 – Напружено-деформований стан стрижня:

а – похибка переміщення вісі у третьому напрямі відносно аналітичного розв'язку

б – розподіл напружень уздовж осі стрижня (МПа)

Можна побачити, що обидві форми елементів вельми точно апроксимують переміщення (похибка менша 1%), але шестигранники (кубики) більш точні в обчисленні напружень (розподіл для шестигранників майже збігається з аналітичним рішенням). Це обумовлено тим, що функції форми шестигранних скінченних елементів (на відміну від тетраедричних) мають чотири нелінійних члени. У результаті для апроксимації похідних використовуються несталі вирази.

Отже, за результатами виконаних обчислювальних експериментів можна зробити висновок, що, з чисельної точки зору, використання чотирикутників і шестигранників дозволяє отримати більш точні розв'язки відносно трикутників і тетраедрів.

6.2 Метод генерації адаптивних дискретних моделей форм об'єктів, заданих функціонально

Загальна ідея генерації адаптивних дискретних моделей форм об'єктів полягає розбитті певних елементів шляхом додавання нових вузлів за результатами аналізу деякої величини, яка визначена у кожному вузлі моделі.

Нехай для форми об'єкта Ω , що задана функцією F , побудована дискретна модель M і множина кодів елементів V , які необхідно розбити. Техніку додавання нових вузлів у центри описаних кіл (або сфер у тривимірному випадку) можна використовувати для згущення дискретних моделей форм об'єктів на базі симплексів. Результатом того, що нові вузли додаються у центри описаних кіл, є виконання критерію Делоне на кожному кроці адаптації. Проте, ця техніка практично не застосовна для чотирикутників і шестигранників.

З точки зору розробки узагальнених методів генерації дискретних моделей форм об'єктів перспективною є розробка шаблонів і правил розбиття кожного елементу моделі та відновлення коректності топології дискретної моделі шляхом розбиття сусідніх елементів.

Наприклад, на першому етапі кожен трикутник можна розбити на три нових додаванням вузлів у середини його сторін (рис. 6.7). Далі кожен трикутник, у якого є спільне ребро з одним із розбитих на першому етапі, розбивається на дві частини шляхом додавання нового вузла у середину цього ребра. Як результат отримано два можливих шаблони розбиття. Ця схема відома під назвою «зелені трикутники» [204]. Вона дозволяє рівномірно розбивати дискретні моделі форм об'єктів на базі трикутних елементів.

Схему «зелені трикутники» можна узагальнити на випадок тетраедрів. На першому етапі кожний обраний тетраедр можна розбити на вісім елементів додаванням нових вузлів у середини його сторін (рис. 6.8–6.9). Далі, якщо у сусіднього тетраедра є спільне ребро з тетраедром, розбитим на першому етапі, то він розбивається на два елементи шляхом додавання нового вузла у середину цього ребра (рис. 6.9, а).

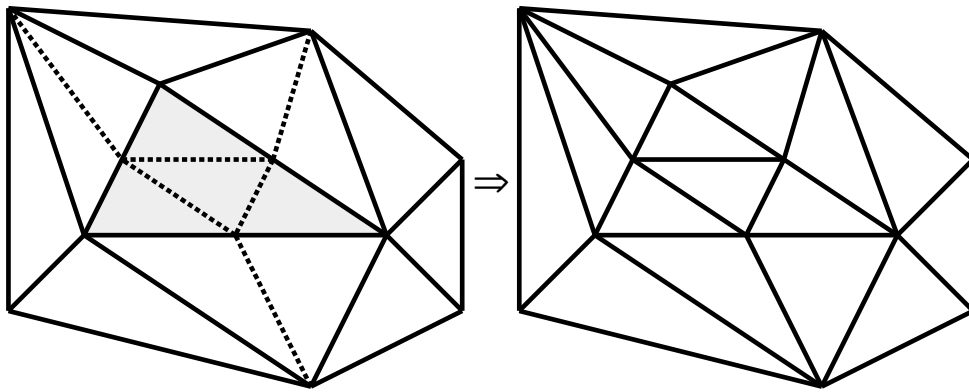


Рисунок 6.7 – Схема розбиття трикутних елементів

Якщо у тетраедра, сусіднього з тетраедром, розбитим на першому етапі, є спільна грань, то він розбивається на чотири частини шляхом додавання нових вузлів у середини ребер, які належать спільній грані (рис. 6.9, б). Отже, для тетраедрів отримаємо три шаблони розбиття.

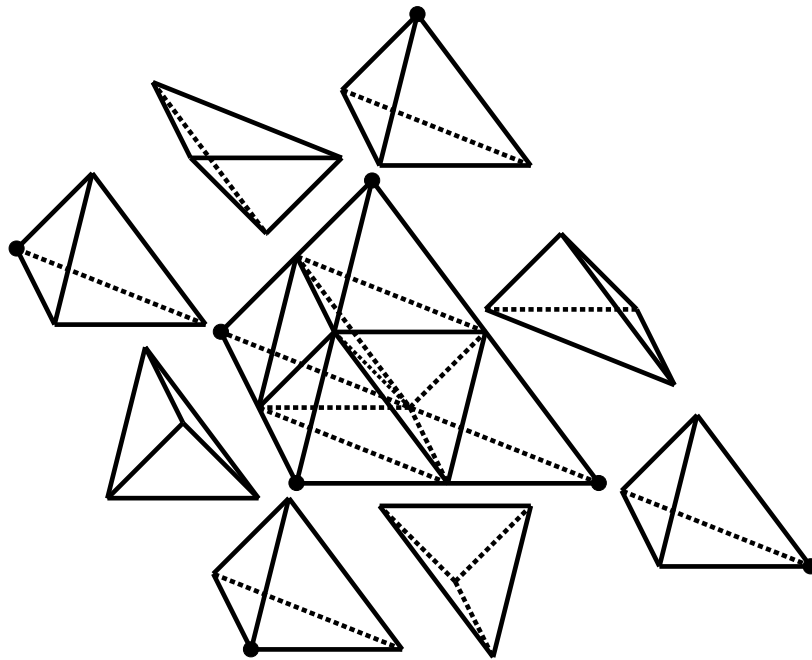


Рисунок 6.8 – Схема розбиття тетраедричних елементів

Для дискретних моделей на базі чотирикутних елементів на першому етапі можна використовувати схему розбиття елемента на дев'ять частин (рис. 6.10) [430]. Водночас для сусідніх елементів на другому етапі буде достатньо двох додаткових шаблонів. Якщо у сусіднього елемента є ребро, спільне з розбитим на першому етапі, то він розбивається на чотири частини. Якщо у чотирикутника є два суміжних ребра, які спільні з розбитим на першому етапі, то він розбивається на п'ять частин.

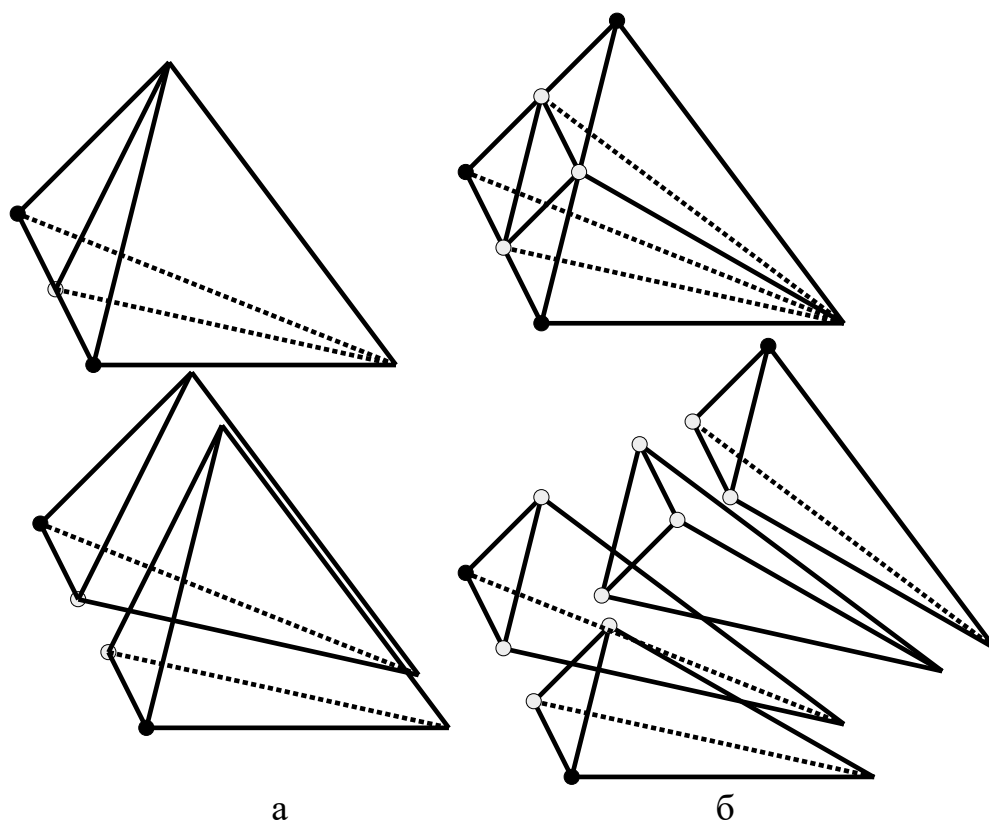


Рисунок 6.9 – Додаткові шаблони розбиття тетраедричних елементів:
 а – спільне ребро
 б – спільна грань

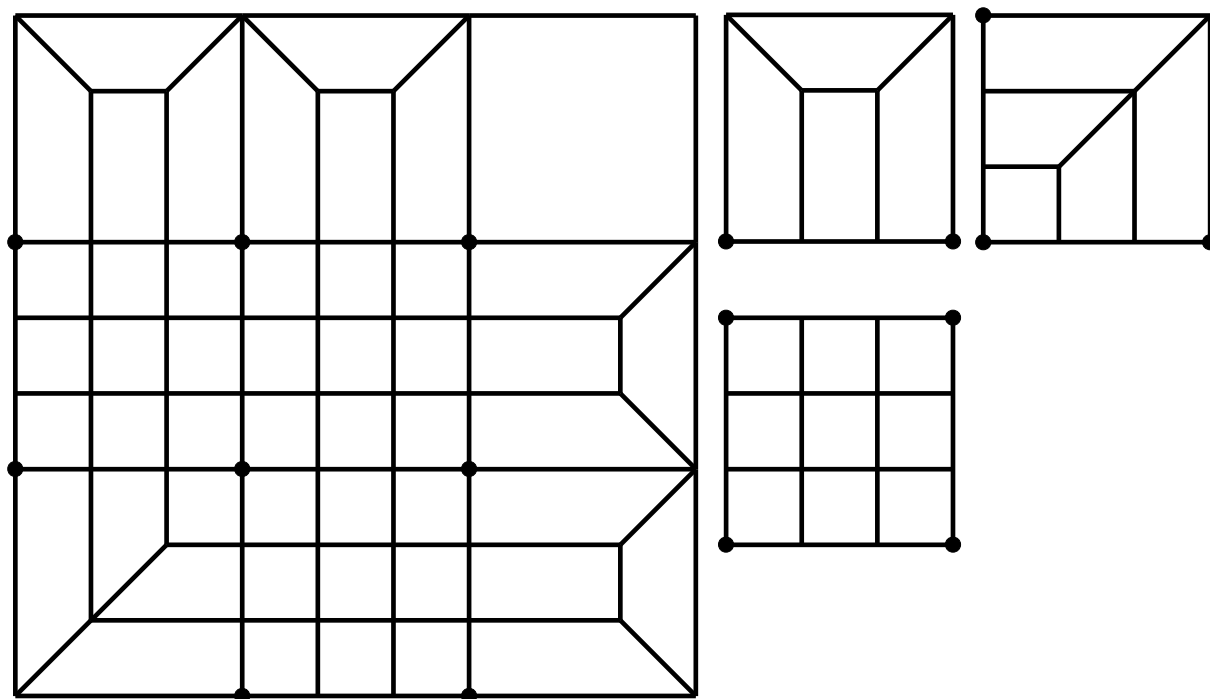


Рисунок 6.10 – Схема розбиття чотирикутних елементів

У випадку дискретних моделей на базі шестигранних елементів на першому етапі кожен шестигранник розбиваємо на 27 частин (рис. 6.11) [430]. На наступному етапі, якщо у деякого елемента ребро є учасником розбиття, то він розбивається на п'ять частин. Якщо у шестигранника одна грань є учасником розбиття, то його розбиваємо на тринадцять частин. Якщо у розбитті беруть участь три вершини, які належать до однієї грані, то такий шестигранник розбиваємо на шість частин. Якщо у розбитті беруть участь шість вершин, які належать двом суміжним граням, то такий елемент розбиваємо на п'ять частин. У двох останніх випадках для нових елементів необхідно додатково застосувати процедуру перевірки необхідності додаткового розбиття.

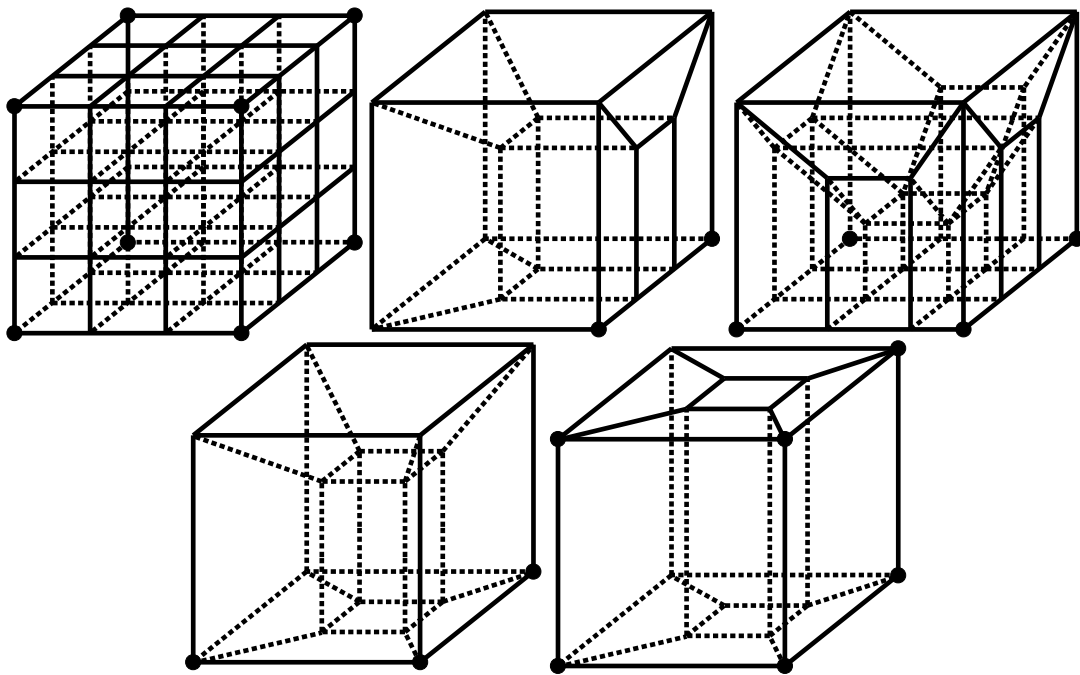


Рисунок 6.11 – Схема розбиття шестигранних елементів

Отже, для кожної найбільш поширеної форми елемента визначені основні схеми розбиття. Всього їх з урахуванням обертання для трикутників – 8, тетраедрів і чотирикутників – 16, шестигранників – 256. Якщо кожному кожному вузлу, що бере участь у розбитті, буде відповідати код 1, а всім іншим – 0, то кожному шаблону розбиття буде відповідати двійковий код. У результаті можна використовувати такий алгоритм побудови розбиття:

algorithm subdivide

input:

F – функція, що подає форму об'єкта Ω

M – дискретна модель

V – множина елементів для розбиття

ε – чисельний нуль

begin

$E \leftarrow \emptyset$

$N \leftarrow \emptyset$

for each $k_e \in V$ **do**

$e \leftarrow M(k_e)$

for each $k_n \in e$ **do**

$N \leftarrow N \cup k_n$

$E \leftarrow E \cup M.\text{incident}(k_n)$

end for

end for

for each $k_e \in E$ **do**

$e \leftarrow M(k_e)$

$code \leftarrow 0$

for $i \leftarrow 1$ **to** $|e|$ **do**

if $e[i] \in N$ **then**

$code \leftarrow code \mid 2^{i-1}$

end if

end for

$S \leftarrow \text{template}(M.V(e), code)$

for each $s \in S$ **do**

$\text{insert\_element}(M, s, \varepsilon)$

end for

$\text{remove\_element}(M, k_e)$

end for


```

for each  $n \in \delta(\mathbf{M}, \mathbf{N})$  do
  if  $F(n.v) \geq \varepsilon$  then
     $h \leftarrow \min \|n.v - \text{neighbors}(\mathbf{M}, \mathbf{M.code}(n))\|$ 
     $n.v \leftarrow \text{find\_border}(n.v, F, h, \varepsilon)$ 
  end if
end for
end

```

Отже, на першому кроці алгоритму формуємо дві множини: $\mathbf{E}$ і $\mathbf{N}$ – відповідно, множини елементів і вузлів, які беруть участь у розбитті. Множина $\mathbf{N}$ складається з кодів всіх вузлів, які належать елементам з вихідної множини $\mathbf{V}$, а множина $\mathbf{E}$ – з елементів, інцидентним у вузлах $\mathbf{N}$.

На другому кроці алгоритму для кожного елемента з $\mathbf{E}$ визначаємо його двійковий код: якщо вузол у вершині бере участь у розбитті, то у позиції, визначеній порядком обходу вершин елемента, ставимо 1, інакше – 0. Далі за допомогою алгоритму *template* формуємо нові елементи.

На третьому етапі поновлюємо координати межових вузлів. Для цього обчислюємо проєкції на межу Ω вузлів, для яких функція F не рівна 0.

Вигляд алгоритму для формування нових елементів залежить від форми елемента. Наприклад, для трикутного елемента (порядок обходу основних і додаткових вузлів наведено на рис. 6.12, а), він матиме такий вигляд:

algorithm *template*

input:

$\mathbf{V}$ – список з трьох вершин трикутника

code – код шаблону

output:

$\mathbf{E}_{code}$ – множина елементів, які відповідають коду шаблону

begin

$\mathbf{V}_4 \leftarrow 0,5 (\mathbf{V}_1 + \mathbf{V}_2)$

$$V_5 \leftarrow 0,5 (V_2 + V_3)$$

$$V_6 \leftarrow 0,5 (V_3 + V_1)$$

$$E_0 \leftarrow \{(V_1, V_2, V_3)\}$$

$$E_1 \leftarrow \{(V_1, V_2, V_3)\}$$

$$E_2 \leftarrow \{(V_1, V_2, V_3)\}$$

$$E_3 \leftarrow \{(V_1, V_4, V_3), (V_2, V_3, V_4)\}$$

$$E_4 \leftarrow \{(V_1, V_2, V_3)\}$$

$$E_5 \leftarrow \{(V_1, V_2, V_6), (V_3, V_6, V_2)\}$$

$$E_6 \leftarrow \{(V_2, V_5, V_1), (V_3, V_1, V_5)\}$$

$$E_7 \leftarrow \{(V_1, V_4, V_6), (V_2, V_5, V_4), (V_3, V_6, V_5), (V_4, V_5, V_6)\}$$

return E_{code}

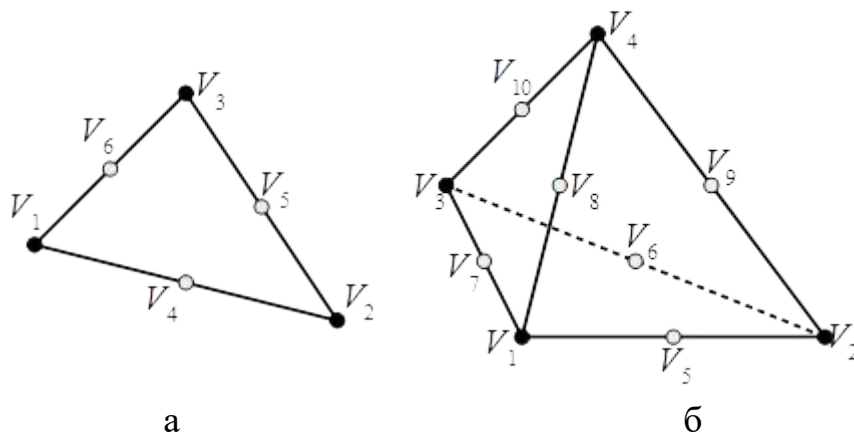


Рисунок 6.12 – Нумерація основних і вузлів у симплексах:

а – трикутник

б – тетраедр

Для тетраедрів порядок обходу основних і додаткових вузлів зображено на рис. 6.12, б. У результаті алгоритм формування нових елементів отримає такий вигляд:

algorithm template

input:

V – список з чотирьох вершин тетраедра

$code$ – код шаблона

output:

E_{code} – множина елементів, які відповідають коду шаблону

begin

$$V_5 \leftarrow 0,5 (V_1 + V_2)$$

$$V_6 \leftarrow 0,5 (V_2 + V_3)$$

$$V_7 \leftarrow 0,5 (V_3 + V_1)$$

$$V_8 \leftarrow 0,5 (V_1 + V_4)$$

$$V_9 \leftarrow 0,5 (V_2 + V_4)$$

$$V_{10} \leftarrow 0,5 (V_3 + V_4)$$

$$E_0 \leftarrow \{(V_1, V_2, V_3, V_4)\}$$

$$E_1 \leftarrow \{(V_1, V_2, V_3, V_4)\}$$

$$E_2 \leftarrow \{(V_1, V_2, V_3, V_4)\}$$

$$E_3 \leftarrow \{(V_1, V_5, V_3, V_4), (V_2, V_3, V_5, V_4)\}$$

$$E_4 \leftarrow \{(V_1, V_2, V_3, V_4)\}$$

$$E_5 \leftarrow \{(V_1, V_2, V_7, V_4), (V_3, V_7, V_2, V_4)\}$$

$$E_6 \leftarrow \{(V_2, V_6, V_1, V_4), (V_3, V_1, V_6, V_4)\}$$

$$E_7 \leftarrow \{(V_1, V_5, V_7, V_4), (V_2, V_6, V_5, V_4), (V_3, V_7, V_6, V_4), (V_5, V_6, V_7, V_4)\}$$

$$E_8 \leftarrow \{(V_1, V_2, V_3, V_4)\}$$

$$E_9 \leftarrow \{(V_1, V_2, V_3, V_8), (V_8, V_2, V_3, V_4)\}$$

$$E_{10} \leftarrow \{(V_1, V_2, V_3, V_9), (V_1, V_9, V_3, V_4)\}$$

$$E_{11} \leftarrow \{(V_1, V_8, V_5, V_3), (V_2, V_5, V_9, V_3), (V_4, V_9, V_8, V_3), (V_5, V_8, V_9, V_3)\}$$

$$E_{12} \leftarrow \{(V_1, V_2, V_3, V_{10}), (V_1, V_2, V_{10}, V_4)\}$$

$$E_{13} \leftarrow \{(V_1, V_7, V_8, V_2), (V_3, V_{10}, V_7, V_2), (V_4, V_8, V_{10}, V_2), (V_7, V_{10}, V_8, V_2)\}$$

$$E_{14} \leftarrow \{(V_2, V_9, V_6, V_1), (V_3, V_6, V_{10}, V_1), (V_4, V_{10}, V_9, V_1), (V_6, V_9, V_{10}, V_1)\}$$

$$E_{15} \leftarrow \{(V_1, V_5, V_7, V_8), (V_2, V_6, V_5, V_9), (V_3, V_7, V_6, V_{10}), (V_8, V_9, V_{10}, V_4), \\ (V_5, V_9, V_7, V_8), (V_7, V_9, V_{10}, V_8), (V_5, V_7, V_9, V_6), (V_7, V_{10}, V_9, V_6)\}$$

return E_{code}

Для чотирикутних елементів основні та додаткові вузли можливо перенумерувати з використанням номерів рядків і стовпців, на яких вони

розташовані (рис. 6.13). У результаті алгоритм формування нових елементів отримає такий вигляд:

algorithm template

input:

V – список з чотирьох вершин чотирикутника

$code$ – код шаблона

output:

E_{code} – множина елементів, які відповідають коду шаблона

begin

$$\xi_j \leftarrow -1 + \frac{2}{3}j, \quad j = \overline{0, 3}$$

$$\eta_i \leftarrow -1 + \frac{2}{3}i, \quad i = \overline{0, 3}$$

$$v_{ij} \leftarrow \frac{1}{4} [V_1(1 - \xi_j)(1 - \eta_i) + V_2(1 + \xi_j)(1 - \eta_i) + V_3(1 + \xi_j)(1 + \eta_i) + V_4(1 - \xi_j)(1 + \eta_i)]$$

$$E_0 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_1 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_2 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_3 \leftarrow \{(v_{00}, v_{03}, v_{12}, v_{10}), (v_{10}, v_{12}, v_{22}, v_{20}), (v_{20}, v_{22}, v_{33}, v_{30}), (v_{12}, v_{03}, v_{33}, v_{22})\}$$

$$E_4 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_5 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_6 \leftarrow \{(v_{00}, v_{03}, v_{12}, v_{11}), (v_{00}, v_{11}, v_{31}, v_{30}), (v_{31}, v_{11}, v_{12}, v_{32}), (v_{32}, v_{12}, v_{03}, v_{33})\}$$

$$E_7 \leftarrow \{(v_{00}, v_{03}, v_{12}, v_{10}), (v_{10}, v_{12}, v_{21}, v_{20}), (v_{20}, v_{21}, v_{31}, v_{30}), (v_{21}, v_{12}, v_{32}, v_{31}), \\ (v_{12}, v_{03}, v_{33}, v_{32})\}$$

$$E_8 \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_9 \leftarrow \{(v_{00}, v_{01}, v_{21}, v_{30}), (v_{01}, v_{02}, v_{22}, v_{21}), (v_{02}, v_{03}, v_{33}, v_{22}), (v_{30}, v_{21}, v_{22}, v_{33})\}$$

$$E_{10} \leftarrow \{(v_{00}, v_{03}, v_{33}, v_{30})\}$$

$$E_{11} \leftarrow \{(v_{00}, v_{01}, v_{11}, v_{10}), (v_{01}, v_{02}, v_{22}, v_{11}), (v_{02}, v_{03}, v_{33}, v_{22}), (v_{10}, v_{11}, v_{22}, v_{20}), \\ (v_{20}, v_{22}, v_{33}, v_{30})\}$$

$$E_{12} \leftarrow \{(v_{00}, v_{03}, v_{13}, v_{11}), (v_{11}, v_{13}, v_{23}, v_{21}), (v_{21}, v_{23}, v_{33}, v_{30}), (v_{00}, v_{11}, v_{21}, v_{30})\}$$

$$\mathbf{E}_{13} \leftarrow \{(v_{00}, v_{01}, v_{21}, v_{30}), (v_{01}, v_{02}, v_{12}, v_{21}), (v_{02}, v_{03}, v_{13}, v_{12}), (v_{21}, v_{12}, v_{13}, v_{23}), \\ (v_{30}, v_{21}, v_{23}, v_{33})\}$$

$$\mathbf{E}_{14} \leftarrow \{(v_{00}, v_{03}, v_{13}, v_{11}), (v_{11}, v_{13}, v_{23}, v_{22}), (v_{22}, v_{23}, v_{33}, v_{32}), (v_{32}, v_{31}, v_{11}, v_{22}), \\ (v_{00}, v_{11}, v_{31}, v_{30})\}$$

$$\mathbf{E}_0 \leftarrow \{(v_{00}, v_{01}, v_{11}, v_{10}), (v_{10}, v_{11}, v_{21}, v_{20}), (v_{20}, v_{21}, v_{31}, v_{30}), (v_{01}, v_{02}, v_{12}, v_{11}), \\ (v_{11}, v_{12}, v_{22}, v_{21}), (v_{21}, v_{22}, v_{32}, v_{31}), (v_{02}, v_{03}, v_{13}, v_{12}), (v_{12}, v_{13}, v_{23}, v_{22}), \\ (v_{22}, v_{23}, v_{33}, v_{32})\}$$

return $\mathbf{E}_{code}$

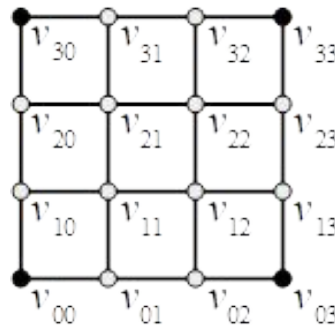


Рисунок 6.13 – Нумерація основних і додаткових вузлів у чотирикутниках

Для шестигранних елементів кількість можливих шаблонів дорівнює 256. Через велику довжину списку елементів алгоритм формування нових шестигранних елементів не приведено.

6.3 Метод формування множини елементів для розбиття

Як зазначено у підрозділі 6.1, для знаходження значень невідомої неперервної величини у вузлах дискретної моделі форми об'єкта будують систему лінійних алгебраїчних рівнянь, матрицю якої називають глобальною матрицею. Процес побудови цієї матриці називають ансамблюванням. Ансамблювання є спеціальною сумою локальних матриць, кожна з яких є результатом інтегрування вихідного диференціального рівняння на скінченному елементі. Розмір глобальної матриці

дорівнює добутку кількості вузлів у дискретній моделі форми об'єкта на кількість степенів свободи.

Права частина системи лінійних алгебраїчних рівнянь визначається як вектор вузлових сил (або значень, що відповідають впливу зовнішнього середовища на об'єкт). Вектор вузлових сил так само, як і глобальна матриця, є спеціальною сумою локальних векторів. Його розмір відповідає розміру глобальної матриці.

Для забезпечення єдиності розв'язання отриманої системи лінійних алгебраїчних рівнянь необхідно враховувати крайові умови, задані для розв'язуваної задачі. Такими умовами, наприклад, виступають умови закріплення.

Значення шуканої величини у вузлах дискретної моделі (наприклад, переміщень) отримують у результаті розв'язання системи лінійних алгебраїчних рівнянь. Після чого на завершальному етапі виконують їх обробку: обчислюють похідні величини (наприклад, напруження), друкують звіти та візуалізують результати.

Отже, для скінченно-елементного аналізу можна сформулювати такий алгоритм:

algorithm fem

input:

M – дискретна модель форми об'єкта

P – множина сил, під дією яких знаходиться об'єкт

B – множина крайових умов

output:

X – значення шуканої величини у вузлах дискретної моделі

begin

$$[\mathbf{K}] \leftarrow \sum_{e \in \mathbf{M}.E} [\mathbf{k}_e]$$

$$\{\mathbf{F}\} \leftarrow \sum_{e \in \mathbf{M}.E} \{\mathbf{f}_e(\mathbf{P})\}$$

process_initial_values($[\mathbf{K}]$, $\{\mathbf{F}\}$, **B**)

$$\{\mathbf{X}\} \leftarrow [\mathbf{K}]^{-1} \{\mathbf{F}\}$$

process_solution($\{\mathbf{X}\}$)

return $\{\mathbf{X}\}$

При адаптивному скінченно-елементному аналізі обчислювальний процес є ітеративним. На кожній ітерації будуємо глобальні матрицю та вектор, враховуємо крайові умови та розв'язуємо систему лінійних алгебраїчних рівнянь. Далі за результатами обробки розв'язку або його похідних формуємо список елементів, які необхідно розбити. Виконуємо розбиття обраних елементів. Після чого починаємо наступну ітерацію. Алгоритм зупиняємо, якщо список елементів для розбиття був пустим (або виконано максимально допустиму кількість ітерацій розбиття). Алгоритм описаного методу можна подати у такому вигляді:

algorithm adaptive_fem

input:

R – функція, яка подає форму об'єкта Ω

M – дискретна модель форми об'єкта Ω

P – множина сил, під дією яких знаходиться об'єкт Ω

B – множина крайових умов

δ – обмеження на максимальне значення похибки на елементі

k – максимальна кількість ітерацій

ε – чисельний нуль

output:

M – адаптована дискретна модель форми об'єкта Ω

X – вузлові значення шуканої величини

begin

$i \leftarrow 1$

do

$$[K] \leftarrow \sum_{e \in M.E} [k_e]$$

$$\{F\} \leftarrow \sum_{e \in M.E} \{f_e(P)\}$$

process_initial_values($[K]$, $\{F\}$, B)

$$\{X\} \leftarrow [K]^{-1} \{F\}$$

$C \leftarrow \text{character\_value}(\{X\})$

```

V  $\leftarrow \emptyset$ 
for each  $e \in \mathbf{M.E}$  do
    if  $\text{error}(e, \mathbf{C}) > \delta$  then
        V  $\leftarrow \mathbf{V} \cup \mathbf{M.code}(e)$ 
    end if
end for
subdivide(R, M, V,  $\epsilon$ )
i  $\leftarrow i+1$ 
while  $i \leq k$  and  $\mathbf{V} \neq \emptyset$ 
    process_solution({X})
return {X}

```

Як тестовий приклад розглянемо задачу дослідження напружено-деформованого стану фланця (рис. 6.14). В безрозмірних величинах товщину вважаємо рівною 20, модуль Юнга – 200000, коефіцієнт Пуассона – 0,33. Нехай на фланець діє сила, яка рівномірно розподілена вздовж отвору з радіусом 30, з інтенсивністю 20 (вектор сили направлений вздовж осі абсцис). Обидва отвори з радіусом 10 вважаємо жорстко закріпленими.

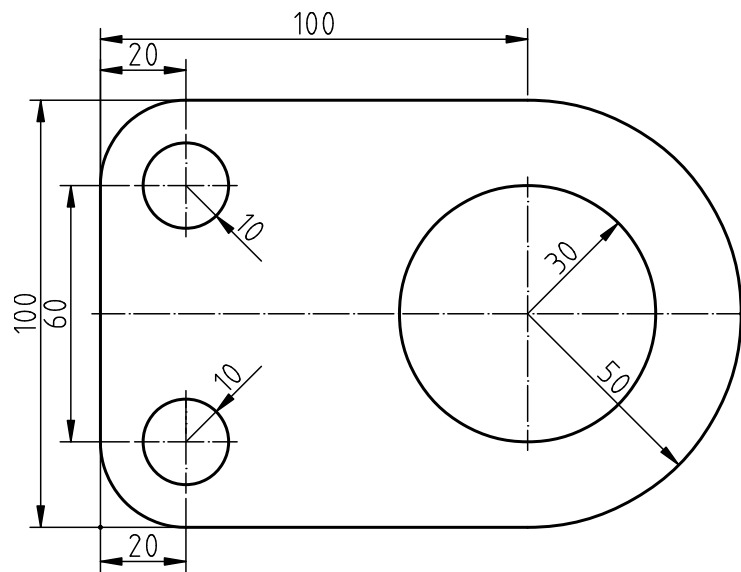


Рисунок 6.14 – Креслення фланця

Модель такого фланця можна описати з використанням функції

$$\text{flange}(x, y) = [\text{rectangle}(x-60, y-50, 80, 100) \vee \text{rectangle}(x-50, y-50, 100, 60) \vee \text{circle}(x-20, y-20, 20) \vee \text{circle}(x-20, y-80, 20) \vee \text{circle}(x-100, y-50, 50)] \wedge \neg \text{circle}(x-100, y-50, 30) \wedge \neg \text{circle}(x-20, y-20, 10) \wedge \neg \text{circle}(x-20, y-80, 10),$$

де circle – функція (2.7);

rectangle – функція (2.10).

Нехай вихідні дискретні моделі отримані за допомогою алгоритму background-grid-mesh з використанням рівномірної фонові сітки для 46×31 вузлів (рис. 6.15).

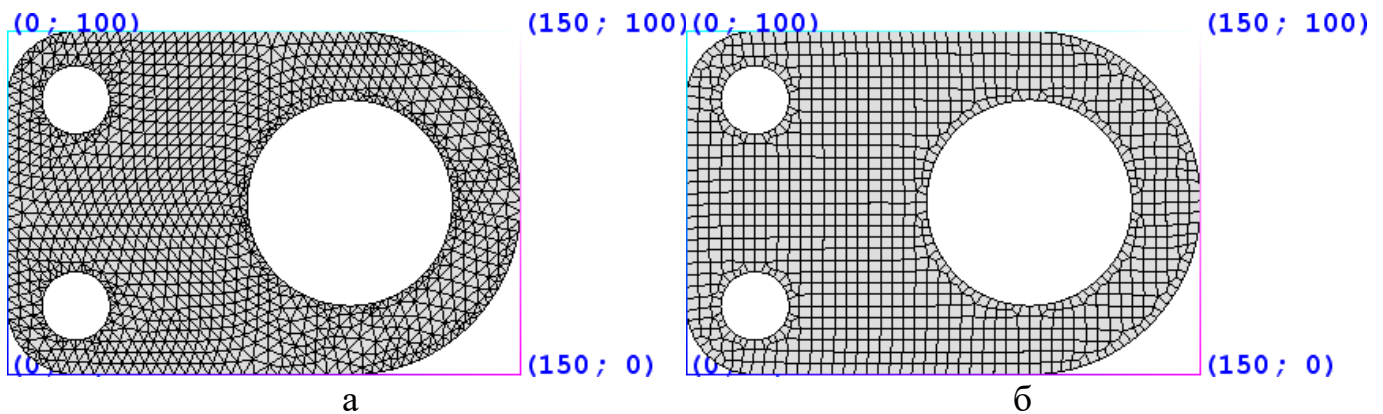


Рисунок 6.15 – Вихідна дискретна модель фланця:

а – на базі трикутників

б – на базі чотирикутників

У результаті скінченно-елементного аналізу плоско-деформованого стану фланця отримаємо, якщо використати трикутні скінченні елементи (лінійні функції форми), то переміщення у першому напрямку (x) знаходяться в інтервалі $[0; 0,0497]$, у другому напрямі (y) – $[-0,0168; 0,0168]$, напруження: σ_{xx} – $[-24,683; 128,299]$, σ_{yy} – $[-67,017; 67,624]$, τ_{xy} – $[-43,924; 46,691]$, інтенсивність напружень за Мізесом σ_i – $[1,558; 132,909]$ (рис. 6.16). Якщо використати чотирикутні скінченні елементи (білінійні функції форми), то переміщення у першому напрямі (x) знаходяться в інтервалі $[0; 0,0503]$, у другому напрямі (y) – $[-0,0172; 0,0172]$, напруження: σ_{xx} – $[-31,130; 141,285]$, σ_{yy} – $[-77,568; 71,964]$, τ_{xy} – $[-47,566; 47,151]$, інтенсивність напружень за Мізесом σ_i – $[2,343; 141,520]$ (рис. 6.16).

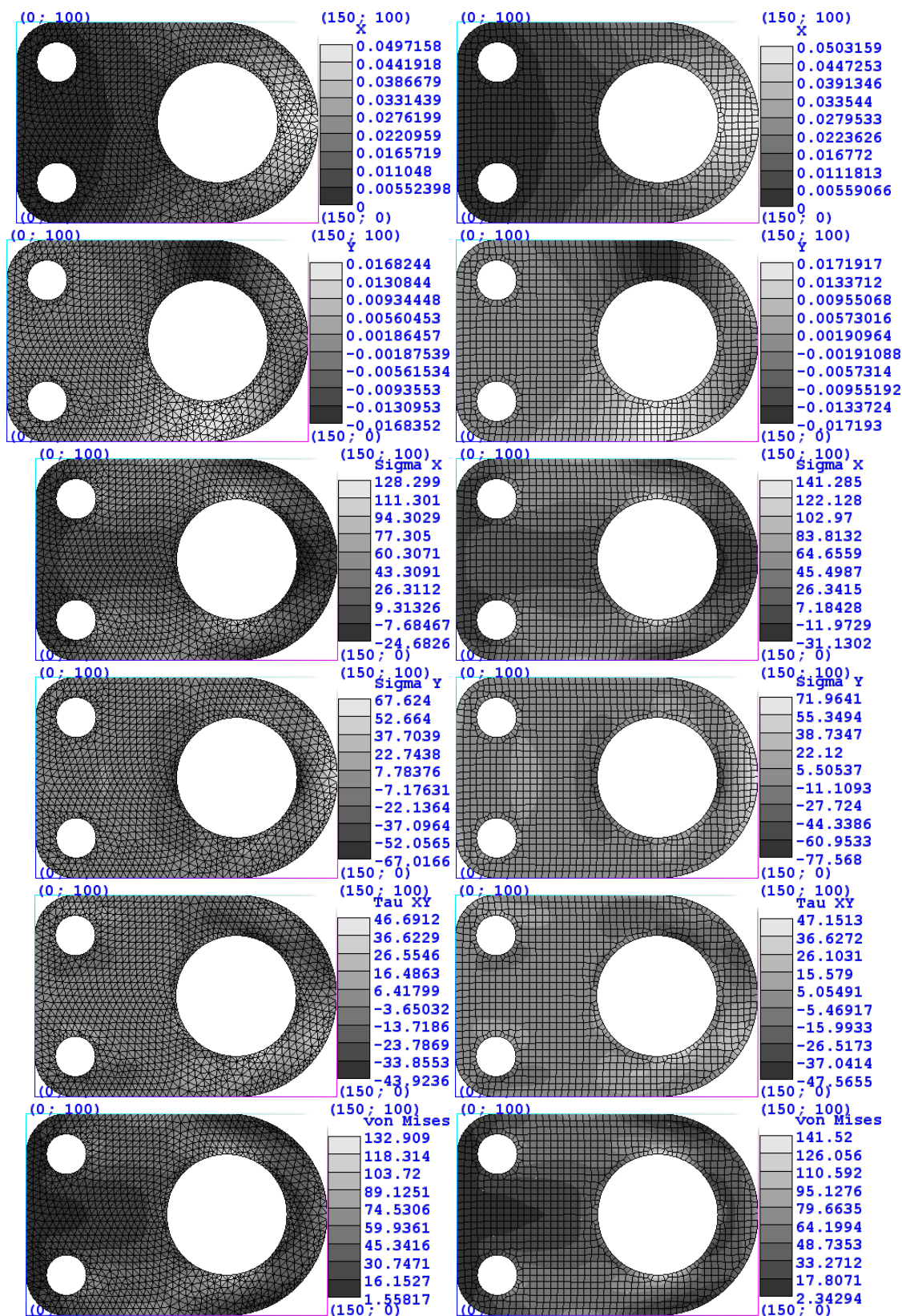


Рисунок 6.16 – Порівняння картин напружено-деформованого стану фланця при використанні трикутних і чотирикутних елементів

Поширені техніки знаходження областей для збільшення щільності вузлів і елементів використовують різні оцінки точності результату. Одним з найпростіших

методів оцінки точності (якщо невідомий точний розв'язок) є порівняння зміни вузлових значень з середнім на елементі [280]. У такому випадку функція $error$, яка використана в алгоритмі адаптивного скінченно-елементного аналізу матиме такий вигляд:

$$error(e, C) = \max_{i, j \in e} \frac{|C_i - C_j|}{\frac{1}{|e|} \left| \sum_{k \in e} C_k \right|}. \quad (6.9)$$

Якщо використати формулу (6.9) в алгоритмі `adaptive_fem` з обмеженням максимальної похибки на елементі $\delta=0,4$ і максимальною кількістю ітерацій $k=3$ (C – довжина вектора переміщень), то для обох форм скінченних елементів щільність вузлів буде більшою біля отвору радіуса 10 (рис. 6.17).

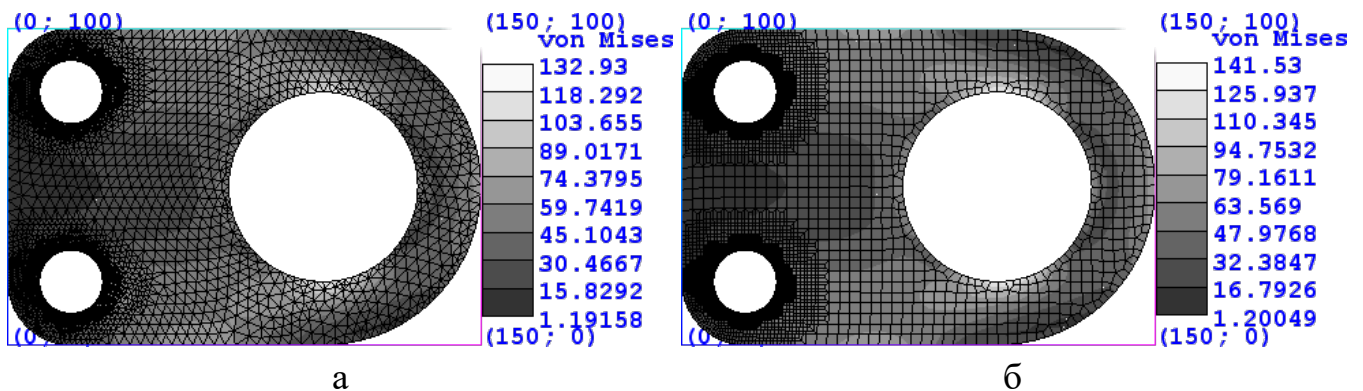


Рисунок 6.17 – Адаптивна дискретна модель фланця з використанням формули (6.9):

а – на базі трикутників

б – на базі чотирикутників

Альтернативою формули (6.9) є градієнтна оцінка точності розв'язку, яка базується на ідеї збільшення щільності вузлів і елементів дискретних моделей форм об'єктів у областях з найбільшою зміною певного скаляра, що потребують особливої уваги для дослідника (наприклад, інтенсивність напружень при дослідженні міцності). Як результат функція $error$, яка використана в алгоритмі адаптивного скінченно-елементного аналізу матиме такий вигляд:

$$\text{error}(e, C) = \max_{i,j \in e} \frac{C_i - C_j}{\max_k C_k - \min_k C_k}. \quad (6.10)$$

Якщо у алгоритмі `adaptive_fem` використати формули (6.10) з обмеженням максимально допустимої похибки на елементі $\delta=0,2$ і максимальною кількістю ітерацій $k=3$ (C – інтенсивність напружень за Мізесом σ_i), то для обох форм скінченних елементів збільшення щільності вузлів буде біля отвору радіуса 30, де найбільше відносна зміна σ_i (рис. 6.18).

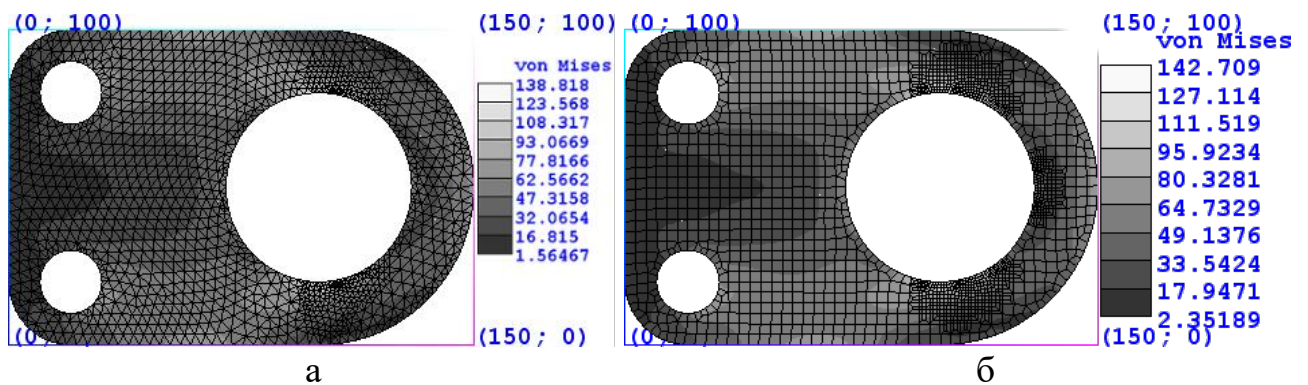


Рисунок 6.18 – Адаптивна дискретна модель фланця з використанням формули (6.10):

а – на базі трикутників

б – на базі чотирикутників

Для дослідження напружено-деформованого стану фланця у тривимірній постановці можна використати таку функцію, що подає його форму:

$$\text{flange3}(x, y, z) = \text{flange}(x, y) \wedge z \wedge (20 - z).$$

У тривимірному випадку, якщо для адаптивного скінченно-елементного аналізу використовувати дискретні моделі на базі і тетраедрів (рис. 6.19, а) і шестигранників (рис. 6.19, б) з обмеженням максимальної похибки на елементі $\delta=0,2$ і максимальною кількістю ітерацій $k=3$ (C – інтенсивність напружень за Мізесом σ_i), то зони з найбільшою щільністю вузлів такі ж самі як у наведених вище двовимірних розв'язках (рис. 6.18).

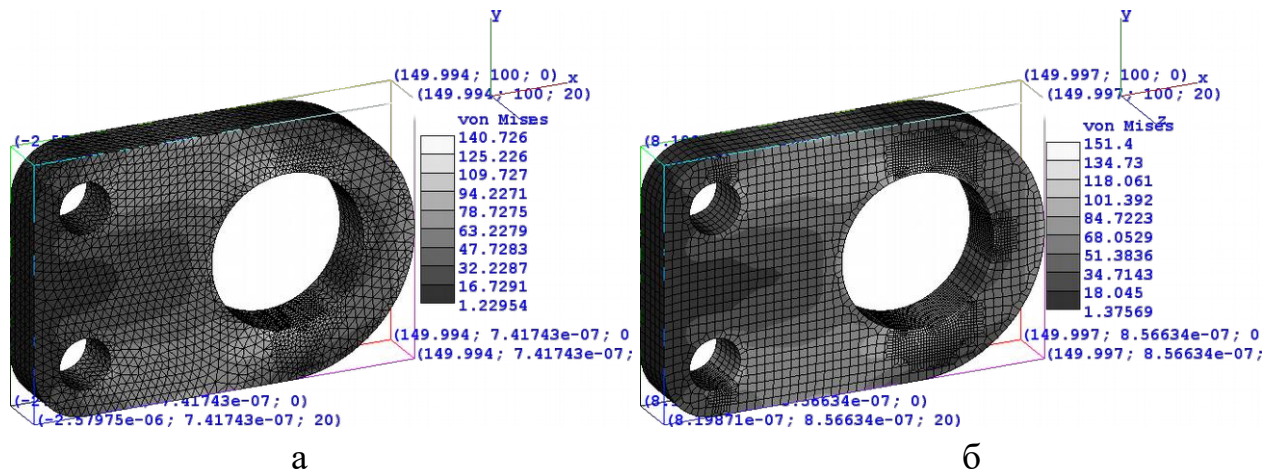


Рисунок 6.19 – Адаптивна тривимірна дискретна модель фланця з використанням формули (6.10):

а – на базі тетраедрів

б – на базі шестигранників

Зазначимо, що для цього прикладу різниця чисельних значень для напружень між звичайним і адаптивним підходами складає декілька відсотків. Якщо порівняти результати плоско-напруженого стану з тривимірними, то різниця складає приблизно 10%, що обумовлено використаними гіпотезами для двовимірного випадку.

Для порівняння результатів використання адаптивних дискретних моделей форм об'єктів з відомими аналітичними розв'язками розглянемо задачу дослідження у тривимірній постановці напружено-деформованого стану стрижня, якій знаходиться під дією власної ваги (з параметрами як у пункті 6.1.2). Якщо як вихідні використовувати рівномірні дискретні моделі, наведені на рис. 6.5, то для моделі на базі тетраедрів з обмеженням максимальної похибки на елементі $\delta=0,05$ і максимальною кількістю ітерацій $k=3$ (C – напруження σ_z) напруження будуть σ_z в інтервалі $[0,006; 1,53]$ МПа. Якщо використовувати модель на базі шестигранників з обмеженням максимальної похибки на елементі $\delta=0,02$ і максимальною кількістю ітерацій $k=3$, то напруження будуть σ_z в інтервалі $[0,009; 1,53]$ МПа. Аналітичний розв'язок (6.8) відповідає інтервалу $[0; 1,53]$ МПа, для вихідної моделі на базі тетраедрів $\sigma_z \in [0,006; 1,29]$ МПа, для кубиків – $\sigma_z \in [0,022; 1,51]$ МПа. Отже, у обох випадках уточнюються екстремальні значення напружень σ_z . Водночас якісно дискретні моделі перебудовуються біля вільного кінця стрижня.

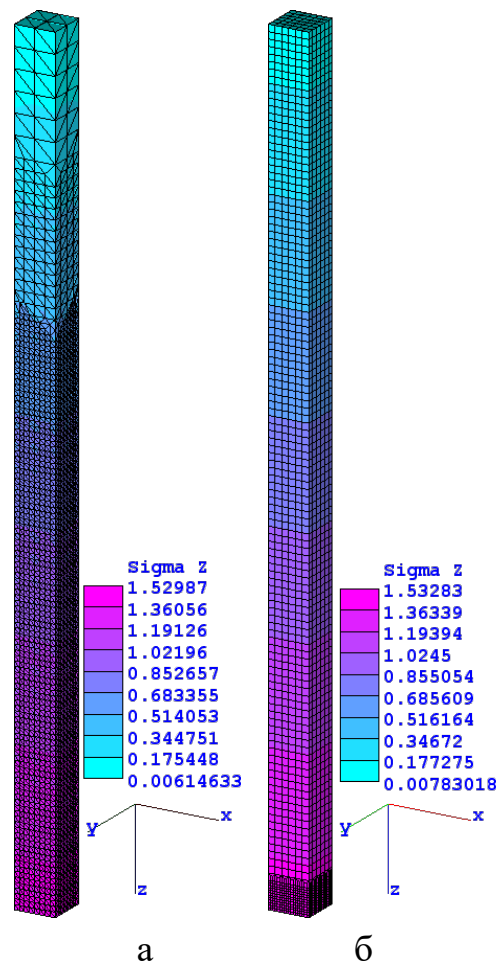


Рисунок 6.20 – Адаптивна тривимірна дискретна модель стрижня:

а – на базі тетраедрів

б – на базі шестигранників

Як додатковий приклад застосування методу генерації адаптивних дискретних моделей розглянемо задачу дослідження поведінки балки з отворами малого радіуса (рис. 6.21). Нехай лівий край балки затиснений, а на правий діє сила $P=1000$ lbs. Довжина балки дорівнює 10 in. Два круглих отвори радіуса $R=0,045$ in знаходяться на верхній і нижній гранях на відстані 9 in.



Рисунок 6.21 – Балка з отворами малого радіуса

У роботі [396] порівняно аналітичний і скінченно-елементний розв'язки на базі системи ANSYS для обчислення концентраторів напруження у горизонтальному напрямі при модулі Юнга $E=1$ psi і коефіцієнті Пуассона $\nu=0$. Так у цій роботі показано, що аналітична оцінка напружень у горизонтальному напрямі біля отворів складає 156849,3 psi. Водночас обчислення у системі ANSYS при використанні дискретної моделі на базі чотирикутників з 16 елементами для апроксимації сторони отвору максимальне значення напружень у горизонтальному напрямі дорівнює 124994 psi.

Функціональну модель форми такої балки можна подати функцією

$$\text{beam}(x, y) = \text{rectangle}(x, y, 10, 1) \wedge \\ \wedge \neg \text{circle}(x+4; y-0,5; 0,045) \wedge \neg \text{circle}(x+4; y+0,5; 0,045),$$

де circle – функція (2.7);

rectangle – функція (2.10).

Якщо як вигідну для узагальненого методу генерації дискретних моделей (алгоритм background-grid-mesh) використати рівномірну структуровану сітку чотирикутників на 101×11 вузлі, то отримаємо дискретну модель, яка має 1103 вузли та 996 елементів (рис. 6.22). За результатами скінченно-елементного аналізу на базі цієї моделі максимальне значення напружень σ_x дорівнює 103482 psi.

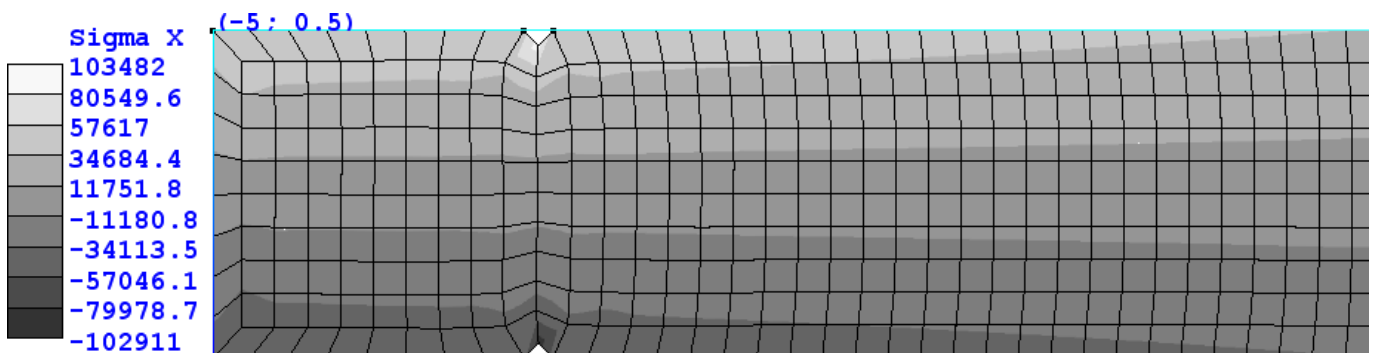


Рисунок 6.22 – Напруження σ_x у балці

Якщо використати адаптивний скінченно-елементний аналіз із обмеженням максимальної похибки на елементі $\delta=0,1$ і максимальною кількістю ітерацій $k=4$ (С

– напруження σ_x), то максимальне за модулем значення напружень у першому напрямі дорівнюватиме 159323 psi. Водночас у підсумковій дискретній моделі буде вже 4975 вузлів і 4744 елементи, щільність яких буде більшою біля отворів (рис. 6.23).

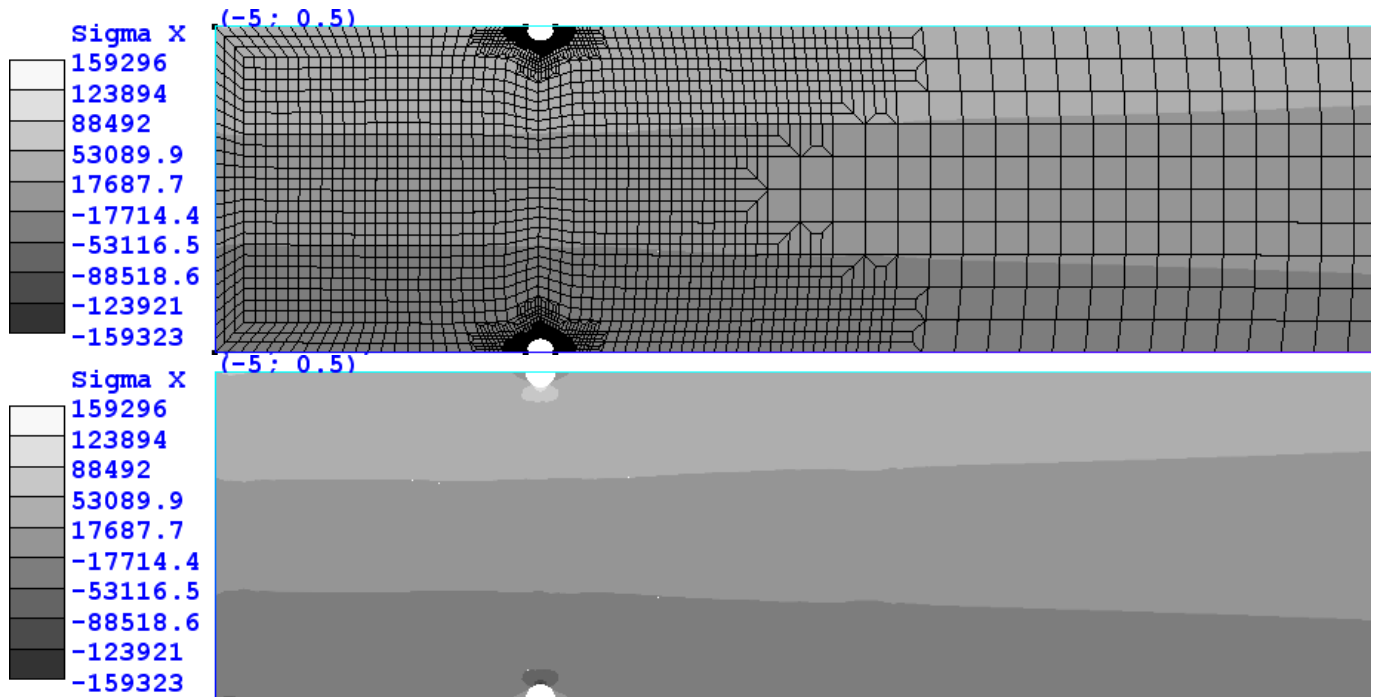


Рисунок 6.23 – Напруження σ_x у балці (адаптивна дискретна модель)

Отже, використання адаптивної генерації дискретних моделей форм об'єктів дозволяє підвищити точність математичного моделювання. Проте такий підхід є екстенсивним щодо використання обчислювальних ресурсів.

Висновки до розділу 6

Отже, основним науковим результатом цього розділу є новий метод генерації адаптивних дискретних моделей форм об'єктів, які задані функціонально. Запропонований метод на відміну від описаних у попередніх розділах аналізує результати методу скінченних елементів, а не тільки інформацію про форму об'єкту.

Як результат кількість вузлів і елементів збільшується у тих областях, де найбільше змінюється потрібна досліднику величина.

Розроблений метод генерації адаптивних дискретних моделей не залежить від форми елементів. Він використовує таблиці шаблонів розбиття елементів, внаслідок чого можна використовувати єдиний алгоритм для основних форм елементів.

Дослідження впливу форми скінченних елементів на точність моделювання дозволяє зробити висновок, що при використанні лінійних функцій форми симплекси (трикутники та тетраедри) є менш точними відносно топологічних кубиків (чотирикутників і шестигранників). Так розглянуті класичні задачі, для яких відомі аналітичні розв'язки, доводять, що при дослідженні напружено-деформованого стану обидві групи елементів майже однакові з точки зору обчислення переміщень, проте чотирикутники та шестигранники точніші при обчисленні напружень.

Результати шостого розділу опубліковано у роботах [103, 172, 175, 182].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилання [142, 204, 280, 396, 430, 503–505].

7 КАРКАС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Завдання, з якими зустрічається сучасне програмне забезпечення, все більше ускладнюються. В основу обчислювальних модулів закладаються різні підходи та методи. Як результат суттєво ускладнюються розробка та підтримка. Можливим способом спрощення розробки та підтримки такого програмного забезпечення є не тільки розробка універсальних узагальнених методів, а й типових програмних рішень, які можуть використовуватися повторно.

7.1 Шаблони проектування для об'єктно-орієнтованих САПР машинобудування

Шаблони проектування як загальні багаторазово використовувані рішення вперше були розроблені в кінці 1980х років і з того часу їх активно використовують в інженерії програмного забезпечення [190, 251, 431]. До недавнього часу розробники наукомісткого програмного забезпечення зазвичай уникали об'єктно-орієнтованого підходу через його накладні обчислювальні витрати [214]. Проте, програмне забезпечення САПР машинобудування стає більшим і складнішим. Виникають питання щодо його гнучкості, розширюваності та супроводу [241]. Шаблони програмного забезпечення розв'язують ці задачі шляхом надання загальних об'єктно-орієнтованих рішень.

Архітектори, механіки, винахідники та інженери для розв'язання інженерних задач використовують підсистеми інженерного аналізу (Computer-Aided Engineering, CAE) САПР машинобудування. Підсистеми інженерного аналізу є наукомісткими частинами САПР, які можуть включати модулі дослідження напружено-деформованого стану, поширення температури, динаміки твердих тіл, газів і рідин.

Загалом, підсистема інженерного аналізу має три підсистеми: препроцесор, процесор і постпроцесор результатів.

Препроцесор відповідає за визначення моделі задачі. Зазвичай модель задачі включає опис форми об'єкта, впливів зовнішнього середовища (наприклад, прикладених сил, температур, вібрацій). Форму досліджуваного об'єкта описують за допомогою однієї з схем подання. Під схемою подання розуміють семантично та синтаксично коректне відношення між множинами формальних описів і допустимих форм об'єктів.

Процесор є ядром підсистеми інженерного аналізу. Процесор – це множина чисельних методів, які зазвичай реалізуються з використанням високопродуктивних обчислень.

Можливо, найбільш поширеним чисельним методом є метод скінченних елементів. У цьому методі неперервна модель замінюється дискретною. Відповідно, перший крок такого моделювання потребує дискретизації неперервної геометричної області на елементи простої форми, наприклад, трикутники, чотирикутники, тетраедри та шестигранники.

Поширені альтернативи методу скінченних елементів (наприклад, безсіткові методи) можуть так само бути використані процесором. Ці методи не вимагають зв'язків між вузлами у моделі форми об'єкта. Проте, вузлова дискретна модель все ще необхідна. Отже, генерація дискретних моделей є важливою складовою, яка з'єднує опис форми об'єкта з його дискретною моделлю.

Після застосування чисельного аналізу на основі значень, отриманих у вузлах дискретної моделі, постпроцесор обчислює похідні величини та візуалізує результати. Тобто, постпроцесор створює подання дискретної моделі. Також значення, отримані у вузлах дискретної моделі, можна використовувати для поліпшення обох вихідної (неперервної) та проміжної (дискретної) моделей.

Отже, абстракціями верхнього рівня при розробці підсистем інженерного аналізу є препроцесор, процесор і постпроцесор. Розробка шаблонів проектування дозволить об'єднати ці абстракції в єдину об'єктно-орієнтовану модель.

7.1.1 Декомпозиція та ідентифікація

Зазвичай, підсистеми інженерного аналізу САПР машинобудування та їх базис чисельних методів розробляють з використанням процедурного підходу та таких мов програмування як FORTRAN, С тощо. Процедурний підхід вельми ефективний у реалізації чисельних методів, включаючи метод скінченних елементів. Проте, підтримка САПР машинобудування (як наукомісткого програмного забезпечення) включає оновлення чисельних моделей, додавання нової функціональності та підвищення продуктивності програмного кода.

Розглянемо підсистему САПР машинобудування з процесором, що ґрунтується на методі скінченних елементів. Можна припустити, що модель кожного об'єкта спочатку описана за допомогою однієї зі схем подання, а також побудована відповідна дискретна модель його форми.

Виконаємо декомпозицію САПР машинобудування з використанням аналізу загальностей та частковостей (commonality and variability analysis) [248]. Результати ідентифікації абстракцій (загальностей) та конкретних реалізацій (частковостей) наведені у табл. 7.1.

Таблиця 7.1 – Абстракції та реалізації САПР машинобудування

Абстракція	Конкретні реалізації
Схема подання (Representation scheme)	граничне подання (BRep); конструктивна блокова геометрія (CSG); функціональне подання (Frep) тощо;
Точка (Point)	двовимірна точка; тривимірна точка;
Вузол (Node)	двовимірний вузол; тривимірний вузол;
Елемент (Element)	сегмент; трикутник; чотирикутник; тетраедр; шестигранник;
Дискретна модель (Mesh)	двовимірна дискретна модель; тривимірна дискретна модель;
Двовимірна дискретна модель	двовимірна дискретна модель на базі трикутників; двовимірна модель на базі чотирикутників; двовимірна модель на базі сегментів;

Продовження таблиці 7.1

Абстракція	Конкретні реалізації
Тривимірна дискретна модель	дискретна модель на базі тетраедрів; дискретна модель на базі шестигранників; тривимірна дискретна модель на базі трикутників; тривимірна дискретна модель на базі чотирикутників;
Процесор МСЕ	процесор двовимірних задач; процесор тривимірних задач;
Процесор двовимірних задач	процесор двовимірних задач пружності; процесор задач дослідження пластин;
Процесор тривимірних задач	процесор тривимірних задач пружності; процесор дослідження оболонок;

7.1.2 Каталог шаблонів проектування

1. Шаблон проектування «UI–Model–Analysis»

Сучасні програмні підсистеми інженерного аналізу САПР машинобудування мають інтегрований графічний інтерфейс користувача (UI). Використовуючи контролери графічного інтерфейсу користувач вводить модель задачі (Model) та ініціює її аналіз процесором (Analysis). Отже, на верхньому рівні абстракції три основних пакета беруть участь у базовому шаблоні «UI–Model–Analysis» (рис. 7.1). Метою цього шаблону проектування є декомпозиція підсистеми інженерного аналізу на високорівневі частини, а також відділення класів, які пов'язані з графічним інтерфейсом від класів моделі та процесора.

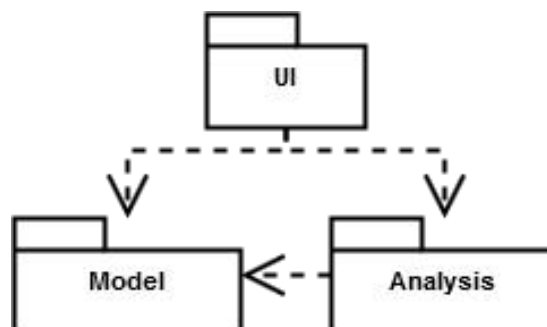


Рисунок 7.1 – Пакети шаблону проектування «UI–Model–Analysis»

Пакет UI (графічний інтерфейс) містить класи, які реалізують пов'язані з графічним інтерфейсом функції програмного забезпечення. Ці класи об'єднують функції пре- і постобробки (наприклад, вводу даних і візуалізацію). Отже, не повинно бути зв'язності між підсистемами аналізу та графічного інтерфейсу. Проте, пакет UI залежить від пакетів Model (препроцесор) і Analysis (процесор).

Загалом модель визначають за допомогою деякого проблемно-орієнтованої мови. Проблемно-орієнтована мова дозволяє описати форму об'єкта та параметри задачі (фізико-механічні характеристики, впливи зовнішнього середовища). Форму об'єкта описують на базі певної схеми подання. Абстрактна схема подання (Representation) є інтерфейсом, який дозволяє перевірити належність точки до моделі об'єкта. Отже, схема подання є однією з ключових абстракцій пакета Model.

Пакет Analysis (процесор) відповідає за чисельний аналіз. Процедури аналізу, зазвичай, виконують за допомогою метода скінченних елементів. У цьому методі два основних етапи: генерація дискретної моделі та безпосередньо скінченно-елементний аналіз. Отже, дискретна модель (Mesh) і процесор задач метода скінченних елементів (FEA Problem) є ключовими абстракціями цього пакета.

2. Шаблон проектування «Representation–Mesh»

Припустимо, що форма об'єкта описана у термінах деякої схеми подання. Найбільш поширеними схемами подання є граничне подання (BRep), конструктивна блокова геометрія (CSG) і функціональне подання (FRep). Спільною властивістю конструктивної блокової геометрії та функціонального подання є можливість легкої перевірки приналежності точки до моделі об'єкта. Водночас (як це показано у розділі 2) граничне подання загалом потребує перевірки парності перетинів променя з точки і межі моделі об'єкта. Тим не менш, спільною поведінкою відповідних класів можна вважати класифікацію точок щодо межі об'єкта.

Для генерації дискретної моделі застосовуються певні методи розбиття частини евклідового простору на елементи простої форми. Можна припустити, що дискретна модель – це абстрактний інтерфейс (Mesh), який дозволяє будувати та ітерувати через колекцію елементів. У такому випадку конкретні класи будуть

похідними від абстрактного класу *Mesh* (рис. 7.2). Конкретні класи генерують елементи певної форми, використовуючи інтерфейс абстракції схеми подання (*Representation*) для класифікації точок простору. Обидва класи *Representation* і *Mesh* беруть участь у шаблоні проектування «*Representation–Mesh*». Метою цього шаблону проектування є розподіл обов’язків між класами подання форм об’єктів і генерації дискретних моделей.

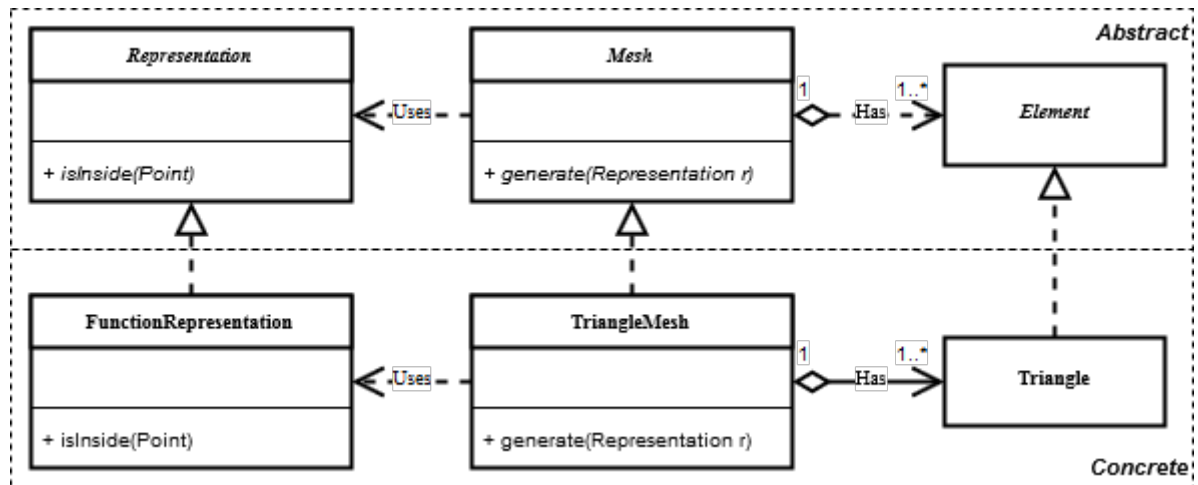


Рисунок 7.2 – Шаблон проектування «*Representation–Mesh*»

Обидва класи *Representation* і *Mesh* можуть бути окремими випадками шаблону проектування «Стратегія» (Strategy [251]). Клас *Mesh* та його похідні можна реалізувати як шаблон проектування «Ітератор» (Iterator [251]) для організації перегляду колекцій вузлів і елементів. Водночас деякі методи генерації дискретних моделей форм об’єктів (наприклад, розглянуті у розділах 4 і 5 узагальнені методи) можна реалізувати як «Шаблонний метод» (Template Method [251]).

3. Шаблон проектування «*Element–Node*»

Загалом елемент дискретної моделі (*Element*) – це впорядкована колекція вузлів (*Node*). У обох (двовимірному та тривимірному) випадках елементи мають ребра. Ребро (*Edge*) – це відрізок прямої, що з’єднує два вузли. Тривимірні елементи на додаток до ребер мають грані. Грань (*Face*) – це плоский елемент, обмежений ребрами. Тобто, конкретна грань – це об’єкт класу, який реалізує інтерфейс *Element*. Отже, *Element*, *Face*, *Edge* і *Node* – структурні частини шаблону проектування

«Element–Node» (рис. 7.3). Метою цього шаблону проектування є розробка об'єктної декомпозиції для визначення елементів.

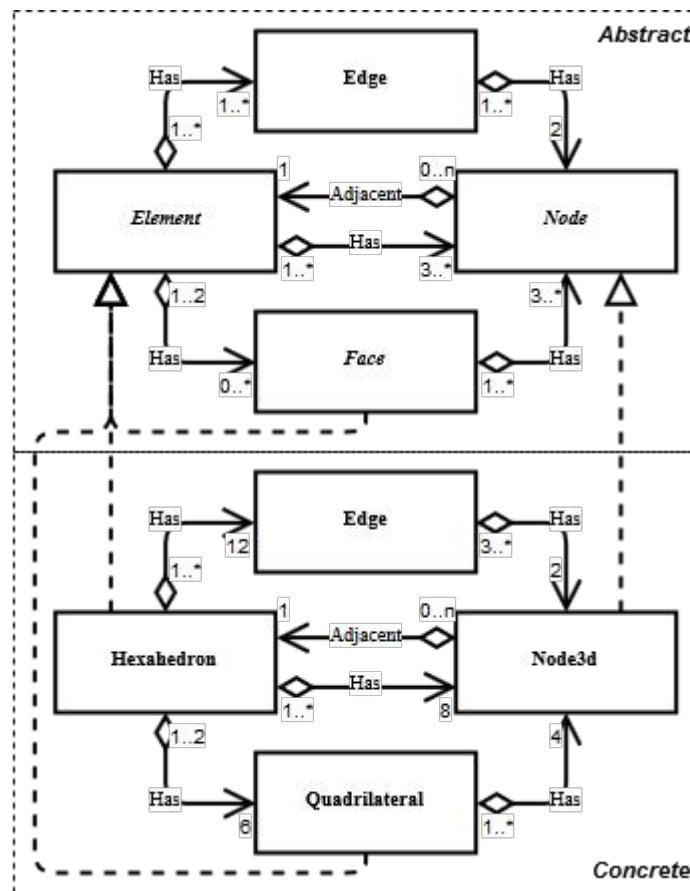


Рисунок 7.3 – Шаблон проектування «Element–Node»

Клас Node – це абстрактний інтерфейс, який визначає спільну для всіх типів вузлів поведінку (наприклад, маніпуляція типами вузлів, управління множиною інцидентних елементів тощо). Можуть бути різні реалізації для конкретних класів вузлів. Наприклад, операції зберігання координат і геометричні операції над ними можна реалізувати безпосередньо в класі. Альтернативно, клас вузла може бути композицією об'єктів. У такому випадку будуть отримані паралельні ієрархії для класів, що реалізують операції над вузлами і векторами.

Клас Element і його похідні (включно грані) можна реалізувати з використанням шаблону проектування «Ітератор» для організації проходження по колекціям вузлів, ребер і граней. Шаблон «Ітератор» також можна використати для реалізації роботи зі списками інцидентних елементів.

4. Шаблон проектування «FEA Problem»

Спочатку метод скінченних елементів був розроблений для дослідження напружено-деформованого стану, а потім застосований у розв'язанні інших задач, що ґрунтуються на рівняннях у частинних похідних. У інженерних задачах, зазвичай, необхідно визначити розподіл значень деякої невідомої величини (наприклад, переміщень у задачах дослідження напружено-деформованого стану). У методі скінченних елементів розподіл значень невідомої величини апроксимують кусково-лінійними функціями на кожному елементі. Як результат отримують систему лінійних алгебраїчних рівнянь, яка дозволяє обчислити значення невідомої величини.

З математичної точки зору кожен елемент можна подати матрицями та векторами. Операції ансамблювання дозволяють трансформувати значення з локальних матриць і векторів у глобальні. Вихідною інформацією у процесі анасамблювання є номери вузлів і зв'язки між ними.

Перед розв'язанням системи лінійних алгебраїчних рівнянь повинні бути враховані крайові умови. Вони спрощують глобальні матрицю і вектор шляхом виключення деяких невідомих. Якщо крайова умова застосована до деякого вузла, то виключаються рядок і стовпець глобальної матриці, які визначені глобальним номером вузла і напрямком крайової умови.

Локальні та глобальні вектори обчислюються при врахуванні сил, що діють в вузлах елементів. Аналогічно крайовим умовам, конкретна позиція вузлового навантаження обчислюється на базі глобального номера вузла та напрямку, в якому це навантаження діє.

Шаблон проектування «FEA Problem» можна отримати з шаблону «Шаблонний метод» [251] шляхом додавання дискретної моделі, крайових умов і сил (рис. 7.4). Метою цього шаблону проектування є визначення структури алгоритму скінченно-елементного аналізу та об'єктної композиції для крайових умов та сил, які беруть участь в задачі.

Сили та крайові умови реалізують інтерфейс FeaValue. Цей інтерфейс дозволяє отримати напрям дії та значення в довільній точці. Крайові умови та сили зазвичай визначаються користувачем з використанням графічного інтерфейсу або проблемно-

орієнтованої мови. Проте, інтерфейс *FeaValue* можна реалізувати за допомогою шаблону проектування «Адаптер» (Adapter [251]).

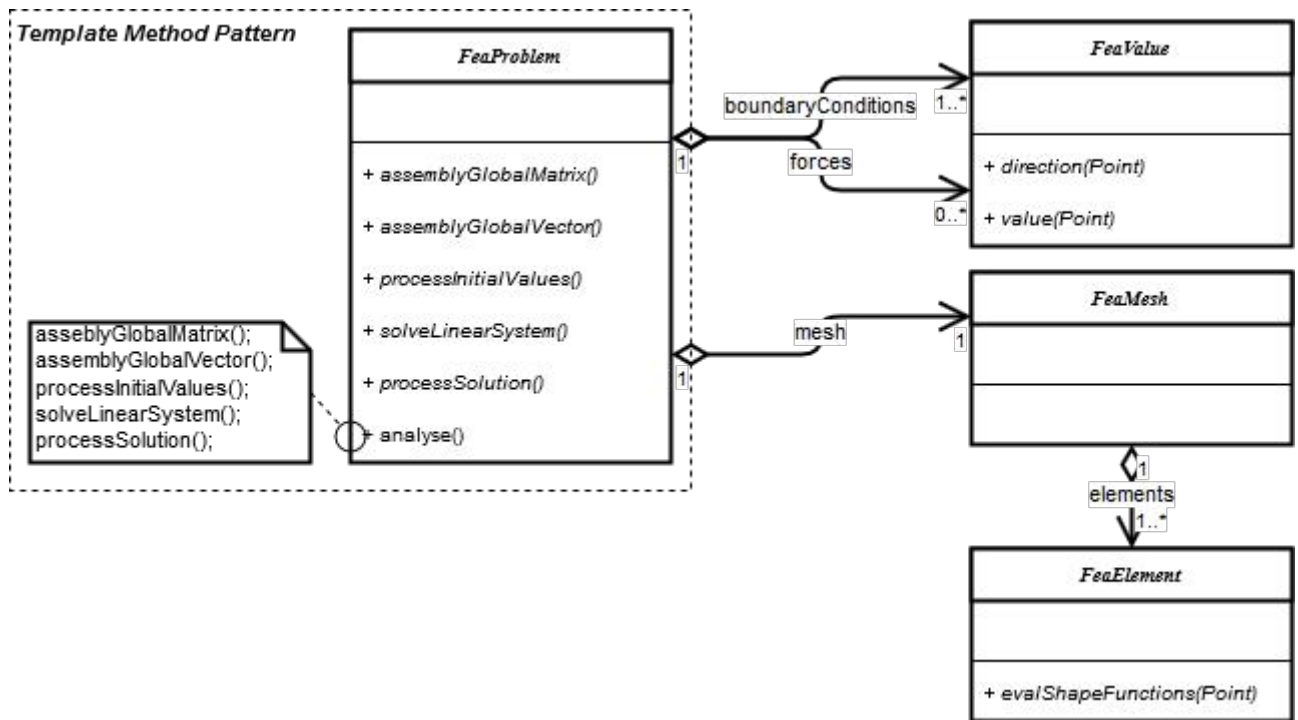


Рисунок 7.4 – Шаблон проектування «FEA Problem»

Важливим кроком процесу ансамблювання є обчислення інтерполяційних функцій (функцій форми) для кожного елемента. Зазвичай як функції форми використовують поліноми. Степінь полінома залежить від кількості вузлів у елементі. Найбільш поширеними є лінійні, білінійні та трілінійні. Поліноми більш високих степенів можна отримати додаванням додаткових вузлів у елементи. Отже, класи *FeaMesh* і *FeaElement*, основним обов'язком яких є обчислення функції форми, можна отримати із застосуванням шаблону проектування «Адаптер» до відповідних класів, що описані вище.

Розроблені вище шаблони основних логічних елементів програмної підсистеми інженерного аналізу САПР машинобудування пропонують схеми об'єктної декомпозиції та розподілу обов'язків без обмежень на реалізацію. Розподіл обов'язків між відносно незалежними блоками дозволяє спростити колективну розробку та підтримку САПР як складного програмного продукту.

7.2 Методи генерації та обробки дискретних моделей форм об'єктів на базі технологій паралельних обчислень

Паралельні обчислення є однією з базових комп'ютерних концепцій. Ідея одночасного використання декількох паралельних процесів для вирішення однієї задачі подібна до ідеї колективної праці для досягнення спільної мети. Паралельні обчислення відносяться до засобів зменшення часу, необхідного на виконання завдання. Отже, актуальною є розробка методів, що використовують паралельні обчислення.

7.2.1 Метод генерації дискретних моделей форм об'єктів з використанням паралельних обчислень

Для складних форм об'єктів генерація дискретних моделей може займати досить тривалий час. Можливим способом зменшення затраченого часу є залучення технологій паралельних обчислень.

Нехай у комп'ютерній системі є k функціональних пристроїв (процесорів, обчислювальних ядер тощо). У такому випадку фонову сітку, яка використовується як вихідна в узагальненому методі генерації дискретних моделей форм об'єктів, можна обробляти паралельно доступними пристроями. Для цього розіб'ємо фонову сітку на k підмножин. Таке розбиття можна здійснювати різними способами: наприклад, розбивши область на m горизонтальних або вертикальних смуг, або рекурсивно розбивши на k квадрантів (загалом, розбиття може залежати від щільності елементів фонові сітки). Потім до кожної підмножини можна одночасно застосувати дії, описані в узагальненому методі генерації дискретних моделей форм об'єктів. Критичними секціями будуть додавання нових вузлів і елементів у

дискретну модель. Отже, алгоритм паралельної генерації дискретної моделі набуде вигляду наступної послідовності кроків.

1. Генерація фонові дискретної моделі $\mathbf{M}$ і розбиття її на k підмножин $\mathbf{M}_i, i \in 1..k$.
2. Паралельно: для кожної з підмножин $\mathbf{M}_i, (i \in 1..k)$ видалення зовнішніх вузлів і елементів.
3. Послідовно: синхронізація підмножин для створення початкової дискретної моделі $\mathbf{M} = \mathbf{M}_1 \cup \mathbf{M}_2 \dots \mathbf{M}_k$ і видалення конфігурацій елементів, які можуть обумовлювати неоднозначність.
4. Послідовно: розбиття множини вузлів $\mathbf{N}$ дискретної моделі $\mathbf{M}$ на k рівних за розміром підмножин $\mathbf{N}_i, i \in 1..k$ ($\mathbf{M} \cdot \mathbf{N} = \mathbf{N}_1 \cup \mathbf{N}_2 \cup \dots \cup \mathbf{N}_k: \forall i \neq j \mathbf{N}_i \cap \mathbf{N}_j = \emptyset$).
5. Паралельно: для кожного вузла, який належить $\mathbf{N}_i$ знайти перетин з границею моделі форми об'єкта.
6. Паралельно: для кожного $\delta(\mathbf{M}_i)$ сформувати межовий шар елементів.

Запропонований алгоритм реалізований у вигляді програми на мові C++ з використанням бібліотеки OpenMP для організації паралельних обчислень. Як тестова платформа використаний комп'ютер з процесором Intel Core i5-3330 (3.0GHz, 4 ядра) і 8 гігабайтами оперативної пам'яті, який знаходиться під управлінням операційною системою LinuxMint 17.1 (64 біти, ядро Linux 3.13, компілятор gcc 4.8.2). У тестових прикладах використано рівномірні фонові дискретні моделі на базі трикутних елементів.

Графіки залежності часу генерації дискретної моделі планки (рис. 2.8), яка задана формулою (2.18), від кількості вузлів у фоновій дискретній моделі наведені на рис. 7.5.

Як видно з рис. 7.5, якщо у фоновій дискретній моделі міститься 2000 вузлів, то час паралельної генерації з використанням двох потоків практично збігається з послідовним, а час, що використаний чотирма потоками, удвічі більший. Починаючи з 10000 вузлів генерації дискретної моделі з використанням чотирьох паралельних потоків стає швидшим за послідовну генерацію, а починаючи з 16000 – за генерацію двома паралельними потоками. З подальшим збільшенням кількості вузлів різниця у

витраченому часі прагне до теоретичної оцінки. Ці результати обумовлені існуванням накладних обчислювальних витрат на створення та управління паралельними потоками.

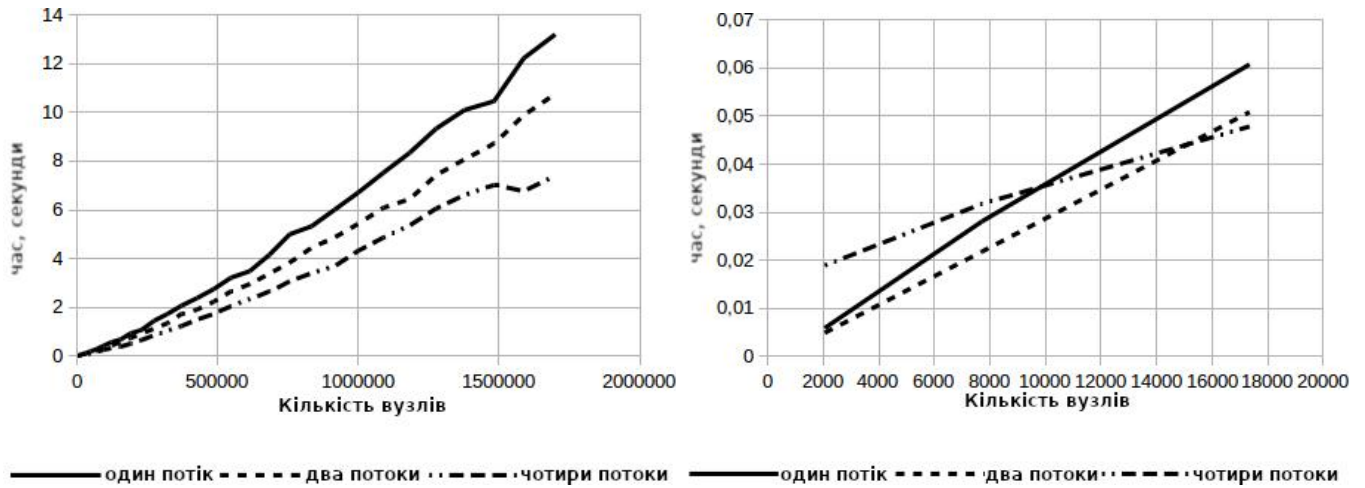


Рисунок 7.5 – Модель планки: залежність часу, витраченого на генерацію дискретної моделі від кількості вузлів у фоновій сітці

Чим складніша модель області, тим за меншої кількості вузлів паралельний алгоритм буде швидшим ніж послідовний. Наприклад, при побудові дискретної моделі прокладки (рис. 2.9) чотири паралельні потоки будуть виконуватися швидше послідовного при 6000 вузлів і двох паралельних потоків при 8500 (рис. 7.6).

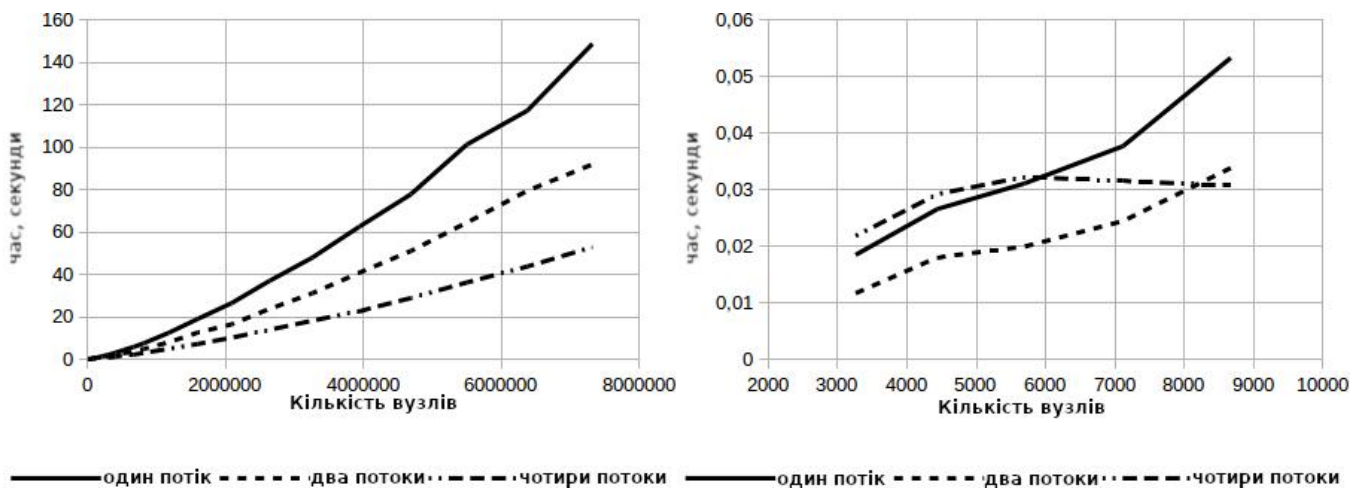


Рисунок 7.6 – Модель прокладки: залежність часу, витраченого на генерацію дискретної моделі від кількості вузлів у фоновій сітці

7.2.2 Скінченно-елементний аналіз з використанням технологій паралельних обчислень

Проектування сучасних деталей та конструкцій пов'язане з необхідністю використання чисельних методів для розв'язання задач високої розмірності. Врахування особливостей багатьох деталей і конструкцій обумовлює виникнення на підготовчих етапах складних моделей їх форми. Методи генерації дискретних моделей у результаті особливостей форми, умов задачі (наприклад, контакту, тріщин, багат шаровості тощо) та точності апроксимації може спричиняти велику кількість скінченних елементів. У підсумку для моделювання необхідні значні комп'ютерні ресурси та, як наслідок, часові витрати.

Природним рішенням задачі мінімізації потрібних комп'ютерних ресурсів є оптимізація кількості вузлів і елементів у дискретних моделях форм об'єктів. Проте, у випадках, коли зменшення кількості скінченних елементів зумовлює невиправдану втрату точності моделі, одним з шляхів оптимізації часових витрат є використання паралельних обчислень.

У методі скінченних елементів, якщо застосовують достатньо складні дискретні моделі форм об'єктів, етапи побудови глобальної матриці жорсткості та розв'язання системи лінійних алгебраїчних рівнянь є найбільш вимогливими щодо обчислювальних ресурсів. Отже, використання паралельних обчислень при виконанні цих етапів дозволить отримати найбільш вагомий результат з точки зору зменшення часу проектування та моделювання.

Першим кроком після етапів завантаження й ініціалізації вихідних даних паралельної реалізації метода скінченних елементів є визначення величини K – кількості наявних функціональних пристроїв (рис. 7.7). Ця величина є основою для обчислення кількості паралельних потоків. В мультипроцесорних системах при обчисленні глобальної матриці жорсткості операції вводу-виводу використовують тільки, якщо недостатньо основної пам'яті для зберігання вихідних даних або самої глобальної матриці. Тобто, за умови використання стисненого способу збереження

глобальної матриці (яка, зазвичай, є розрідженою), час очікування операцій вводу-виводу буде мінімальним. Отже, на практиці величина K також визначає кількість паралельних комп'ютерних потоків.

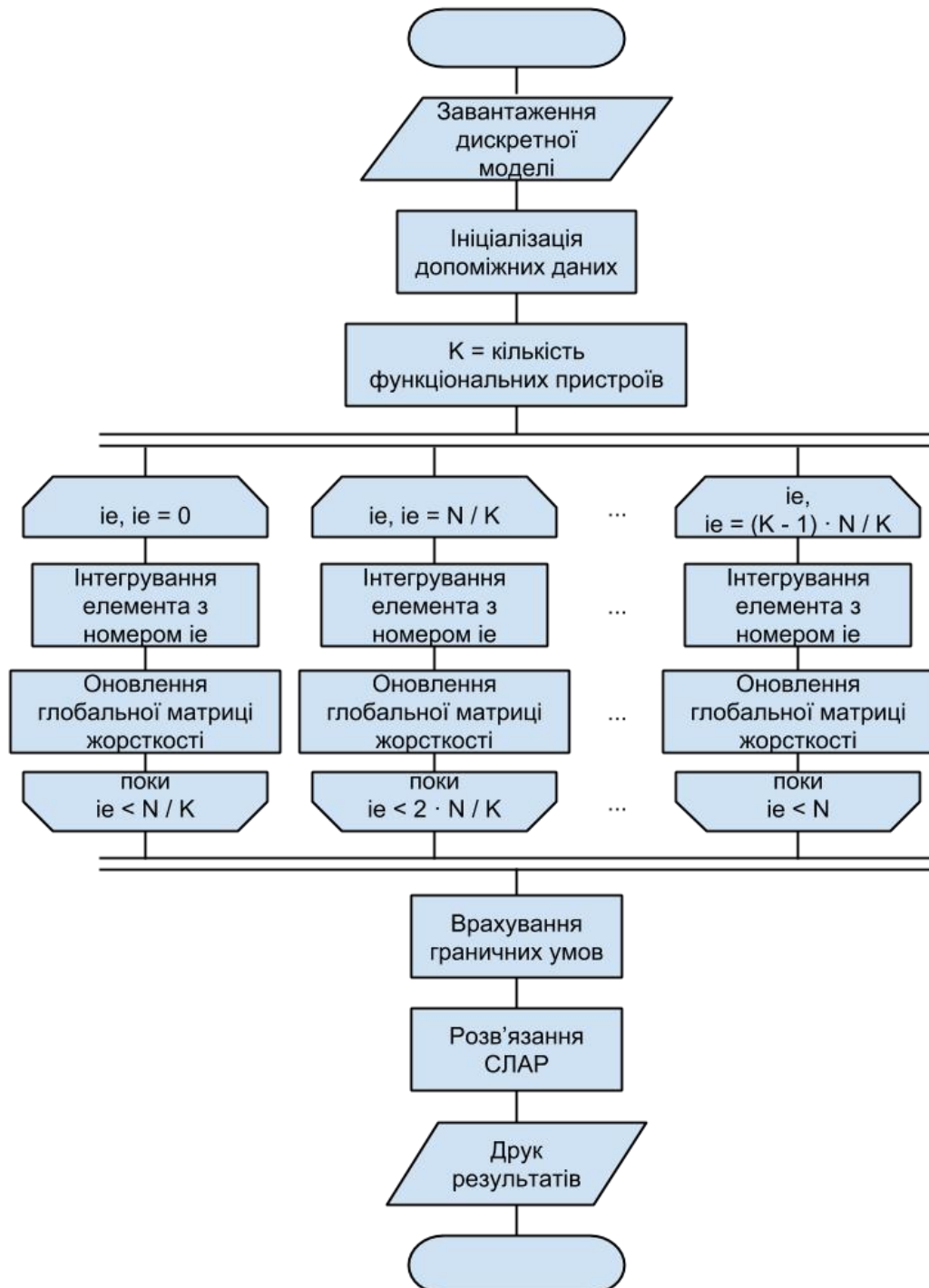


Рисунок 7.7 – Схема метода скінченних елементів на базі паралельних обчислень

Ансамблювання є додаванням за спеціальними правилами, отже, кожен з K паралельних потоків повинен виконувати ансамблювання N / K елементів (N –

кількість елементів у дискретній моделі). Водночас всі дані (включно змінні циклів), які пов'язані з інтегруванням на скінченному елементі, повинні бути локальними для паралельних потоків (бути доступними тільки для потоку, що володіє; блок-схема зображена на рис. 7.7).

Крок оновлення глобальної матриці жорсткості, враховуючи стиснений формат збереження даних, є критичним з точки зору безпеки використання паралельних потоків. Отже, доступ іншим потокам до глобальної матриці жорсткості необхідно блокувати, поки один потік модифікує її дані.

Час, який необхідний для побудови глобальної матриці жорсткості за допомогою паралельного обчислювального процесу, можна оцінити величиною

$$T_p = N \left(\frac{I}{K} + m \right) + K t_{thread},$$

де I – процесорний час, який необхідний для інтегрування вихідного диференціального рівняння на одному скінченному елементі;

m – процесорний час, який необхідний для ансамблювання результатів інтегрування на одному елементі:

t_{thread} – процесорний час, який необхідний для ініціалізації одного комп'ютерного потоку.

Аналогічно, час, який необхідний для побудови глобальної матриці жорсткості на базі послідовного алгоритму, можна оцінити величиною

$$T_s = N(I + m).$$

Застосування паралельних обчислень має сенс, якщо $T_s > T_p$:

$$N(I + m) > N \left(\frac{I}{K} + m \right) + K t_{thread},$$

у результаті спрощення якої отримаємо таку оцінку:

$$NI > \frac{K^2}{K-1} t_{thread}.$$

Отже, наприклад, для двох паралельних потоків, добуток кількості елементів на час інтегрування одного елемента має бути вчетверо більшим часу ініціалізації паралельних потоків, що може не виконуватися при малій кількості скінченних елементів.

На практиці для розв'язання отриманих в результаті побудови глобальної матриці жорсткості СЛАР, зазвичай, використовують ітераційні методи (наприклад, метод спряжених градієнтів [320, 378] та інші). Їх ключовою особливістю є те, що результати поточної ітерації є вихідними даними для наступної. Отже, для скорочення часу пошуку розв'язку необхідно прискорювати внутрішні операції методу: добуток матриці на вектор, поелементний добуток векторів, обчислення нев'язки тощо.

З урахуванням того, що в результаті ансамблювання глобальна матриця жорсткості є розрідженою, схема обчислення добутку матриці на вектор в K паралельних потоків може отримати вигляд, наведений на рис. 7.8. Спільними для потоків даними є: матриця A , розмір якої $D \times D$, вектор x , вектор-результат r і число K . Оскільки обчислювальними потоками оновлюються різні елементи вектора r , то відсутня необхідність у блокуванні при синхронізації потоків. Лічильники циклів i та j , змінна t – локальні для кожного обчислювального потоку. У внутрішніх циклах по j обчислення виконуємо тільки для ненульових елементів i -го рядка матриці A .

Як тестовий стенд використано програмно-апаратну конфігурацію з процесором Intel Core i3-380 (2.53 ГГц, 2 фізичних ядра + 2 віртуальних ядра) з 3 ГБ оперативної пам'яті під управлінням операційної системи openSUSE 12.2 (компілятор gcc 4.7), для маніпуляцій паралельними потоками – бібліотеки стандарту openMP.

Для оцінки часу кожен тестовий приклад послідовно запускаємо три рази та фіксуємо найменший витрачений час.

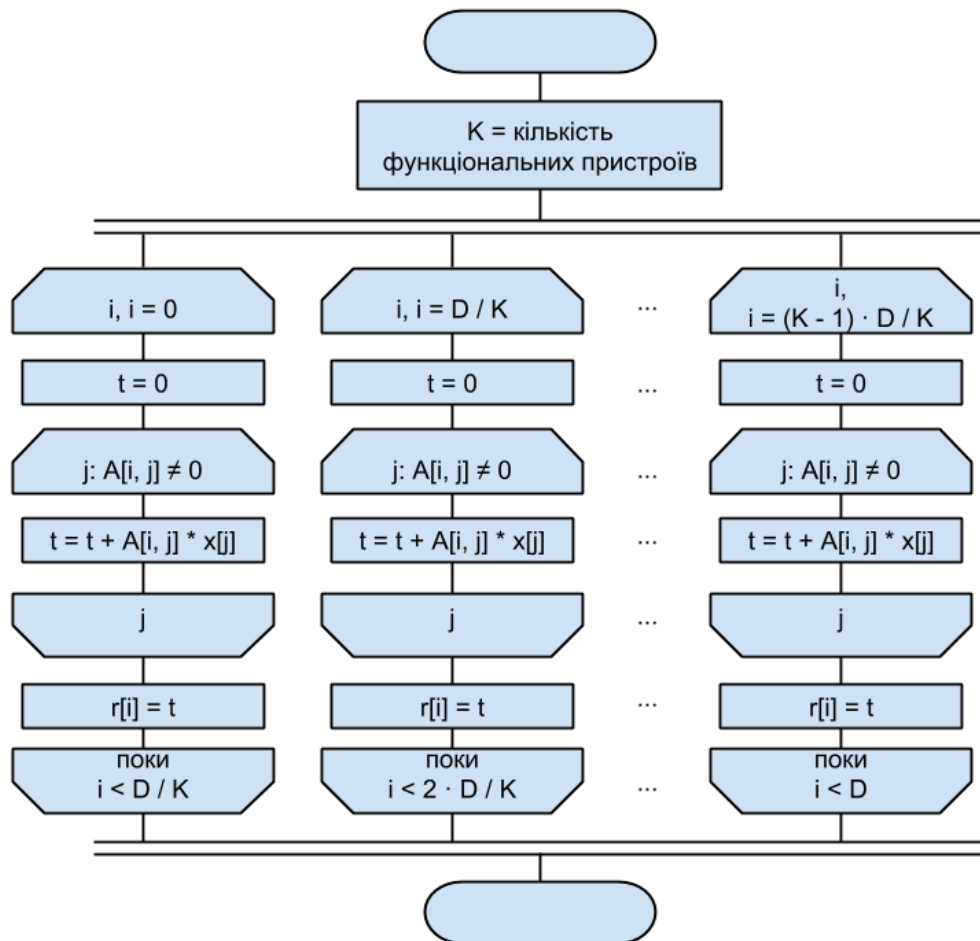


Рисунок 7.8 – Схема паралельного добутку розрідженої матриці та вектора

1. Задача Діріхле

Розглянемо розв'язок рівняння Лапласа з умовами Діріхле:

$$\frac{\partial U}{\partial x^2} + \frac{\partial U}{\partial y^2} + \frac{\partial U}{\partial z^2} = 0,$$

$$U_{\delta\Omega} = f(x, y, z).$$

Як область Ω задане зубчасте колесо (дискретна модель: 49500 вузлів, 40920 шестигранних скінченних елементів, рис. 7.9). Функція $f(x, y, z)$ має такий вигляд:

$$f(x, y, z) = \begin{cases} 0, & \text{на поверхні центрального отвору,} \\ 4 \sin^2(5\varphi), & \text{на поверхні зубців,} \end{cases}$$

де φ – кут повороту полярного радіуса до відповідної точки у площині Oxy .

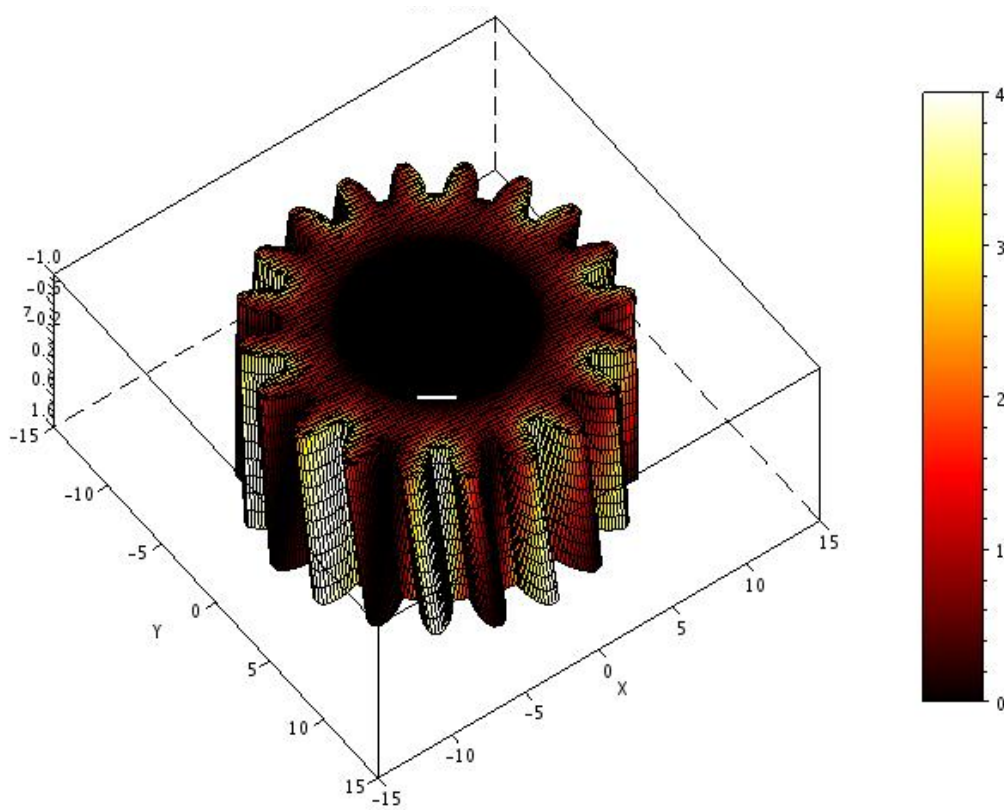


Рисунок 7.9 – Задача Діріхле: зубчасте колесо

Формула для обчислення локальної матриці $[K_e]$ отримає такий вигляд:

$$[K_e] = \iiint_{\Omega_e} \left(\begin{pmatrix} \frac{\partial N_1}{\partial x} \\ \dots \\ \frac{\partial N_8}{\partial x} \end{pmatrix} \left\{ \frac{\partial N_1}{\partial x} \quad \dots \quad \frac{\partial N_8}{\partial x} \right\} + \begin{pmatrix} \frac{\partial N_1}{\partial y} \\ \dots \\ \frac{\partial N_8}{\partial y} \end{pmatrix} \left\{ \frac{\partial N_1}{\partial y} \quad \dots \quad \frac{\partial N_8}{\partial y} \right\} + \right. \\ \left. + \begin{pmatrix} \frac{\partial N_1}{\partial z} \\ \dots \\ \frac{\partial N_8}{\partial z} \end{pmatrix} \left\{ \frac{\partial N_1}{\partial z} \quad \dots \quad \frac{\partial N_8}{\partial z} \right\} \right) dx dy dz,$$

де N_i ($i = 1, \dots, 8$) – функції форми шестигранного скінченного елемента.

За результатами обчислювального експерименту отримано результати, що наведено в табл. 7.2 (час у секундах).

Таблиця 7.2 – Розв’язання задачі Діріхле

Етап	Кількість потоків		
	1	2	4
	час, с		
Побудова глобальної матриці жорсткості	3,07639	1,56709	1,25949
Розв’язання СЛАР, 73 ітерації	0,304023	0,240893	0,264904

2. Дослідження прогину тришарової круглої пластинки, яка знаходиться під дією рівномірного навантаження (мікромодель)

Дана затиснена по контуру тришарова кругла пластинка, радіус якої 0,4 м, товщина – 0,01 м. Внутрішній шар: товщина – 0,008 м, модуль Юнга – $E_i = 72017,3327$ МПа, коефіцієнт Пуассона – $\nu_i = 0,2999518536$. Зовнішні шари: товщина – 0,001 м, модуль Юнга – $E_o = 203200$ МПа, коефіцієнт Пуассона – $\nu_o = 0,27$. На на поверхню пластинки діє рівномірне поперечне навантаження $q = -0,05$ МПа.

Для дослідження побудована тривимірна дискретна модель сектора пластинки (рис. 7.10) на базі десяти шарів шестигранних скінченних елементів (59213 вузлів і 52240 елементів), товщина яких 0,001 м. Межі шарів матеріалу апроксимовані ребрами та гранями скінченних елементів.

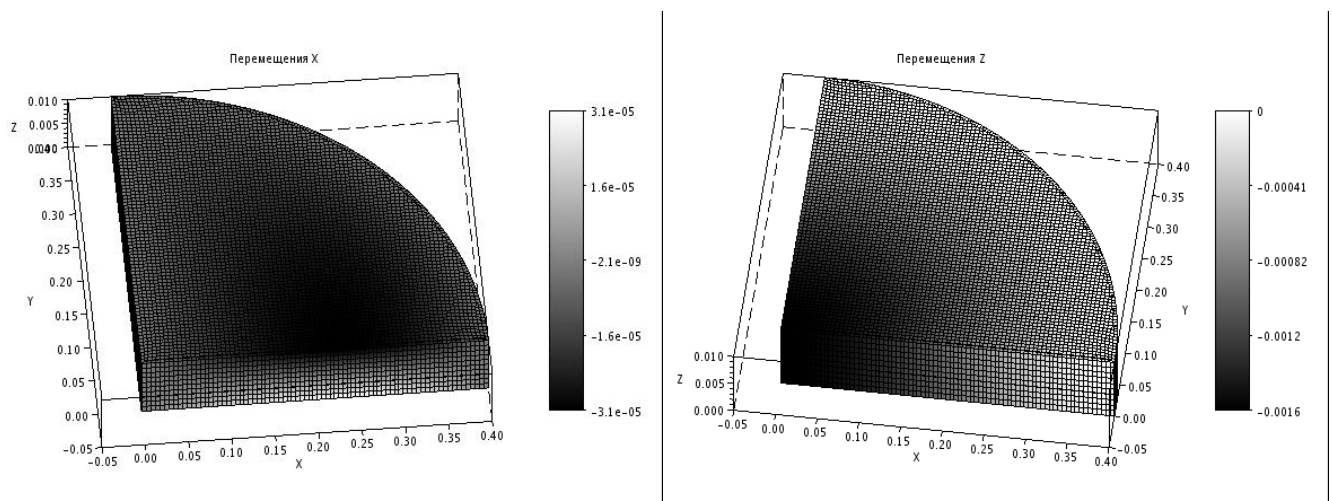


Рисунок 7.10 – Прогин тришарової пластинки

При обчисленні матриці $[D]$ у формулі (6.5) для врахування властивостей матеріалів шарів вважатимемо, що

$$E = \begin{cases} E_o, & z \geq 0,009 \vee z \leq 0,001, \\ E_i, & 0,001 \leq z \leq 0,009; \end{cases} \quad v = \begin{cases} v_o, & z \geq 0,009 \vee z \leq 0,001, \\ v_i, & 0,001 \leq z \leq 0,009. \end{cases}$$

Як результат обчислювального експерименту отримано результати, що наведені у табл. 7.3 (час у секундах).

Таблиця 7.3 – Дослідження пластинки

Етап	Кількість потоків		
	1	2	4
	час, с		
Побудова глобальної матриці жорсткості	83,7293	41,5063	31,7687
Розв'язання СЛАР, 6952 ітерації	298,287	191,299	189,134

Отже, обчислювальні експерименти показали, що використання паралельних обчислень в методі скінченних елементів дозволяє скоротити час на проектування та моделювання деталей та конструкцій. З таблиць 7.2 і 7.3 видно несуттєва перевага у часі при використанні чотирьох потоків відносно двох, що зумовлено двома віртуальними ядрами (які не мають відповідну фізичним ядрам швидкодію).

7.3 Проблемно-орієнтована мова моделювання форм об'єктів

Досить поширеним шляхом вирішення проблеми формалізації вихідних даних у системах інженерного аналізу є розробка лінгвістичного забезпечення – вхідних мов. Вхідна мова повинна включати в себе дві частини: мову опису об'єктів та мову опису завдань [99].

Мова опису об'єктів, в контексті цього дослідження, повинна надавати засоби опису моделей форм об'єктів, а також їх фізико-механічних характеристик і зовнішніх впливів (наприклад, прикладених сил).

Мова опису завдань повинна включати засоби виконання певних проектних операцій та процедур: побудови дискретних моделей та чисельного аналізу параметрів стану об'єкта. Прикладами таких параметрів можуть бути умови зупинки алгоритмів.

7.3.1 Базовий синтаксис

Можливим розв'язком задачі створення лінгвістичного забезпечення для інженерного аналізу в САПР є розширення синтаксису, визначеного стандартом певної універсальної мови програмування, шляхом доповнення його специфічними для проблемної області мовними конструкціями. Водночас фактично буде створено нову проблемно-орієнтовану мову. Використання як базового стандарту, на якому ґрунтуються популярні мови програмування, дозволяє спростити вивчення синтаксису проблемно-орієнтованої мови користувачами САПР. У даній роботі запропонована мова моделювання, що базується на стандарті ЕСМА-262 [257]. В основі стандарту ЕСМА покладено кілька вихідних технологій, найбільш відомою з яких є JavaScript. Серед його переваг можна виділити динамічну типізацію, яка дозволяє істотно скоротити код моделей.

Стандарт ЕСМА визначає специфікації об'єктно-орієнтованої мови для прикладних обчислень і маніпулювання обчислювальними об'єктами хост-оточення. Базовий синтаксис такої мови можна подати за допомогою розширеної форми Бекуса-Наура (рис.7.11). На рівні базового синтаксису програма є набором приписів і декларацій функцій, за допомогою яких можна визначати форми об'єктів і параметри розрахунку. Як алфавіт для ідентифікаторів можна використовувати латинські літери. Рядкові константи можна складати з довільного набору символів таблиці Unicode, які обмежені подвійними або одинарними лапками.

Згідно зі специфікацією, наведеною на рис. 7.11, кожний припис може бути блоком приписів, оголошенням змінних, виразом (простим, умовним, перемиканням, поверненням або циклічним, зупинки ітерації або циклу), пустим виразом або

пробілом. Якщо припис не є блоком, то він має закінчуватися символом вертикальної крапки з комою. Блоком приписів вважається деяка множина приписів, об'єднаних символами відкритої та закритої фігурних дужок.

```

програма = { припис | декларація функції } ;
припис = блок | пустий вираз | ( оголошення змінних, ';' ) | ( вираз, ';' ) | умовний
    вираз | циклічний вираз | стоп ітерація | стоп цикл | вираз повернення | вираз
    перемикавання | пробіл ;
пустий вираз = ';' ;
блок = '{', { припис }, '}' ;
оголошення змінних = 'var', пробіл, ідентифікатор, [ '=', вираз ], { ',', ідентифікатор, [
    '=', вираз ] } ;
вираз = виклик функції | декларація функції | ( ідентифікатор, присвоєння,
    ( обчислювальний вираз | виклик функції ) ) | створення об'єкта | інкремент |
    декремент ;
умовний вираз = 'if', '(', вираз, ')', припис, { 'else', припис } ;
циклічний вираз = ( 'do', припис, 'while', '(', вираз, ')', ';' ) |
    ( 'while', '(', вираз, ')', припис ) |
    ( 'for', '(', ( вираз, ';', вираз, ';', вираз ) | ( ініціалізація змінної, ';',
        вираз, ';' , вираз ) | ( вираз, 'in', вираз ) | ( ініціалізація змінної, 'in' вираз ),
        ')', припис ) ;
стоп ітерація = 'continue', ';' ;
стоп цикл = 'break', ';' ;
вираз повернення = 'return', { вираз }, ';' ;
вираз перемикавання = 'switch', '(', вираз, ')', '{'
    { 'case', вираз, ':', [ припис ], [ 'default', ':', [ припис ] ], '}' ;
обчислювальний вираз = простий вираз | умовний вираз | логічний вираз ;
умовний вираз = '(', вираз, ')', '?', вираз, ':', вираз ;
ініціалізація змінної = 'var', пробіл, ідентифікатор, '=', обчислювальний вираз ;
присвоєння = '=' | '+=' | '-=' | '*=' | '/=' ;
створення об'єкта = 'new', пробіл, ідентифікатор, ( вираз ) ;
декларація функції = 'function', { пробіл, ідентифікатор, '(', [ ідентифікатор, [ ',',
    ідентифікатор ] ], ')', предписание ;
виклик функції = ідентифікатор, '(', [ вираз, [ ',', вираз ] ], ')' ;
інкремент = ('++', ідентифікатор) | (ідентифікатор, '++') ;
декремент = ('--', ідентифікатор) | (ідентифікатор, '--') ;
логічний вираз = лог_і [ '{' | '}', лог_і ] ;
лог_і = простий вираз, [ '&&', простий вираз ] ;
простий вираз = арифметичний вираз, [ оператор відношення, арифметичний вираз ] ;
оператор відношення = '<' | '<=' | '>' | '>=' | '==' | '!=' | '===' | '!==' ;
арифметичний вираз = [ '+' | '-' ] терм { ('+' | '-') , терм } ;
терм = фактор { ('*' | '/' | '%'), фактор } ;
фактор = ідентифікатор | константа | рядок | вираз | ('!', фактор) ;
константа = [ '+' | '-' ] число [ '.' [число] ] [ ('e' | 'E') [ '+' | '-' ] число ] ;
число = цифра { цифра } ;
цифра = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' ;
ідентифікатор = буква { буква | цифра | '_' } ;
рядок = ('"' , { буква | цифра | символ }, '"') | (''' , { буква | цифра | символ } ,
    ''') ;

```

Рисунок 7.11 – РБНФ базового синтаксису

Всі змінні та функції при такому підході розглядаються як об'єкти, для яких можна визначити поля і методи зі заданими областями видимості. Оголошення змінних починається з ключового слова `var`, за яким після пробільного символу вказуємо список змінних, розділених комами, з необов'язковою ініціалізацією через знак рівності. Якщо змінна є об'єктом, то вона обов'язково створюється з

використанням оператора `new`. Для зберігання чисел використовується 64-бітовий формат числа з рухомою комою IEEE 754 [301]. Рядкові значення зберігаються в форматі UTF-16.

Умовний вираз декларується з використанням оператора `if`, за яким в круглих дужках описується вираз, що визначає логічну гілку. Потім вказується припис, відповідний логічній гілці “істина”. Далі після оператора `else` можна зазначити припис, відповідний логічній гілці “хибність”.

Можна побачити, що можливе використання циклічних виразів з передумовою, з постумовою та циклів з лічильником (`for`). Також синтаксис дозволяє використовувати оператори логічного перемикавання (`switch`). Користувач може декларувати та викликати додаткові функції. Декларація функції здійснюється за допомогою ключового слова `function`, за яким наводиться необов’язковий ідентифікатор – ім’я функції (функції можуть бути анонімними), потім в круглих дужках вказується необов’язковий список параметрів, останнім вказується припис, відповідний тілу функції.

При оголошенні змінна розміститься в область видимості, яка відповідатиме операції в якій ця змінна оголошена. Якщо змінна оголошена поза функцією, то її область видимості вважається глобальною. Створення змінної в пам’яті відбувається при отриманні керування відповідною функцією (рис. 7.12).

```
var a; // a - undefined
var pi = 3.14; // pi ініціалізується значенням 3.14
function foo() {
    var a = 10; // a=10 - локальне в області видимості foo
}
function sqr(a) {
    return a * a;
}
var pi_2 = sqr(pi); // pi в квадраті
var arr = new Array(0, pi_2, 2); // створено масив з трьох елементів
print(arr[0]); // зневажувальний друк значення першого елемента (0)
```

Рисунок 7.12 – Змінні, функції та області видимості

Для створення масивів передбачений вбудований клас об’єктів `Array` (рис. 7.12, змінна `arr`), який дозволяє створювати масиви як заданої довжини, так і шляхом перерахування його елементів. Індексація елементів починається з нуля.

Отже, специфікації стандарту й описаний вище базовий синтаксис дозволяють визначити мову для маніпулювання об'єктами на базі примітивних типів. У підсистемі інженерного аналізу САПР машинобудування ці типи необхідно розширити високорівневими проблемно-орієнтованими абстракціями, які дозволять визначати такі об'єкти як точка, дискретна модель, параметри моделі, обчислювальний завдання тощо.

7.3.2 Допоміжні функції й операції

Допоміжні (арифметичні та тригонометричні) функції реалізовані в глобальному об'єкті Math, всі поля та методи якого статичні (для їх використання не потрібно створення екземпляра об'єкта). Методи даного об'єкта дозволяють обчислювати квадратний корінь (sqrt), натуральний логарифм (log), арккосинус (acos), арксинус (asin), арктангенс (atan, одного аргументу або відношення першого до другого аргументу, якщо їх два), синус (sin), косинус (cos), тангенс (tan), степінь (pow), модуль числа (abs), експоненту числа (exp), мінімум (min), максимум (max), найбільше ціле, що менше або дорівнює аргументу (floor), найменше ціле, що більше або дорівнює аргументу (ceil), округляти до найближчого цілого (round), генерувати випадкове число на відріжку [0; 1] (random).

В об'єкті Math оголошені деякі відомі математичні константи: число Ейлера ($E \approx 2,71828$), натуральний логарифм 2 ($LN2 \approx 0,693$), логарифм за основою 2 числа Ейлера ($LOG2E \approx 1,442$), логарифм за основою 10 числа Ейлера ($LOG10E \approx 0,434$), число π ($PI \approx 3,14159$), значення $\frac{1}{\sqrt{2}}$ ($SQRT12 \approx 0,707$), $\sqrt{2}$ ($SQRT2 \approx 0,414$) і натуральний логарифм 10 ($LN10 \approx 2,302$).

При моделюванні часто виникає необхідність використання таких операцій, як сума або добуток значень деякої множини аргументів, приблизна перевірка на рівність тощо. Операції знаходження суми та добутку є багатомісними (рис. 7.13)

```

sum = 'sum', '(', вираз, [ ',', вираз ], ')';
prod = 'sum', '(', вираз, [ ',', вираз ], ')';
approx = 'approx', '(', вираз, ',', вираз, [ ',', вираз ], ')';

```

Рисунок 7.13 – РБНФ суми, добутку та приблизної рівності

Операція перевірки на приблизну рівність є тримісною (рис. 7.13): перші два аргументи – вирази для перевірки на приблизну рівність, третій необов’язковий аргумент – вираз, що задає точність перевірки. Математично ця операція подається такою формулою:

$$\text{approx}(x, y, \varepsilon) = \begin{cases} 1, & |x - y| < \varepsilon, \\ 0, & |x - y| \geq \varepsilon, \end{cases}$$

де ε – точність перевірки.

Отже, вираз $\text{approx}(2.718281, 2.7183, 0.001)$ буде відповідати значенню 1 (“істина”), а вираз $\text{approx}(2.718281, 2.7183, 0.00001)$ – значенню 0 (“хибність”).

7.3.3 Основні геометричні примітиви

Для логічного конструювання функціональних моделей форм об’єктів використовуються операції кон’юнкції, диз’юнкції та різниці. Відповідно, проблемно-орієнтована мова має дозволяти користувачеві визначати нові функціонально задані форми з використанням цих операцій.

Багатомісні операції кон’юнкції, диз’юнкції та різниці можна подати такими формулами:

$$\begin{cases} \text{con}(x_1, x_2, \dots, x_n) = x_1 \wedge x_2 \wedge \dots \wedge x_n, \\ \text{dis}(x_1, x_2, \dots, x_n) = x_1 \vee x_2 \vee \dots \vee x_n, \\ \text{diff}(x_1, x_2, \dots, x_n) = x_1 \wedge \neg x_2 \wedge \dots \wedge \neg x_n, \end{cases}$$

яким відповідають РБНФ, наведено на рис. 7.14.

```
con = 'con', '(', вираз, [ ',', вираз ], ')';
dis = 'dis', '(', вираз, [ ',', вираз ], ')';
diff = 'diff', '(', вираз, ',', вираз, [ ',', вираз ], ')';
```

Рисунок 7.14 – РБНФ логічних операцій з функціональними формами

Кожній формулі, що відповідає геометричному примітиву з розділу 2, можна протиставити однойменну функцію проблемно-орієнтованої мови з еквівалентною сигнатурою.

Наприклад, модель l-області, яка задана формулою (2.13), можна записати за допомогою проблемно-орієнтованої мови у такому вигляді:

```
var w = 30;
var h = 40;
var a = 10;
var count = 11;
function main(x, y) {
    return dis(rectangle(x - w / 2, y - a / 2, w, a), rectangle(x - a / 2, y - h / 2, a, h));
}
```

Тривимірну модель пробки (формула (2.29)) можна записати у такому вигляді:

```
function main(x, y, z){
    return con(cylinder(x, y, z, 25, 35), -con(-circle(x, y, 18), band(z - 17.5 + 25 - 2.5, 5)), -
    con(-circle(x, y, 21), band(z + 17.5 - 15 - 10, 20)), circle(x, y, 21 + 14.5 - z), -
    con(regular(x, y, 22 / Math.sqrt(3), 6), band(z + 17.5 - 12.5, 25)));
}
```

Опис гібридних моделей здійснюється у два кроки. На першому кроці необхідно визначити відповідну до межі параметричну функцію. Водночас для визначення кривих і поверхонь Безьє передбачені відповідні примітиви.

Клас *Bezier* дозволяє визначити криву або поверхню Безьє на заданому наборі контрольних точок. Його конструктор має єдиний обов'язковий параметр – набір контрольних точок. Якщо параметром конструктора є масив двовимірних точок, то буде побудована крива. Для обчислення координати кривої, відповідної значенням параметра t передбачений метод класу *curve(t)*.

Якщо параметром конструктора класу *Bezier* буде матриця 4×4 , то результат – поверхня Безьє. Для обчислення точки на поверхні, що відповідає параметрам u і v передбачено метод *surface(u, v)*.

Для побудови дискретних моделей меж об'єктів передбачена функція-конструктор *BoundaryMesh*. Якщо у цій функції буде 4 параметри: кількість точок розбиття, початок і кінець інтервалу змін параметра, функція одного параметра, то буде побудована адаптивна дискретна модель для кривої. Якщо ж будуть передані 5 параметрів: число точок в напрямку u , число точок в напрямку v , початок і кінець інтервалу в першому напрямку, початок і кінець інтервалу в другому напрямку, функція двох параметрів, то буде побудована дискретна модель для параметричної поверхні.

Для перевірки належності точки простору до множини внутрішніх точок моделі форми об'єкту за допомогою тесту парності зі знаком в об'єктах класу *BoundaryMesh* передбачений метод *cfunction*, параметрами якого є дво- або тривимірні координати.

Наприклад, такий код буде відповідати моделі профілю НАСА (рис. 2.23):

```
var bt = new Bezier(new Array(new Point(0, 0), new Point(1.14034, 2.44284), new
Point(11.1025, 3.65375), new Point(30.8809, 1.13309), new Point(39.9276, 0.09712),
new Point(40, 0.0)));
var bb = new Bezier(new Array(new Point(0, 0), new Point(1.14034, -2.44284), new
Point(11.1025, -3.65375), new Point(30.8809, -1.13309), new Point(39.9276, -0.09712),
new Point(40, 0.0)));
function curve(t) {
    if (t <= 1.) return bt.curve(t);
```

```

    return bb.curve(2.0 - t);
};
var s = new BoundaryMesh(51, 0, 2.0, curve);
function main(x, y){
    return con(s.cfunction(x, y), -rectangle(x - 10, y, 6, 3), -circle(x - 20, y, 1.5), -
rectangle(x - 25, y, 2, 2));
}

```

7.3.4 Генерація дискретних моделей

Для забезпечення можливості генерації дискретних моделей в проблемно-орієнтованій мові передбачені відповідні вбудовані функції-конструктори. Найпростіші з них дозволяють генерувати рівномірні дискретні моделі.

Функція-конструктор *QuadrilateralMesh2D*($m, n, Point, w, h$) повертає екземпляр дискретної моделі на базі чотирикутників, яка складається з $m \times n$ вузлів, для прямокутника, нижній лівий кут якого заданий координатами *Point*, ширина – w , висота – h . Функція-конструктор *TriangularMesh2D* з такими ж параметрами поверне еквівалентний екземпляр на основі трикутників.

У тривимірному випадку конструктор *HexahedralMesh*($m, n, k, Point, w, h, d$) повертає екземпляр дискретної моделі на базі шестигранників, яка складається з $m \times n \times k$ вузлів, для прямокутної призми дальній нижній лівий кут, якої заданий координатами *Point*, ширина – w , висота – h , глибина – d . Конструктор *TetrahedralMesh* з такими ж параметрами поверне еквівалентний екземпляр на основі тетраедрів.

Зазначимо, що конструктори *QuadrilateralMesh2D*, *TriangularMesh2D*, *HexahedralMesh* і *TetrahedralMesh*, дозволяють конвертувати дискретні моделі форм об'єктів. Для цього єдиним параметром цих функцій повинен бути екземпляр дискретної моделі для конвертації.

Для генерації дискретних моделей меж об'єктів передбачено функцію-конструктор *BoundaryMesh*. Якщо функції *BoundaryMesh* передати два параметри: екземпляр дискретної моделі та неявну функцію області, то за допомогою алгоритму *background_grid_surface* буде згенерована дискретна модель межі, яка задана неявною функцією. Якщо у функції *BoundaryMesh* тільки один параметр – дискретна модель, то її результатом буде множина межових граней. Наприклад, для кубика, який заданий формулою (4.1), дискретну модель його межі (тривимірні трикутники) на базі фонові сітки тетраедрів (з конвертацією з шестигранників, рис. 4.5) можна описати таким кодом:

```
SLEVEL = 2;
OLEVEL = 8;
function quad(x, y, z){
    return con(cuboid(x, y, z, 2, 2, 2), -circle(x, y, 0.5), -circle(y, z, 0.5));
}
var n = 21;
var background = new TetrahedralMesh(new HexahedralMesh(n, n, n, new Point(-1, -1, -1), 2, 2, 2));
var mesh = new BoundaryMesh(background, quad);
```

У наведеному вище прикладі глобальні параметри SLEVEL і OLEVEL використовуються, відповідно, для обмеження кількості ітерацій згладжування Лапласа і покращення дискретних моделей на основі мінімізації функціоналу відстані-довжини (підрозділ 4.2).

Для генерації дискретної моделі з використанням узагальненого методу для елементів типу солід (алгоритм *background-grid-mesh*, підрозділ 5.2) необхідно передбачено функцію *BackgroundGridMesh* з двома параметрами: екземпляром фонові дискретної моделі та неявною функцією, що подає форму об'єкта. Наприклад, для кубика, форма якого подана формулою (4.1), дискретну модель на базі шестигранників можна описати таким кодом:

```

SLEVEL = 4;
OLEVEL = 8;
function quad(x, y, z){
    return con(cuboid(x, y, z, 2, 2, 2), -circle(x, y, 0.5), -circle(y, z, 0.5));
}
var n = 21;
var b = new HexahedralMesh(n, n, n, new Point(-1, -1, -1), 2, 2, 2);
var mesh = new BackgroundGridMesh(b, quad);

```

При генерації дискретних форм об'єктів на базі солідів глобальні параметри SLEVEL і OLEVEL використовуються для, відповідно, обмеження кількості ітерацій згладжування Лапласа внутрішніх вузлів та оптимізації дискретних моделей на основі локальної мінімізації функціоналу площі (або обсягу) зі знаком.

Для спрощення генерації дискретних моделей найбільш поширених типів оболонок у проблемно-орієнтованій мові передбачені такі функції: *CylinderTriangles*(n, m, r, h) і *CylinderQuads*(n, m, r, h), *ConeTriangles*($n, m, r1, r2, h$) і *ConeQuads*($n, m, r1, r2, h$). Перша пара призначена для створення рівномірних дискретних моделей (на базі трикутників або чотирикутників) для циліндричних поверхонь ($n \times m$ вузлів, радіуса – r і висота – h). Друга пара – для рівномірних дискретних моделей конічних поверхонь з $n \times m$ вузлів, з радіусами основ $r1$ і $r2$, висотою h . Якщо до цих параметрів додати функцію, яка описує форму отворів, то результатом буде дискретна модель, побудована у полі значень параметрів відповідної поверхні, з адаптацією до меж, заданих цією функцією. Наприклад, для конічної оболонки зі складною конфігурацією отворів (рис. 5.22) буде така формальна модель:

```

var r1 = 3.00 / 2.0;
var r2 = 3.98 / 2.0;
var h = 4.014;
var m = 80;

```

```

var n = ceil(m * 2 * PI * r1 / h);
var h1 = h - 0.580;
var h2 = h - 0.485;
var h3 = h - 0.460;
var h4 = 0.52;
var h5 = 0.72;
function holes(x, y, z){
    var xx1 = x * cos(-0.65) + z * sin(-0.65);
    var xx2 = x * cos(-1.31) + z * sin(-1.31);
    var xx3 = x * cos(-2.09) + z * sin(-2.09);
    var xx4 = x * cos(-1.31) + z * sin(-1.31);
    var zz4 = -x * sin(-1.31) + z * cos(-1.31);
    var xx5 = x * cos(2.09) + z * sin(2.09);
    var zz5 = -x * sin(2.09) + z * cos(2.09);
    var xx6 = x * cos(0.65) + z * sin(0.65);
    var zz6 = -x * sin(0.65) + z * cos(0.65);
    var xx7 = x * cos(-2.43) + z * sin(-2.43);
    var zz7 = -x * sin(-2.43) + z * cos(-2.43);
    return -dis(rectangle(xx1, y - h1, 0.40, 0.40, 0.055), rectangle(xx2, y - h1, 0.40, 0.40,
0.055), rectangle(xx3, y - h2, 0.40, 0.25, 0.065), con(rectangle(xx4, y - h4, 0.68, 0.45,
0.075), zz4), con(rectangle(xx5, y - h4, 0.68, 0.45, 0.075), zz5), con(rectangle(xx6, y - h5,
0.44, 0.88, 0.075), zz6), con(circle(xx7, y - h3, 0.105), zz7));
}
var mesh = ConeTriangles(n, m, r1, r2, h, holes);

```

Для побудови дискретних моделей передбачена можливість об'єднання частин. У кожного екземпляру дискретної моделі є метод *add*, єдиним параметром якого є екземпляр іншої дискретної моделі. Водночас метод *translate* дозволяє перемістити всі вузли моделі на заданий вектор. У результаті об'єднання моделей

вузли з однаковими координатами матимуть однакові коди. Наприклад, оболонку як комбінацію конуса та циліндра (рис. 7.15) можна подати у вигляді такого кода:

```
var pi = Math.PI;
var count = 50;
var cl = 5.912 - 0.070; // довжина циліндричної частини
var l = 3.1+6.012-5.912; // довжини конічної частини
var r1 = 0.536/2; // радіус меншої основи конічної частини
var r5 = 2.4/2; // радіус більшої основи конічної частини, радіус циліндричної частини
var mesh = new ConeQuads(count, count, r1, r5, l);
var c1 = new CylinderQuads(count, Math.ceil(count * cl / (2 * pi * r5)), r5, cl);
c1.translate(0, l, 0); // переміщення циліндра на радіус-вектор з координатами (0; l; 0)
mesh.add(c1);
```

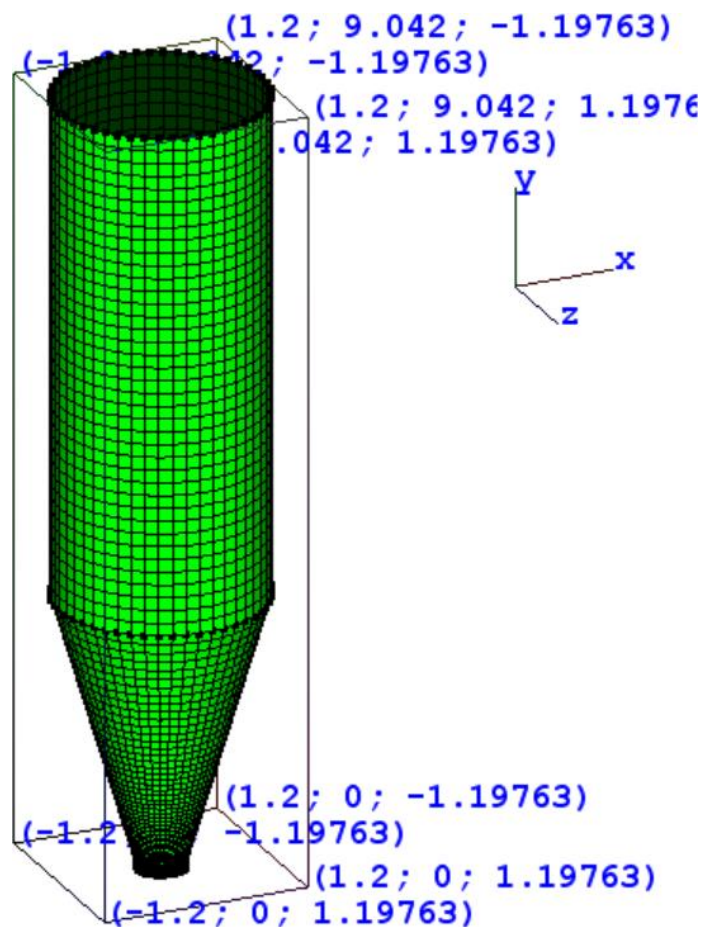


Рисунок 7.15 – Дискретна модель оболонки “Конус-Циліндр”

7.3.5 Опис скінченно-елементних задач

Після генерації дискретної моделі форми об'єкта для скінченно-елементного аналізу необхідно визначити крайові умови та вплив зовнішнього середовища. Як показано у підрозділі 7.1 крайові умови та впливи зовнішнього середовища можуть бути екземплярами одного базового класу *FeaValue* (рис. 7.4).

Для задання крайових умов передбачена функція-конструктор *BoundaryCondition(direction, select, value)* з трьома параметрами. Перший параметр – напрям дії *direction*, який може бути одним з таких значень: ALL (всі), FIRST (перше), SECOND (друге), THIRD (третє), FOURTH (четверте), FIFTH (п'яте), SIXTH (шосте), SEVENTH (сьоме), EIGHTH (восьме), NINTH (дев'яте), TENTH (десяте). Другий параметр – функція координат *select*, що повертає істину для точок, в яких діє певна умова (цей параметр необов'язковий, якщо він не заданий, то вважається, що умова діє на всі точки). Останній параметр – функція координат або значення *value* (якщо воно константне), які відповідні крайовій умові.

Для задання навантажень передбачені конструктори *NodalForce* (вузлові навантаження), *SurfaceForce* (поверхневі навантаження) і *VolumeForce* (об'ємні сили). Їх параметри синтаксично та семантично еквівалентні параметрам функції *BoundaryCondition* (результати є об'єктами спільного базового класу).

Для двовимірного скінченно-елементного аналізу передбачена функція *Fem2D(mesh, h, D, ...)*, обов'язковими трьома параметрами якої (за сигнатурою) є екземпляр дискретної моделі на базі двовимірних елементів *mesh*, товщина об'єкта *h*, матриця зв'язків між напруженнями та деформаціями *D*. Далі можна перерахувати довільна кількість крайових умов і навантажень.

Наприклад, задачу аналізу напружено-деформованого стану фланця з підрозділу 6.3 (рис. 6.16) можна описати у такому вигляді:

```
function main(x, y) {
    return con(dis(rectangle(x, y, 20, 0, 80, 100), rectangle(x, y, 0, 20, 100, 60),
```

```

circle(x - 20, y - 20, 20), circle(x - 20, y - 80, 20), circle(x - 100, y - 50, 50)),
-circle(x - 100, y - 50, 30), -circle(x - 20, y - 20, 10), -circle(x - 20, y - 80, 10));
}
SLEVEL = 4;
OLEVEL = 8;
var c = new Array(new Point(100, 20));
var b = new QuadrilateralMesh2D(1.5 * 31, 31, new Point(0, 0), 150, 100);
var mesh = new BackgroundGridMesh(b, main);
var E = 200000;
var nu = 0.33;
var q = 20 * 20;
var fixed = new BoundaryCondition( ALL,
    function (x, y){
        if (approx(circle(x - 20, y - 20, 10), 0.0) || approx(circle(x - 20, y - 80, 10), 0.0))
            return true;
        return false;
    },
    0);
var force = new SurfaceForce( FIRST,
    function (x, y){
        if (approx(circle(x - 100, y - 50, 30), 0.0)) return true;
        return false;
    },
    q);
Fem2D(mesh, 20, new PlaneStressMatrix(E, nu), fixed, force);

```

У наведеному вище прикладі функція-конструктор *PlaneStressMatrix*(*E*, *nu*) створює матрицю зв'язків між напруженнями та деформаціями (плоско-напружений стан) для заданих модуля Юнга *E* та коефіцієнта Пуассона *nu*. Аналогічно, *PlaneStrainMatrix* створює матрицю зв'язків для плоско-деформованого стану.

Для аналізу напружено-деформованого стану у тривимірному випадку передбачена функція *Fem3D(mesh, D, ...)* з двома обов'язковими параметрами. Перший параметр – екземпляр тривимірної дискретної моделі форми об'єкта на базі тривимірних солідів. Другий параметр – матриця зв'язків між напруженнями та деформаціями у тривимірному випадку. Починаючи з третього параметра можна перераховувати крайові умови і навантаження.

Для створення матриці зв'язків між напруженнями та деформаціями у тривимірному випадку передбачено функцію-конструктор *VolumeStressMatrix(E, nu)*, де *E* – модуль Юнга, *nu* – коефіцієнт Пуассона. Наприклад, аналіз напружено-деформованого стану стрижня під дією власної ваги з параметрами як у пункті 6.1.2 можна описати з використанням такого коду:

```
var l = 20;
var c = 1;
var E = 210000;
var nu = 0.3;
var gamma = 0.076500; // питома вага
var nx = 3;
var nz = (nx - 1) * 20 + 1;
var mesh = new HexahedralMesh(nx, nx, nz, new Point(-c/2, -c/2, 0), c, c, l);
var fixed = new BoundaryCondition( THIRD,
    function (x, y, z){
        if (approx(z, l)) return true;
        return false;
    }, 0);
var force = new VolumeForce( THIRD, -gamma);
Fem3D(mesh, new VolumeStressMatrix(E, nu), fixed, force);
```

Додатково передбачено функції *MindlinPlate* і *MindlinShell*, які можна використати для дослідження, відповідно, пластин (на базі двовимірних елементів) і

оболонок (на базі тривимірних трикутників або чотирикутників) на базі теорії деформацій зсуву другого порядку (узагальнення теорії Тимошенко-Міндліна), параметри цих функцій ідентичні тим, що використано у *Fem2D*. Але товщина у цих двох функціях може бути не константною, а функцією координат. Також, якщо товщина та матриця зв'язків між напруженнями та деформаціями є масивами знаень, то використовується відповідна теорія для багатопластин і оболонок.

Для всіх функцій скінченно-елементного аналізу можна активувати адаптивність. Для управління адаптивним скінченно-елементним аналізом передбачено три глобальні параметри: ADAPT, ALEVEL і ADELTA. Якщо параметр ADAPT дорівнює true, то використовується адаптивний аналіз. Водночас параметр ALEVEL є цілим числом – обмеженням на максимальну кількість ітерацій, а параметр ADELTA – на точність. За замовченням ADAPT=false, ALEVEL=1, ADELTA=1. Наприклад, для адаптивного аналізу напружено-деформованого стану оболонки-комбінації циліндра та конуса (вихідна дискретна модель зображена на рис. 7.15) можна використати такий код:

```
var cylinder_fix = new BoundaryCondition(ALL,
    function (x, y, z) {
        if (approx(y, cl + l)) return true;
        return false;
    }, 0);
var axial = new SurfaceForce(SECOND,
    function (x, y, z) {
        if (approx(y, 0)) return 0.48 * 9806.65 * 1.0E-6;
        return 0;
    });
var q = 0.5 * 98066.5 * 10.0E-6;
var sl = Math.sqrt(l*l + r5*r5);
var force_x = new ShellDistributedForce(FIRST,
    function (x, y, z) {
```

```

    if (y <= l) return -q * (l / sl) * Math.cos(Math.atan2(z, x));
    return -q * Math.cos(Math.atan2(z, x));
});

var force_y = new ShellDistributedForce(SECOND,
    function (x, y, z) {
        if (y <= l) return q * (r5 - r1) / sl;
        return 0;
    });

var force_z = new ShellDistributedForce(THIRD,
    function (x, y, z) {
        if (y <= l) return -q * (l / sl) * Math.sin(Math.atan2(z, x));
        return -q * Math.sin(Math.atan2(z, x));
    });

function thickness(x, y, z){
    if ((l - 0.025 <= y && y <= l + 0.025) || l + cl - 0.025 <= y) return 0.0418;
    if ((l - 0.1 <= y && y <= l + 0.1) || l + cl - 0.1 <= y) return 0.0358;
    return 0.0282;
}

ADAPT = true; ALEVEL = 3; ADELTA = 0.2;
MindlinShell(mesh, thickness, new PlaneStressMatrix(34000, 0.3), cylinder_fix, axial,
force_x, force_y, force_z);

```

У наведеному вищій прикладі крайова умова *cylinder_fix* буде застосована до всіх вузлів, у яких координата *y* дорівнює *cl+l* (відповідним вузлам заборонено переміщення у всіх напрямках). Сила *axial* діє тільки на сторони елементів, для яких *y=0* (менший радіус конічної частини), та відповідає осьовому стисненню. Сили *force_x*, *force_y* і *force_z* моделюють дію рівномірного зовнішнього тиску (вони застосовують до всіх елементів). Товщина є функцією координат оболонки, вона збільшується на з'єднанні конічної та циліндричної частин і біля закріпленої основи циліндричної частини (результати інтерпретації цього коду наведені на рис. 7.16).

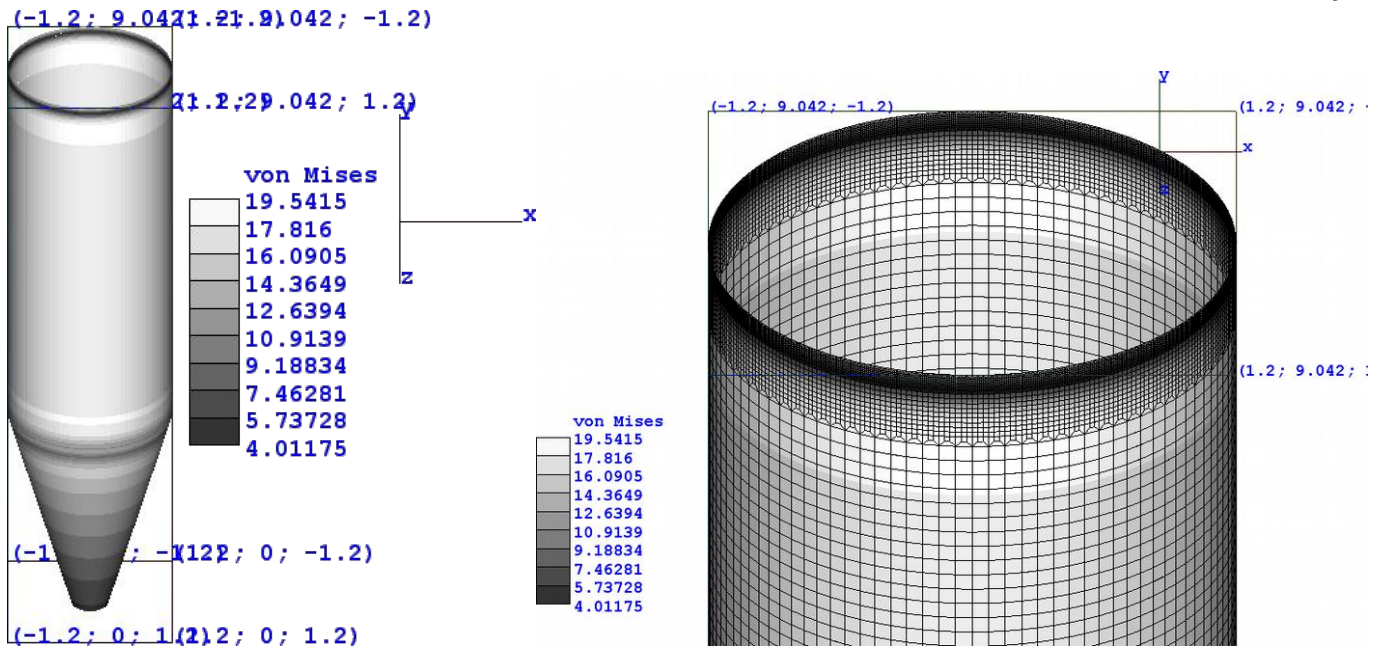


Рисунок 7.16 – Напружено-деформований стан оболонки «Конус-Циліндр»

Висновки до розділу 7

Отже, у цьому розділі розроблено чотири базові шаблони проектування. Перший – шаблон «UI–Model–Analysis» виконує декомпозицію САПР машинобудування на високорівневі програмні підсистеми. Другий – шаблон «Representation–Mesh» визначає відношення між класами, які відповідають за подання моделі форми об'єкта, і класами, які відповідають за генерацію дискретних моделей. Шаблон «Element–Node» визначає об'єктну декомпозицію в дискретних моделях форм об'єктів. Четвертий, шаблон «FEA Problem» визначає загальну структуру задачі скінченно-елементного аналізу. Ці шаблони як програмні абстракції можна поєднати з іншими відомими для забезпечення більшої гнучкості та спрощення процесу супроводження програмного кода. Їх основною перевагою є абстракції для розробки САПР машинобудування без обмежень на вихідні коди.

Запропонований метод генерації дискретних моделей з використанням технологій паралельних обчислень дозволяє зменшити час, витрачений на моделювання, якщо необхідна велика кількість вузлів для досягнення точності.

Зокрема, отримані залежності часу від кількості вузлів доводять, що за малої кількості вузлів економія часу від використання паралельних технологій може бути відсутньою, що підтверджує закон Амадала. Водночас, чим складніша модель форми об'єкта, тим менша кількість вузлів потрібна для економії витрат часу за рахунок паралельності метода.

Розроблена у цьому розділі проблемно-орієнтована мова дозволяє, використовуючи схожий з конструюванням підхід, описувати форми об'єктів. Для цього у синтаксисі мови передбачено примітиви, відповідні неявним функціям, які подають поширені форми. Водночас вбудовані та створені користувачем функції можна використовувати при описі скінченно-елементних задач (наприклад, для визначення областей дії навантажень). Як результат, забезпечено гнучкість моделювання об'єктів у САПР машинобудування.

Результати сьомого розділу опубліковано у роботах [166, 167, 169, 157, 173, 174, 180, 182, 252, 240].

Список джерел, які використано у даному розділі, наведено у переліку джерел посилення [190, 214, 241, 248, 251, 257, 301, 431].

ВИСНОВКИ

У дисертаційній роботі отримані результати, які у сукупності є вирішенням науково-технічної проблеми підвищення ефективності проектування об'єктів машинобудування за рахунок збільшення точності математичного моделювання та зменшення часу, необхідного для дослідження їх стану.

1. У роботі виконано аналіз сучасного стану проблеми підвищення ефективності проектування об'єктів машинобудування, на основі якого встановлено наступне: математичне моделювання дозволяє замінити вартісний натурний експеримент дослідженням математичної моделі; поширені в аналізі поведінки об'єктів системи диференціальних рівняння, зазвичай, не мають аналітичного розв'язку; на практиці застосовують чисельні методи, засновані на ідеї переходу від неперервної моделі форми об'єкта до дискретної; функціональне подання, що засноване на використанні неявних функцій, є найбільш універсальним і теоретично дозволяє описати форму об'єкта довільної складності; для скінченно-елементного аналізу із заданою точністю необхідна розробка методів генерації дискретних моделей функціонально поданих форм; виявлено протиріччя між необхідністю побудови дискретних моделей форм об'єктів, які дозволять із заданою точністю виконувати чисельний аналіз стану об'єктів у САПР машинобудування, та обмеженістю методів, що існують.

2. Типова задача побудови математичної моделі, що описує форму об'єкта машинобудування, як вихідні дані має креслення. Використовуючи функціональний підхід, можна побудувати модель об'єкта як єдину неявну функцію. Проте, у машинобудуванні зустрічаються об'єкти, форма яких описуються (повністю або частково) параметричними функціями. Запропонований математичний апарат дозволяє одночасно використовувати примітиви, межі яких визначені за допомогою параметричних і неявних функцій. Водночас моделі на базі параметричних функцій

можуть виступати як операнди R-операцій. Отримані моделі можна розглядати як гібридні, що поєднують функціональне подання на базі неявних функцій і граничне подання на базі параметричних функцій. Вперше запропоновано гібридний функціональний підхід до подання форм об'єктів з використанням примітивів, які задані неявними функціями (функціональне подання) та власними границями (граничне подання), що дозволяє зменшити час побудови математичних моделей.

3. Подання дискретної моделі множинами вузлів і елементів водночас зі зберіганням для кожного вузла разом з його координатами списку інцидентних елементів дозволяє знизити порядок складності алгоритмів вставки та видалення вузлів. Спискові структури даних є більш ефективними при видаленні вузлів, а векторні – при безпечному додаванні (при вставці з перевіркою існування вузлів з однаковими координатами).

4. Вперше запропоновано проекційний метод генерації дискретних моделей меж об'єктів, форма яких задана функціонально, що дозволяє будувати дво- та тривимірні дискретні моделі. Водночас щільність вузлів у підсумковій моделі залежить від щільності вузлів у вихідній. Проте, запропонований метод не враховує геометричних особливостей форми об'єкта, отже, повинна застосовуватися додаткова оптимізація.

5. Вперше запропоновано метод згладжування координат вузлів дискретних моделей меж об'єктів, форма яких задана функціонально, у результаті чого можна підвищити точність моделювання в околах зламів їх поверхонь. Метод ґрунтується на мінімізації функціоналу, який враховує відстань від елементів дискретної моделі до поверхні та довжини їх сторін. Цей метод має локальний характер, тому потрібно його застосовувати ітераційно. За малої кількості ітерацій (менше чотирьох) можна отримати дискретні моделі форм об'єктів низької якості. Починаючи з четвертої ітерації, геометричні особливості форм об'єктів вже апроксимуються вузлами. Водночас виконання більше восьми ітерацій несуттєво змінює якість моделі. У результаті застосування цього методу мінімальні кути для трикутників несуттєво

збільшуються, а у чотирикутниках можна отримати елементи, за формою близькі до трикутників, отже, результати загладжування потребують додаткової обробки топологічними перебудовами елементів.

6. Отримав подальшого розвитку метод локальних перебудов елементів дискретних моделей у частині його узагальнення на випадок дискретних моделей на базі трикутних або чотирикутних елементів для меж об'єктів, що дозволяє підвищити точність скінченно-елементного аналізу. Зокрема, метод локальних перебудов елементів дискретних моделей меж об'єктів дозволяє збільшити значення мінімального кута в елементі для дискретних моделей на базі трикутників і зменшити максимальний кут в елементі для чотирикутників.

7. Вперше запропоновано метод генерації дискретних моделей на базі елементів типу солід для форм об'єктів, заданих функціонально, який дозволяє будувати дво- і тривимірні моделі як з використанням симплексів (трикутників або тетраедрів), так і з використанням топологічних кубів (чотирикутників або шестигранників). Цей метод дозволяє використовувати спільний алгоритм для генерації дискретних моделей на базі різних форм скінченних елементів. Підсумкові дискретні моделі, зазвичай, потребують додаткової обробки.

8. Вперше запропоновано метод згладжування координат внутрішніх вузлів дискретних моделей форм об'єктів, відмінність якого від інших методів полягає в усуненні вироджених елементів та підвищенні точності скінченно-елементного аналізу. Використання у цьому методі локального алгоритму мінімізації (ітераційно для кожного вузла) дозволяє отримати якість дискретних моделей, порівнянну з результатами глобального алгоритму (для всіх вузлів одночасно), з використанням меншого обсягу обчислювальних ресурсів.

9. Вперше запропоновано метод генерації адаптивних дискретних моделей форм об'єктів (adaptive meshing), заданих функціонально, який дозволяє будувати дво- і тривимірні дискретні моделі для виконання скінченно-елементного аналізу із потрібною точністю. Запропонований метод враховує результати скінченно-

елементного аналізу, а не тільки інформацію про форму об'єкта. Як результат, кількість вузлів і елементів збільшується у тих областях, де найбільше змінюється потрібна досліднику величина. Метод не залежить від форми елементів. Він використовує таблиці шаблонів розбиття елементів. Отже, можна використовувати єдиний алгоритм для основних форм елементів.

10. Отримали подальшого розвитку методи генерації дискретних моделей форм об'єктів і скінченно-елементного аналізу у частині розробки паралельних алгоритмів, що дозволяє зменшити витрати часу на моделювання. Зокрема, досліджені залежності часу від кількості вузлів доводять, що за малої кількості вузлів економія часу від використання паралельних технологій може бути відсутньою, що підтверджує закон Амадала. Водночас чим складніша модель форми об'єкта, тим менша кількість вузлів потрібна для економії витрат часу за рахунок паралельності метода.

11. Вперше розроблена проблемно-орієнтована мова, яка дозволяє описувати об'єкти машинобудування складної форми з використанням гібридного підходу, а також задачі побудови дискретних моделей та скінченно-елементного аналізу.

12. Розроблено шаблони проектування САПР машинобудування як складної програмної системи. Ці шаблони як програмні абстракції можна поєднати з іншими відомими для забезпечення більшої гнучкості та спрощення процесу супроводження програмного коду. Їх основною перевагою, є можливість розробки САПР машинобудування без обмежень на вихідні коди.

Перевагою розроблених методів генерації та обробки дискретних моделей форм об'єктів є їх узагальненість. Отже, можлива їх реалізація на базі шаблону проектування «Стратегія», що дозволяє підвищити загальну технологічність САПР машинобудування як програмного продукту.

Для практичної реалізації отриманих результатів розроблено програмний продукт, який дозволяє генерувати дво- і тривимірні дискретні моделі, а також здійснювати скінченно-елементний аналіз поведінки об'єктів машинобудування.

Практичне використання результатів дисертаційної роботи дозволяє отримувати такі науково-технічні ефекти:

- підвищення якості візуального подання моделей об'єктів машинобудування;
- зниження витрат на натурні випробування об'єктів, що проектуються, за рахунок підвищення точності моделювання при дослідженні їх стану з використанням чисельних методів на базі дискретних моделей.

Напрями подальших досліджень за темою дисертаційної роботи можуть бути пов'язані з використанням методів машинного навчання при створенні моделей форм об'єктів машинобудування, генерацією адаптивних дискретних моделей, а також прогнозуванням можливих зон руйнування в конструкціях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Авершин А. Г. Аэродинамическое проектирование внешних форм экомобиля // Вестник ХНАДУ. 2012. № 56. С. 138–143.
2. Алгоритм декомпозиции геометрических объектов в 2D-задачах упаковки и раскроя / Ю. Г. Стоян, Н. И. Гиль, Т. Е. Романова, М. В. Злотник // Кибернетика и системный анализ. 2011. Т. 47, № 6. С. 28–37.
3. Алгоритм оцінки економічної ефективності впровадження інформаційної технології у торговельну організацію / Л. І. Нефьодов, Д. О. Маркозов, І. Г. Ільге, В. О. Беспалий. // Вісник Національного технічного університету "ХПІ". Сер. : Нові рішення в сучасних технологіях. 2013. № 4. С. 55–60.
4. Анисимов Д. О., Кудрявцев С. М. Оптимальное проектирование силового каркаса кузова полуприцепа // Известия вузов. Машиностроение. 2008. № 9. С. 51–54.
5. Бойко А. П., Бондаренко А. В. Автоматизированное проектирование судов с малой площадью ватерлинии // Судовождение. 2013. Вып. 22. С. 20–30.
6. Борисовская Ю. А., Гоменюк С. И. Современные САПР в аэрокосмической промышленности // Вісник Запорізького національного університету. Фізико-математичні науки. 2016. № 1. С. 20–31.
7. Бреславський Д. В., Коритко Ю. М., Татарінова О. А. Проектування та розробка скінченноелементного програмного забезпечення. Харків: Вид-во «Підручник НТУ "ХПІ"», 2017. 232 с.
8. Бубенчиков А. М., Фирсов Д. К., Котовщикова М. А. Ламинарное вязкое течение в искривленных каналах сложного сечения // Математическое моделирование. 2004. Т. 16. № 1. С. 107–119.
9. Бувайло Д. П., Гоменюк С. І., Толок В. О. FORTU – мова опису схем розв’язання задач математичної фізики // Вісник Запорізького державного університету. 2000. № 1. С. 19–25.
10. Водопьянов Р. Ю., Титок В. П., Артамонова А. Е. Программный комплекс ЛИРА-САПР 2015: учебное пособие / ред. А. С. Городецкого. Москва: Электронное издание, 2015. 459 с.

11. Гагауз П. М., Гагауз Ф. М. Применение одномерной дискретной модели для анализа напряженно-деформированного состояния адгезионных соединений // Вопросы проектирования и производства конструкций летательных аппаратов. 2015. Вып. 3. С. 45–55.
12. Галанин М. П., Щеглов И. А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы. Москва: ИПМ им М. В. Келдыша РАН, 2006. № 9. 32 с. (Препринт / РАН, ИПМ им. М. В. Келдыша ; 06-01-00421).
13. Глушко К. А., Мухін В. В., Чопоров С. В. Автоматизована підсистема побудови трьох характеристик поверхні функції двох змінних // Актуальні проблеми математики та інформатики: збірка тез доповідей III Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 26–27 квітня 2012 р.). Запоріжжя: ЗНУ, 2012. – С. 69.
14. Голованов Н. Н. Геометрическое моделирование. Москва: Academia, 2011. 272 с.
15. Гоменюк С. И. Принципы визуализации результатов расчета трехмерных задач механики деформируемого твердого тела в инструментальной системе FORTU // Вестник Херсонского государственного технического университета. 2003. № 3(19). С. 102–105.
16. Гоменюк С. И. Объектно-ориентированные модели и методы анализа механических процессов // Запорожский государственный университет. Запорожье, 2004. С. 293–310.
17. Гоменюк С. И. Вычислительный комплекс для автоматизации проектирования сложных инженерных конструкций и сооружений // Вестник Херсонского государственного технического университета. 2004. № 1. С. 339–342.
18. Гоменюк С. І. Об'єктно-орієнтовані моделі і методи аналізу механічних процесів у системі автоматизації проектування: автореф. дис. на здобуття наук. ступеня д-ра техн. наук: [спец.] 05.13.12 “системи автоматизації проектувальних робіт” / Харківський національний університет радіоелектроніки. Харків, 2004. 34 с.
19. Гоменюк С. И., Мухин В. В., Морозов Д. Н. Инструментальный подход к построению универсальной САПР в машиностроении и строительстве

FORTU-FEM // Вестник Херсонского национального технического университета. 2005. № 2. С. 107–113.

20. Гоменюк С. И., Гребенюк С. Н., Тархова В. М. Пространственно-временной конечный элемент для определения напряженно-деформированного состояния вязкоупругих материалов // Пробл. обчисл. механіки і міцності конструкцій. 2010. Вип. 14. С. 109–115.

21. Гоменюк С. И., Чопоров С. В., Аль-Омари М. А. В. Функциональное сглаживание сеток конечных элементов // Технология приборостроения. 2015. № 1. С. 10–14.

22. Гоменюк С. И., Чопоров С. В., Лисняк А. А. FORTU-FEM – автоматизированная система для исследования сложных механических процессов // Вестник Херсонского национального технического университета. 2012. № 2(45). С. 101–105.

23. Городецкий А. С. К расчету комбинированных систем методом конечных элементов // Сопротивление материалов и теория сооружений. 1972. № 16. С. 123–125.

24. Городецкий А. С., Шмуклер В. С., Бондарев А. В. Информационные технологии расчета и проектирования строительных конструкций: учебное пособие. Харьков: НТУ «ХПИ». 2003. 889 с.

25. Городецкий А. С., Евзеров И. Д. Компьютерные модели конструкций. Киев: Факт, 2007. 394 с.

26. Делоне Б. Н. О пустой сфере // Известия АН СССР. ОМОН. 1934. № 4. С. 793–800.

27. Дырда В. И., Гребенюк С. Н., Гоменюк С. И. Аналитические и численные методы расчета резиновых деталей : монография. Днепропетровск; Запорожье, 2012. 370 с.

28. Исследование задач механики деформируемого твердого тела в вычислительном комплексе «МІРЕЛА+» / Р. В. Киричевский, Е. С. Решевская, В. М. Тархова, Е. В. Прокопенко, В. В. Киричевский // Радіоелектроніка. Інформатика. Управління. 2006. № 2. С. 54–60.

29. Карпов Я. С., Гагауз П. М., Гагауз Ф. М. Общая методика определения напряженно-деформированного состояния адгезионных соединений в линейной одномерной постановке // Вопросы проектирования и производства конструкций летательных аппаратов. 2014. Вып. 2. С. 7–20.

30. Карпов Я. С., Литвинова Т. А., Павленко В. Н. Моделирование напряженно-деформированного состояния стрингерных панелей из композиционных материалов // Вопросы проектирования и производства конструкций летательных аппаратов. 2010. Вып. 1. С. 7–14.

31. Киричевский В. В. Система «КОДЕТОМ» для исследования высокоэластичных массивных конструкций на основе МКЭ // Комплексный расчет зданий и сооружений с применением ЭВМ. Киев: КИСИ, 1978. С. 142–148.

32. Киричевский В. В., Дохняк Б. М., Козуб Ю. Г. Метод конечных элементов в механике разрушения эластомеров. Киев: Наукова думка, 1998. 200 с.

33. Киричевский В. В., Толок В. А. Метод конечных элементов и суперэлементов в приложении к трехмерным задачам механики. Киев: Наукова думка, 2001. 364 с.

34. Киричевский В. В., Гребенюк С. М., Гоменюк С. І. Напружено-деформований стан циліндричного амортизатора в умовах контактної взаємодії // Систем. Технології. 2004. № 3. С. 92–97.

35. Киричевский В. В., Гребенюк С. Н., Мизерная Е. Л. Напряженно-деформированное состояние полого ортотропного цилиндра в условиях вязкоупругого деформирования // Нові матеріали і технології в металургії та машинобудуванні. 2005. № 2. С. 74–80.

36. Кириченко І. Г., Черніков О. В. Розробка комп'ютерних моделей будівельних і дорожніх машин та проведення віртуальних експериментів щодо режимів їх роботи // Проблеми розвитку дорожньо-транспортного і будівельного комплексів: матеріали міжнародної науково-практичної конференції, присвяченої 40-річчю кафедри будівельних, дорожніх машин і будівництва (м. Кіровоград, 3–5 жовтня 2013 р.). Кіровоград: КНТУ, 2013. С. 181–184.

37. Кліценко Г. О., Чопоров С. В. Розробка способу врахування переміщення вузлів скінченно-елементної сітки // Актуальні проблеми математики та

інформатики: збірка тез доповідей IV Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 25–26 квітня 2013 р.). Запоріжжя: ЗНУ, 2013. 135 с.

38. Кліценко Г. О., Чопоров С. В. Розробка способу врахування тріщин конструкцій в сітці на базі чотирикутників // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. С. 147.

39. Козлова О. С., Гоменюк С. И. Современные тенденции развития САПР в механике // Вісник Запорізького національного університету. Фізико-математичні науки. 2014. № 1. С. 38–51.

40. Комп'ютерна програма «Автоматизована система "Stability"»: Свідоцтво про реєстрацію авторського права на твір № 53201 Україна / С.І. Гоменюк, С. М. Гребенюк, В. З. Грищак, Ю. М. Тамуров, С.В. Чопоров; дата реєстрації 20.01.2014.

41. Комп'ютерна програма «Бібліотека консольної індикації «qzIO» («qzIO»): Свідоцтво про реєстрацію авторського права на твір № 70462 Україна / С. В. Чопоров, В. З. Грищак, С. М. Гребенюк, С. І. Полюга; дата реєстрації 15.02.2017.

42. Комп'ютерна програма «Бібліотека паралельної обробки розріджених матриць qzMatrix» («qzMatrix»): Свідоцтво про реєстрацію авторського права на твір № 70465 Україна / С. В. Чопоров, С. І. Гоменюк, А. О. Лісняк; дата реєстрації 15.02.2017.

43. Комп'ютерна програма «Система автоматизованого проектування FORTU-FEM»: Свідоцтво про реєстрацію авторського права на твір № 43043 Україна / С. І. Гоменюк, В. З. Грищак, С. М. Гребенюк, С. В. Чопоров, А. О. Лісняк; дата реєстрації 02.04.2012.

44. Комп'ютерна програма «Система автоматизованого проектування «qMesher»»: Свідоцтво про реєстрацію авторського права на твір № 53829 Україна / С.В. Чопоров, С.І. Гоменюк, С.М. Гребенюк, А.О. Лісняк; дата реєстрації 24.02.2014.

45. Комп'ютерна програма «Паралельний інтерпретатор мови специфікацій математичних моделей геометричних об'єктів»: Свідоцтво про реєстрацію

авторського права на твір № 82295 Україна / С. В. Чопоров, А. О. Лісняк, О. В. Кудін, М. С. Ігнатченко; дата реєстрації 17.10.2018.

46. Кравченко В. Ф., Рвачев В. Л. Алгебра логики, атомарные функции и вейвлеты в физических приложениях. Москва: ФИЗМАТЛИТ, 2006. 416 с.

47. Кудин А. В., Тамуров Ю. Н., Чопоров С. В. Аналитический и численный анализ изгиба круглой трехслойной пластины под действием локальных нагрузок // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 67–81.

48. Кудин А. В., Чопоров С. В. Компьютерное моделирование изгиба круглой трехслойной пластины с использованием аналитического и численного подходов // Радиоэлектроника, информатика, управление. 2014. № 1(30). С. 75–81.

49. Куценко Л. М. Застосування чотиримісних R-операцій для опису геометричних об'єктів // Прикладна геометрія та інженерна графіка: Міжвідомчий науково-технічний збірник. 2001. Вип. 69. С. 17–20.

50. Куценко Л. Н., Росоха С. В., Сулима В. В. Геометрическое моделирование и некоторые приложения обкатки треугольником Релло // Проблемы машиностроения. 2001. Т. 4, № 3–4. С. 85–94.

51. Куценко Л. М., Росоха С. В. Геометричне моделювання розсіювання потоку альфа-частинок полем іншої частки // Праці. Таврійська державна агротехнічна академія. 2001. Т. 13. Вип. 4. С. 17–21.

52. ЛИРА 9.4. Примеры расчета и проектирования : учебное пособие / В. Е. Боговис, Ю. В. Гензерский, Ю. Д. Гераймович [и др.]. Киев: Факт, 2008. 280 с.

53. ЛИРА 9.4. Руководство пользователя. / Е. Б. Стрелец-Стрелецкий, В. Е. Боговис, Ю. В. Гензерский [и др.] Основы. Киев: Факт, 2008. 164 с.

54. Лисин Д. А., Максименко-Шейко К. В., Шейко Т. И. Математическое моделирование поверхностей автомобилей с помощью R-функций // Проблемы машиностроения. 2013. Т. 16. № 3. С. 51–60.

55. Лісняк А. А. Трикутні скінченні елементи у математичному моделюванні геометричних об'єктів на базі теорії R-функцій: автореф. дис. на здобуття наук. ступеня канд. фіз.-мат. наук: [спец.] 01.05.02 “математичне моделювання та обчислювальні методи” / Запорізький національний університет. Запоріжжя, 2012. 20 с.

56. Лисняк А. А., Гоменюк С. И. Применение R-функций для геометрического моделирования объектов сложной формы // *Радіоелектроніка, інформатика, управління*. 2009. № 2. С. 76–81.

57. Лисняк А. А. Способ построения дискретных математических геометрических объектов, заданных с помощью R-функций // *Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки*. 2013. № 1. С. 59–69.

58. Лисняк А. А. Дискретизация границы трехмерных моделей геометрических объектов, заданных с помощью R-функций // *Радіоелектроніка, інформатика, управління*. 2014. № 1. С. 82–88. DOI: 10.15588/1607-3274-2014-1-12

59. Лисняк А. А., Чопоров С. В. Способы определения областей сгущения вычислительной сетки для моделей на базе теории R-функций // *Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 5–8 листопада 2012 р.)*. Харків: Ротапринт Інституту проблем машинобудування ім. А. М. Підгорного НАН України, 2012. – С. 38.

60. Литвин О. М., Першина Ю. І. Розв'язання тривимірної задачі комп'ютерної томографії з використанням невеликої кількості томограм // *Доповіді Національної академії наук України*. 2011. № 5. С. 40–44.

61. Литвин О. М., Першина Ю. І. Відновлення розривної внутрішньої структури двовимірного тіла за відомими її проекціями вздовж взаємно перпендикулярних ліній // *Вісник Національного технічного університету «ХПІ»*. Математичне моделювання в техніці та технологіях. 2013. № 54. С. 144–154.

62. Литвин О. М., Ткаченко О. В. Математичне моделювання процесів інтерполяційними сплайнами на нерегулярній сітці вузлів // *Доповіді Національної академії наук України*. 2010. № 1. С. 34–39.

63. Литвин О. М., Штепа Н. І. Математичне моделювання розподілу корисних копалин за допомогою інтерлінації функцій трьох змінних // *Доповіді Національної академії наук України*. 2009. № 2. С. 36–40.

64. Литвин О. Н., Ткаченко А. В., Литвин О. О. Общий метод построения уравнений кривых и поверхностей в неявной форме с помощью интерликации и

интерфлетиции функций // Кибернетика и системный анализ. 2011. Т. 47, № 1. С. 61–67.

65. Литвинова Ю. С., Максименко-Шейко К. В., Шейко Т. И. R-функции в математическом моделировании 3D-дома для реализации технологий 3D-печати // Вісник Харківського національного університету імені В. Н. Каразіна. Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. 2014. № 1105(24). С. 122–128.

66. Литвинова Ю. С., Максименко-Шейко К. В., Шейко Т. И. Математическое и компьютерное моделирование строительных конструкций на основе R-функций // Проблемы машиностроения. 2014. Т. 17. № 3. С. 45–52.

67. Литвинова Ю. С., Максименко-Шейко К. В., Шейко Т. И. Применение R-функций в моделировании строительных конструкций // Прикладні проблеми механіки і математики. 2014. Вип. 12. С. 165–173.

68. Лялюхина И. В. Проектирование балки из композитов при нескольких расчетных случаях нагружения // Вопросы проектирования и производства конструкций летательных аппаратов. 2011. Вып. 1. С. 38–48.

69. Максименко-Шейко К. В., Мацевитый А. М., Шейко Т. И. Автоматизация построения уравнений геометрических объектов в методе R-функций // Кибернетика и системный анализ. 2006. № 2. С. 148–157.

70. Максименко-Шейко К. В., Шейко Т. И. R-функции в математическом моделировании геометрических объектов, обладающих симметрией // Кибернетика и системный анализ. 2008. № 6. С. 75–83.

71. Максименко-Шейко К. В. Метод R-функций в математическом моделировании теплообмена при движении жидкости по цилиндрическим каналам с пристеночными винтовыми вставками // Вісник Харківського національного університету імені В. Н. Каразіна. Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. 2009. № 863(12). С. 180–190.

72. Максименко-Шейко К. В. Нормальные и нормализованные уравнения геометрических объектов в методе R-функций // Вісник Харківського національного університету імені В. Н. Каразіна. Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. 2009. № 847(11). С. 251–265.

73. Максименко-Шейко К. В. R-функции в математическом моделировании геометрических объектов и физических полей. Харьков: ИПМаш НАН Украины, 2009. 306 с.

74. Максименко-Шейко К. В., Шейко Т. И. R-функции в математическом моделировании геометрических объектов в 3D по информации в 2D // Вісник Запорізького національного університету: Збірник наукових праць. Фізико-математичні науки. 2010. № 1. С. 98–104.

75. Максименко-Шейко К. В., Шейко Т. И. Математическое моделирование тепловых режимов радиоэлектронных плат с размещением источников по схеме «ковёр Серпинского» // Математичні методи та фізико-механічні поля. 2012. Т. 55. № 3. С. 170–177.

76. Максименко-Шейко К. В., Шейко Т. И. Математическое моделирование геометрических фракталов с помощью R-функций // Кибернетика и системный анализ. 2012. Т. 48. № 4. С. 155–162.

77. Мартюшов С. Н. Расчет пространственных задач обтекания на основе TVD схемы Хартена // Вычислительные технологии. 1995. Т. 14. № 12. С. 219–228.

78. Мартюшов С. Н. Построение дву- и трехмерных сеток для задач газодинамики на основе уравнения Пуассона // Известия высших учебных заведений. Математика. 1997. № 4. С. 108–110.

79. Мартюшов С. Н., Мартюшова Я. Г. Численное моделирование струйных течений методом конечного объема на основе TVD-схемы 2-го порядка точности // Вычислительные технологии. 2004. Т. 9. № 4. С. 57–65.

80. Мартюшов С. Н., Мартюшова Я. Г. Численное моделирование течений детонации газовых смесей методом конечного объема // Вычислительные технологии. 2008. Т. 13. № 1. С. 88–98.

81. Математическое моделирование и исследование прочности силовых элементов конструкций космических летательных аппаратов / Д. В. Акимов, В. З. Грищак, С. И. Гоменюк, С. Н. Гребенюк, А. А. Лисняк, С. В. Чопоров, И. Ф. Ларионов, Д. В. Клименко, В. Н. Сиренко // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 6–13.

82. Математическое моделирование некоторых крепежных соединений на базе теории R-функций / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, Е. В. Панасенко. // Радиоэлектроника, информатика, управление. 2012. №2(27). С. 117–122.

83. Математическое моделирование температурных полей при затвердевании металлов в формах сложного профиля / В. Л. Рвачев, А. Н. Шевченко, Т. И. Шейко [та ін.] // Математичні методи та фізико-механічні поля. 1997. № 1. С. 59–65.

84. Математическое обеспечение анализа прочности силовых элементов ракетно-космической техники / Д. В. Акимов, В. З. Грищак, И. Ф. Ларионов, С. И. Гоменюк, Д. В. Клименко, С. В. Чопоров, С. Н. Гребенюк // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 26. С. 5–17.

85. Мацак В. А., Бледная Н. Ф. Расчетные исследования кузовов автобусов из различных материалов // Вестник ХНАДУ. 2005. № 29. 4 с.

86. Метод конечных элементов в проектировании транспортных сооружений / А. С. Городецкий, В. И. Заварицкий, А. А. Рассказов, А. И. Лантух-Лященко. Москва: Транспорт, 1981. 142 с.

87. Метод конечных элементов: теория, алгоритмы, реализация / В. А. Толок, В. В. Киричевский, С. И. Гоменюк [и др.] К.: Наукова думка, 2003. 316 с.

88. Методы построения дискретных моделей: неструктурированные сетки / С. В. Чопоров, А. А. Лисняк, Ю. А. Борисовская, О. С. Козлова, Л. С. Снежкова // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 2. С. 237–250.

89. Методы построения дискретных моделей: структурированные и блочно-структурированные сетки / С. В. Чопоров, С. И. Гоменюк, Х. Х. Алатамнех, К. С. Оспищев // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2016. № 1. С. 272–284.

90. Моделирование напряженно-деформированного состояния композитов и конструкций на их основе / С. Н. Гребенюк, С. И. Гоменюк, Е. Л. Мизерная, В. В. Киричевский // Радіоелектроніка. Інформатика. Управління. 2006. № 2. С. 50–54.

91. Модель напряженно-деформированного состояния тонкостенных конструкций ракетной техники с учетом температурных деформаций / В. З. Грищак,

С. И. Гоменюк, С. Н. Гребенюк, С. В. Чопоров, Д. В. Акимов // АВИА-2017: материалы XIII Міжнародної науково-технічної конференції (м. Київ, 19–21 квітня 2017 р.). Київ: Національний авіаційний університет, 2017. С. 7.41–7.45.

92. Модель напружено-деформованого стану зубчатого колеса / Х. Х. Алатемнех, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омарі, С. В. Чопоров // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 2. С. 173–186.

93. МОНОМАХ 4.2. Примеры расчета и проектирования. Учебное пособие / С. В. Юсипенко, Л. Г. Батрак, Д. А. Городецкий [и др.] Киев: Факт, 2007. 292 с.

94. Мухин В. В., Чопоров С. В. Визуализация дифференциальных характеристик дискретных данных // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 2. С. 60–65.

95. Научное обеспечение проектирования и производства конструкций авиакосмической техники из полимерных композиционных материалов. Сообщение 1 / С. И. Весельский, Ф. М. Гагауз, А. В. Гайдачук [и др.] // Вопросы проектирования и производства конструкций летательных аппаратов. 2009. Вып. 4. С. 7–18.

96. Научное обеспечение проектирования и производства конструкций авиакосмической техники из полимерных композиционных материалов. Сообщение 2 / Ф. М. Гагауз, А. В. Гайдачук, В. Е. Гайдачук [и др.] // Вопросы проектирования и производства конструкций летательных аппаратов. 2011. Вып. 1. С. 7–19.

97. Нефьодов Л. І., Маркозов Д. О., Дерезуцький В. О. Модель автоматизованої системи охорони промислового підприємства // Технологія приборостроєння. 2015. № 1. С. 47–50.

98. Нефьодов Л. І., Філь Н. Ю. Розробка методу оцінки небезпеки зсувів на ділянках автомобільних доріг в умовах нечіткої інформації // Scientific Journal «ScienceRise». 2014. № 1. С. 25–31.

99. Норенков И. П., Маничев В. Б. Основы теории и проектирования САПР: учебник для вузов по специальности «Вычислительные машины, комплексы, системы и сети». Москва: Высшая школа, 1990. 335 с.

100. Норенков И. П. Автоматизированное проектирование: учебник. Москва: Издательство МГТУ им. Н. Э. Баумана, 2000. 188 с.
101. Норенков И. П. Основы автоматизированного проектирования: учебник для вузов. Москва: Издательство МГТУ им. Н. Э. Баумана, 2009. 430 с.
102. Овчинников Н. А. Конечно-элементный анализ напряженно-деформированного состояния элементов поперечных силовых сечений кузова автобуса в эксплуатации // Инженерный вестник Дона. 2013. № 2(25). С. 9–20.
103. Оспищев К. С., Чопоров С. В. Моделирование напряженно-деформированного состояния эластомерных конструкций на базе современных информационных технологий // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 89.
104. Підгорний О. Л. Геометричні основи використання відбитого сонячного випромінювання для інсоляції в архітектурі // Прикладна геометрія та інженерна графіка. 2012. Вип. 90. С. 237–244.
105. Підгорний О. Л. Можливості застосування алгебраїчних торсів однакового схилу в моделюванні задач інсоляції // Енергоефективність в будівництві та архітектурі. 2013. Вип. 4. С. 205–209.
106. Підгорний О. Л., Плоский В. О., Сергійчук О. В. Актуальні проблеми геометричного моделювання в задачах енергозбереження у будівництві // Вентиляція, освітлення та теплогазопостачання. 2010. Вип. 14. С. 25–31.
107. Підгорний О. Л. Поверхні сонячних променів як особливий випадок торсових поверхонь однакового схилу // Прикладна геометрія та інженерна графіка. 2013. Вип. 91. С. 333–340.
108. Плоский В. О., Скочко В. І. Геометричне моделювання структури іонних кристалів // Чернігівський науковий часопис Чернігівського державного інституту економіки і управління. Техніка і природа. 2012. Вип. 1. С. 100–106.
109. Плоский В. О., Скочко В. І. Побудова дискретних каркасів поверхонь із використанням рівнянь параметрів стану та положення в'язей сітчастих структур // Сучасні технології в машинобудуванні та транспорті. 2014. № 2. С. 94–103.

110. Плоский В. О., Скочко В. І. Визначення НДС рідкого діелектрика під дією електростатичного поля // Містобудування та територіальне планування. 2010. Вип. 38. С. 313–326.

111. Плоский В. А., Толок А. В., Бондарь Е. А. Визуально-геометрическое представление потокораспределения ресурсов в СТС «Прикладная геометрия» // Геометрическое и компьютерное моделирование. 2007. Вып. 16. С. 162–168.

112. Построение дискретных моделей геометрических объектов: монография / В. М. Колодяжный, Д. А. Лисин, А. А. Лисняк, С. В. Чопоров. Харьков: ХНАДУ, 2013. 268 с.

113. Проектирование малогабаритных крыльев из композитов / П. М. Гагауз, Ф. М. Гагауз, Я. С. Карпов [и др.] // Вопросы проектирования и производства конструкций летательных аппаратов. 2012. Вып. 2. С. 7–19.

114. Рвачев В. Л., Ющенко Е. Л. Некоторые вопросы аналитического описания геометрических объектов сложной логической системы. Киев: Общество «Знание» УССР, 1965. 53 с.

115. Рвачев В. Л. Геометрические приложения алгебры логики. Київ: Техніка, 1967. 212 с.

116. Рвачев В. Л. Теория R-функций и некоторые ее приложения. Київ: Наукова Думка, 1982. 552 с.

117. Рвачев В. Л., Шевченко А. Н. Проблемно-ориентированные языки и системы для инженерных расчетов. Київ: Техніка, 1988. 198 с.

118. Рвачев В. Л., Суворова И. Г., Шейко Т. И. Численная реализация задачи о течении вязкой жидкости методом R-функций в системе ПОЛЕ-3 // Мат. методы и физ.-мех. поля. 1991. № 33. С. 95–99.

119. Рвачев В. Л., Шапиро В., Шейко Т. И. Метод R-функций (RFM) в краевых задачах с геометрической и физической симметрией // Математичні методи та фізико-механічні поля. 1998. № 1. С. 146–159.

120. Рвачев В. Л., Шапиро В., Шейко Т. И. Применение метода R-функций к построению уравнений локусов, обладающих симметрией // Электромагнитные волны и электронные системы. 1999. № 4. С. 4–20.

121. Рвачев В. Л., Уваров Р. А., Шейко Т. И. Построение уравнений локусов в 3D с помощью R-функций // Радиоэлектроника и информатика. 2001. № 1. С. 158–164.
122. Рвачев В. Л., Шейко Т. И. Введение в теорию R-функций // Проблемы машиностроения. 2001. Т. 4. № 1–2. С. 46–58.
123. Рвачев В. Л., Уваров Р. А., Шейко Т. И. Интерлокационные формулы при моделировании задач Плато и Софи Жермен // Радиоэлектроника и информатика. 2002. № 3. С. 126–128.
124. Рябченко В. М., Лавриненко В. А., Первак В. Д. Проектировочная модель несущей конструкции фюзеляжа пассажирского самолета // Вопросы проектирования и производства конструкций летательных аппаратов. 2011. Вып. 2. С. 42–48.
125. Скворцов А. В. Триангуляция Делоне и её применение. Томск : Издательство Томского университета, 2002. 128 с.
126. Скворцов А. В. Обзор алгоритмов построения триангуляции Делоне // Вычислительные методы и программирование. 2002. № 3. С. 14–39.
127. Скворцов А. В., Мирза Н. С. Алгоритмы построения и анализа триангуляции. Томск : Издательство Томского университета, 2006. 168 с.
128. Сковпень А. В. Усовершенствованный алгоритм построения нерегулярных четырехугольных сеток. Снежинск: РФ ЯЦ-ВНИИТФ, 2004. 62 с.
129. Сковпень А. В. Усовершенствованный алгоритм построения нерегулярных четырехугольных сеток // Журнал вычислительной математики и математической физики. 2005. Т. 45. № 8. С. 1506–1528.
130. Снежкова Л. С., Чопоров С. В. Дискретні моделі в контактних задачах на базі теорії R-функцій // Актуальні проблеми математики та інформатики: збірка тез доповідей V Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 24–25 квітня 2014 р.). Запоріжжя: ЗНУ, 2014. 147 с.
131. Снежкова Л. С., Чопоров С. В. Методы и подходы к моделированию геометрических объектов в контактных задачах // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2014. № 1. С. 171–195.

132. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Актуальні програми математики та інформатики: збірка тез доповідей VI Всеукраїнської наукової конференції молодих дослідників (м. Запоріжжя, 23–24 квітня 2015 р.). Запоріжжя: ЗНУ, 2015. С. 90–91.

133. Снежкова Л. С., Чопоров С. В. Функциональное представление контактного взаимодействия тел // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2015. № 3. С. 239–248.

134. Соколовський Я. І., Сінкевич О. В., Криштапович В. І. Розроблення автоматизованої системи моделювання та прогнозування фізичних процесів у камерах сушіння деревини // Науковий вісник НЛТУ України. 2017. Т. 27, № 1. С. 186–196. DOI: 10.15421/40270145

135. Сохацький О. А. Математичне моделювання та дослідження теплообмінних процесів у периферійній зоні доменної печі // Вісник Академії митної служби України. Технічні науки. 2012. № 1. С. 144–154.

136. Стоян Ю. Г., Панкратов А. В., Романова Т. Е. Математическое моделирование ограничений на допустимые расстояния между геометрическими объектами // Кибернетика и системный анализ. 2012. Т. 48, № 3. С. 12–17.

137. Стоян Ю. Г., Пацук В. Н. Метод покрытия выпуклого многогранного множества минимальным количеством одинаковых шаров // Доповіді Національної академії наук України. 2009. № 5. С. 41–45.

138. Стоян Ю. Г., Сосюрка Е. С. Покрытие компактной многогранной области конечным семейством прямых параллелепипедов // Доповіді Національної академії наук України. 2010. № 8. С. 43–48.

139. Стоян Ю. Г., Яковлев С. В. Конфигурационное пространство геометрических объектов // Кибернетика и системный анализ. 2018. Т. 54, № 5. С. 38–50.

140. Термопружне деформування композитної пластини / В. В. Киричевский, В. О. Толок, С. І. Гоменюк, С. М. Гребенюк // Систем. Технології. 2003. № 4. С. 109–113.

141. Тимирязев В. А., Беянкина О. В., Серебряков А. А. Автоматизированное проектирование технологических процессов с использованием ЭВМ // Горный информационно-аналитический бюллетень (научно-технический журнал). 2009. № 10. С. 220–222.

142. Тимошенко С. П. Гудьер Дж. Теория упругости. Москва: Наука, 1975. 576 с.

143. Толок А. В. Синтез компьютерных образов геометрических характеристик для оценки рельефа поверхности функции двух переменных // Доповіді Національної академії наук України. 2004. №4. С. 63–69.

144. Толок А. В. Графические образы-модели в информационных технологиях // Прикладная информатика. 2009. № 4(22). С. 31–41.

145. Толок А. В. Using voxel models in automation of mathematical modeling // Automation and Remote Control. 2009. Т. 70. № 6. С. 1067–1079.

146. Толок А. В., Толок Н. Б. Решение задач математического программирования функционально-воксельным методом // Проблемы управления. 2017. № 3. С. 37–42.

147. Толок А. В., Толок Н. Б. Основы аналитического проектирования на функционально-воксельных моделях // Информационные технологии в проектировании и производстве. 2016. № 4 (164). С. 15–23.

148. Толок А. В. Функционально-воксельный метод в компьютерном моделировании. М.: Физматлит, 2016. 112 с.

149. Толок В. А., Шапар В. В. Операторно-символьные ряды Власова В. З. в решении задач теории упругости в системе MAPLE // Гидроакустический журнал. 2006. № 3. С. 66–74.

150. Толок В. А., Шапар В. В. Моделирование решения задач теории упругости операторно-символьными рядами в системе MAPLE // Проблемы машиностроения. 2009. Т. 12. № 1. С. 86–91.

151. Толок Н. Б., Толок А. В. Функционально-воксельный метод компьютерных вычислений // Научная визуализация. 2017. Т.9, № 2. С. 1–12.

152. Триангуляция поверхностей при гибридном параметрическом и неявном представлении / С. В. Чопоров, С. И. Гоменюк, А. А. Лисняк, М. А. В. Аль-Омари, Х. Х. Алатамнех // Вестник Херсонского национального технического университета. 2016. № 3(58). С. 542–549.

153. Физические поля с геометрическими сингулярностями / В. Л. Рвачев, Т. И. Шейко, В. Шапиро [и др.] // Радиоэлектроника и информатика. 2002. № 3. С. 22–28.

154. Флетчер К. Вычислительные методы в динамике жидкостей. Москва: Мир, 1991. Т. 2: Методы расчета различных течений. 552 с.
155. Флэнаган Д. JavaScript. Подробное руководство. Санкт-Петербург: Символ-Плюс, 2008. 992 с.
156. Формалізація геометричної моделі методу двох зображень Підгорного засобами точкового числення Балюби-Найдиша / О. Л. Підгорний, А. В. Найдиш, Є. В. Конопацький [та ін.] // Прикладна геометрія та інженерна графіка. 2013. Вип. 91. С. 325–332.
157. Формализация моделей многослойных тонкостенных конструкций в САПР / С. В. Чопоров, С. И. Гоменюк, Б. Г. М. Аль-Атемнех, М. А. В. Аль-Омари, Х. Х. Алатемнех // Вісник Національного технічного університету «ХПІ». Збірник наукових праць. Серія: Нові рішення в сучасних технологіях. 2015. № 62(1171). С. 99–106.
158. Функциональный поход к геометрическому моделированию технических систем : монография / С. В. Чопоров, С. Н. Гребенюк, С. И. Гоменюк [и др.] Запорожье: Запорожский национальный университет, 2016. 176 с.
159. Черников А. В. Геометрические модели в исследованиях физических и технологических процессов // Геометричне та комп'ютерне моделювання: Збірник наук. праць. 2006. Вип. 17. С. 159–163.
160. Черніков О. В., Архіпов О. В., Грайворонський Є. С. Оптимізація геометрії елементів ДВЗ методами комп'ютерного моделювання (на прикладі двигуна МТ-801) // Геометричне та комп'ютерне моделювання: Збірник наук. праць. 2009. Вип. 22. С. 43–47.
161. Черніков О. В., Грайворонський Є. С. Комп'ютерне моделювання геометричних параметрів та процесів в робочій камері двигуна для удосконалення його конструкції // Геометричне та комп'ютерне моделювання: Збірник наук. праць. 2009. Вип. 24. С. 11–15.
162. Черніков О. В., Грайворонський Є. С. Комп'ютерне моделювання процесів в камері згоряння двигуна // Міжвідомчий науково-технічний збірник «Прикладна геометрія та інженерна графіка». 2010. Вип. 85. С. 244–249.

163. Черніков О. В. Комп'ютерне моделювання та аналіз кінематичних особливостей робочого обладнання фронтального навантажувача // Міжвідомчий науково-технічний збірник «Прикладна геометрія та інженерна графіка». 2010. Вип. 86. С. 107–111.

164. Черніков О. В., Москаленко А. І. Розробка параметричних моделей з урахуванням динамічних та міцнісних характеристик // Прикладна геометрія та інженерна графіка. 2011. Вип. 87. С. 431–434.

165. Чисельне моделювання напружено-деформованого стану еластомерних елементів сит грохотів у САПР «Мірела+» / В. В. Киричевський, С. М. Гребенюк, С. І. Гоменюк, Р. В. Киричевський // Радіoeлектроніка. Інформатика. Управління. 2005. № 2. С. 53–57.

166. Чопоров С. В. Автоматизация анализа температурной устойчивости пластин с использованием метода конечных элементов // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2017. № 2. С. 304–316.

167. Чопоров С. В., Акимов Д. В. Использование элементов связи при функциональном подходе к заданию нагрузок в моделях оболочечных конструкций // інформатика, управління, штучний інтелект: матеріали IV Міжнародної конференції студентів, магістрів та аспірантів (м. Харків, 21–23 листопада 2017 р). Харків: НТУ «ХП», 2017. С. 105.

168. Чопоров С. В. Алгебраические способы учета перемещения узла сетки конечных элементов // Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки. 2013. № 1. С. 117–121.

169. Чопоров С. В., Аль-Атемнех Б. Г. М., Гоменюк С. І. Підхід до геометричного моделювання в паралельних обчислювальних системах зі спільною пам'яттю // Проблеми обчислювальної механіки і міцності конструкцій. 2017. Вип. 27. С. 164–172.

170. Чопоров С. В. Гибридное представление сплошных тел с использованием неявных и параметрических функций // Радиоэлектроника, информатика, управление. 2017. № 3. С. 60–70.

171. Чопоров С. В. Гибридный подход к представлению геометрических объектов // Сучасні проблеми машинобудування: тези доповідей конференції молодих вчених та спеціалістів (м. Харків, 21–24 листопада 2016 р.). Харків: Інститут проблем машинобудування ім. А. Н. Підгорного НАН України, 2016. С. 36.

172. Чопоров С. В., Гоменюк С. И., Лисняк А. А. Сравнительный анализ треугольных и четырехугольных конечных элементов // Вестник Херсонского национального технического университета. 2013. № 2(47). С. 382–386.

173. Чопоров С. В., Гоменюк С. И. Параллельный способ построения сеток треугольных элементов при функциональном представлении // Вестник Херсонского национального технического университета. 2015. № 3(54). С. 511–517.

174. Чопоров С. В. Использование технологий параллельных вычислений в методе конечных элементов // Радиоэлектроника, информатика, управление. 2013. № 2(29). С. 88–94.

175. Чопоров С. В., Лисняк А. А., Гоменюк С. И. Подход к определению областей сгущения сетки конечных элементов на базе теории R-функций // Методи розв'язування прикладних задач механіки деформівного твердого тіла. Дніпропетровськ: Ліра. 2012. Вип. 13. С. 413–419.

176. Чопоров С. В. Метод фоновой сетки для триангуляции двумерных областей при функциональном подходе // Радиоэлектроника, информатика, управление. 2015. № 4(35). С. 31–38.

177. Чопоров С. В. Построение неравномерных дискретных сеток для функциональных математических моделей на базе теории R-функций // Радиоэлектроника, информатика, управление. 2011. № 2. С. 70–75.

178. Чопоров С. В. Представление вычислительных сеток в системах инженерного анализа // Вісник Херсонського національного технічного університету. 2018. Т. 1, № 3(66). С. 372–379.

179. Чопоров С. В. Проекційний підхід до генерації поверхневих сіток при функціональному поданні тіл // Сучасні проблеми механіки та математики: збірник наукових праць (м. Львів, 22–25 травня 2018 р.) / ред. А. М. Самойленка та Р. М. Кушніра. 2018. Т. 1. С. 139–140. URL: www.iapmm.lviv.ua/mpmm2018 (дата звернення: 18.09.2018).

180. Чопоров С. В. Проблемно-орієнтована мова для моделювання геометричних об'єктів при функціональному підході // Вестник Херсонского национального технического университета. 2014. № 3(50). С. 532–539.
181. Чопоров С. В. Сглаживание сеток четырехугольных элементов с использованием локальной минимизации функционала // Вестник Херсонского национального технического университета. 2017. Т. 2, № 3(62). С. 234–239.
182. Чопоров С. В. Дискретные модели технических объектов: монография. Херсон: Видавничий дім «Гельветика», 2018. 324 с.
183. Шевченко Е. В., Танасогло А. В. Оптимальное проектирование конструкций башенных анкерно-угловых опор ВЛ 110 кВ // Вісник Донбаської національної академії будівництва і архітектури. 2013. Вип. 3. С. 114–116.
184. Agoston M. K. Computer Graphics and Geometric Modelling: Implementation and Algorithms. London : Springer-Verlag, 2005. 908 p.
185. Ahusborde E., Glockner S. A 2D Block-structured Mesh Partitioner for Accurate Flow Simulations on Non-rectangular Geometries // Computers & Fluids. 2011. Vol. 43. P. 2–13. DOI: 10.1016/j.compfluid.2010.07.009
186. Akinlar M. A., Salako S., Liao G. A Method for Orthogonal Grid Generation // General Mathematics Notes. 2011. Vol. 3, No 1. P. 55–72.
187. Akcelik V., Jaramaz B., Ghattas O. Nearly Orthogonal Two Dimensional Grid Generation with Aspect Ratio Control // Journal of Computational Physics. Vol. 171, Iss. 2. P. 805–821. DOI: 10.1006/jcph.2001.6811
188. Akkouche S., Galin E. Adaptive Implicit Surface Polygonization using Marching Triangles // Computer Graphics Forum. 2001. Vol. 20, Iss. 2. P. 67–80. DOI: 10.1111/1467-8659.00479
189. Alumbaugh T. J., Jiao X. Compact array-based mesh data structure // International Meshing Roundtable: Proceedings of The 14th International Conference (San Diego, California, USA, 11–14 September 2005). Sandia National Laboratories, Springer-Verlag, 2005. P. 585–504.
190. Alexandrescu A. Modern C++ Design: Generic Programming and Design Patterns Applied. USA: Addison-Wesley, 2001. 352 p.

191. Allen J. D. The Unicode Standard / the Unicode Consortium. Mountain View: the Unicode Consortium, 2015. 1016 p.
192. Amenta N., Bern M, Eppstein D. Optimal Point Placement for Mesh Smoothing // Journal of Algorithms. 1999. Vol. 30, Iss. 2. P. 302–322. DOI: 10.1006/jagm.1998.0984
193. Analytical and Numerical Modelling of the Axisymmetric Bending of Circular Sandwich Plates with the Nonlinear Elastic Core Material / A. Kudin, S. Choporov, Yu. Tamurov, M. A. V. Al Omari // Journal of Solid Mechanics. 2016. Vol. 8, Iss. 3. P. 590–601.
194. Anderson B. D., Benzley S. E., Owen S. J. Automatic All Quadrilateral Mesh Adaption through Refinement and Coarsening // International Meshing Roundtable: Proceedings of The 18th International Conference (Salt Lake City, UT, USA, 25–28 October 2009). Sandia: Sandia National Laboratories, 2009. P. 557–574. DOI: 10.1007/978-3-642-04319-2_32
195. ANSYS LS-DYNA. URL: <http://www.ansys.com/Products/Structures/ANSYS-LS-DYNA> (дата звернення 15.08.2016).
196. ANSYS LS-DYNA. URL: <http://cae-expert.ru/product/ansys-ls-dyna> (дата звернення 15.08.2016).
197. Ashrafizadeh A., Raithby G. D. Direct Design Solution of the Elliptic Grid Generation Equations // Numerical Heat Transfer. 2006. Vol. 50, Iss. 3. P. 217–230. DOI: 10.1080/10407790600599009
198. Applying Parmetis to Structured Remeshing for Industrial CFD Applications / S. Laflamme, J. Dompierre, F. Guibault, R. Roy // The International Journal of High Performance Computing Applications. 2003. Vol. 17, No 1. P. 63–76.
199. Ashrafizadeh A., Jalalabadi R. Structured Grid Generation Via Constraint on Displacement of Internal Nodes // International Journal of Basic & Applied Science. 2011. Vol. 11, No 4. P. 79–87.
200. A New Approach to the Development of Automatic Quadrilateral Mesh Generation / J. Z. Zhu, O. C. Zienkiewicz, E. Hinton, J. Wu // International Journal For Numerical Methods in Engineering. 1991. Vol. 32, Iss. 4. P. 849–866. DOI: 10.1002/nme.1620320411

201. A new automated scheme of quadrilateral mesh generation for randomly distributed line constraints / C. Park, J.-S. Noh, I.-S. Jang, J. M. Kang // *Computer-Aided Design*. 2007. Vol. 39. P. 258–267.
202. A two-level optimization feature for the design of aerospace structures / E. Carrera, L. Mannella, G. Augello, N. Gualtieri // *Journal of Aerospace Engineering*. 2003. Vol. 217, No 4. P. 189–206.
203. Baker B. S., Grosse E., Rafferty C. S. Nonobtuse Triangulation of Polygons // *Discrete and Computational Geometry*. 1988. Vol. 3, Iss. 2. P. 147–168.
204. Bank R. E. PLTMG: A Software Package for Solving Elliptic Partial Differential Equations: Users' Guide 8.0. SIAM, 1998. 155 p. DOI: 10.1137/1.9780898719635
205. Barrera-Sánchez P., González-Flores G. F., Dominguez-Mota F. J. Robust Discrete Grid Generation on Plane Irregular regions // *Computational Mathematics and Mathematical Physics*. 2003. Vol. 43, Iss. 6. P. 845–853.
206. Barth T. J., Jespersen D. C. The design and application of upwind schemes on unstructured meshes // *The 27th Aerospace Sciences Meeting: Proceedings of Multidisciplinary Meeting* (Reno, NV, USA, 9–2 January 1989). Reno: AIAA, 1989. AIAA PAPER 89-0366. 13 p. DOI: 10.2514/6.1989-366
207. Baumgart B. G. A polyhedron representation for computer vision // *AFIPS'75: Proceedings of National Computer Conference and Exposition* (New York, USA, 1975). ACM, 1975. P. 589–596.
208. Bern M., Eppstein D. Mesh generation and optimal triangulation // *Computing in Euclidean Geometry*. 1992. No 1. P. 23–90.
209. Bern M., Eppstein D., Gilbert J. Provably good mesh generation // *Journal of Computer and System Sciences*. 1994. Vol. 48, Iss. 3. P. 384–409.
210. Bern M., Dobkin D., Eppstein D. Triangulating polygons without large angles // *International Journal of Computational Geometry & Applications*. 1995. Vol. 5. P. 171–192.
211. Bern M., Michell S., Ruppert J. Linear-size nonobtuse triangulation of polygons // *Discrete & Computational Geometry*. 1995. Vol. 14, Iss. 4. P. 411–428.

212. Blacker T. D., Stephenson M. B. Paving: A new approach to automated quadrilateral mesh generation // International Journal For Numerical Methods in Engineering. 1991. Vol. 32, Iss. 4. P. 811–847. DOI: 10.1002/nme.1620320410
213. Blacker T. D., Meyers R. J. Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm // Engineering with Computers. 1993. Vol. 9, Iss. 2. P. 83–93.
214. Blilie C. Patterns in scientific software: an introduction // Computing in Science & Engineering. 2002. Vol. 4, Iss. 3. P. 48–53. DOI: 10.1109/5992.998640
215. Blinn J. F. A Generalization of Algebraic Surface Drawing // ACM Transactions on Graphics. 1982. Vol. 1, Iss. 3. P. 235–256. DOI: 10.1145/357306.357310
216. Buhmann M. D. Radial Basis Functions: Theory and Implementations. Cambridge: Cambridge University Press, 2004. 271 P.
217. Busé L., Cox D., D'Andrea C. Implicitization of Surfaces in P^3 in the Presence of Base Points // Journal of Algebra and Its Applications. 2003. Vol. 2, No 2. P. 189–214.
218. Boubekur T., Schlick C. A Flexible Kernel for Adaptive Mesh Refinement on GPU // Computer Graphics Forum. 2008. Vol. 27, Iss. 1. P. 102–114. DOI: 10.1111/j.1467-8659.2007.01040.x
219. Buckling Analysis of Circular Sandwich Plates with the Nonlinear Elastic Core Material / O. Kudin, S. Choporov, S. Gomenyuk, V. Gristchak // Mechanika 2018: Proceedings of 23rd international conference at Kaunas University of Technology (Kaunas, Lithuania, 18 May), 2018. Kaunas University of Technology, 2018. P. 92–100.
220. Campagna S., Kobbelt L., Seidel H.-P. Directed Edges: A Scalable Representation for Triangle Meshes // Journal of Graphics Tools. 1998. Vol. 3, Iss. 4. P. 1–11.
221. Canann S.A., Tristano J. R., Staten M. L. An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes // International Meshing Roundtable: Proceedings of the 7th International Conference (Dearborn, Michigan, U.S.A., 26–28 October 1998). Sandia: Sandia National Laboratories, 1998. P. 479–494.
222. Castillo J. E. An adaptive direct variational grid generation method // Computers & Mathematics with Applications. 1991. Vol. 21, Iss. 5. P. 57–64.
223. Čermák M., Skala V. Polygonisation of disjoint implicit surfaces by the adaptive edge spinning algorithm // International Journal of Computational Science and Engineering. 2007. Vol. 3, No 1. P. 45–52. DOI: 10.1504/IJCSE.2007.014464

224. Chf: A scalable topological data structure for tetrahedral meshes / M. Lage, T. Lewiner, H. Lopes, L. Vehlo // Computer Graphics and Image Processing: Proceedings of the XIII Brazilian Symposium (Washington, DC, USA, 2005). IEEE Computer Society, 2005. P. 349–356.
225. CGALmesh: a Generic Framework for Delaunay Mesh Generation / C. Jamin, P. Alliez, M. Yvinec, J.-D. Boissonnat // ACM Transactions on Mathematical Software. 2014. Vol. 1, No 1. P. 1–26.
226. Chen L. Mesh Smoothing Schemes Based on Optimal Delaunay Triangulations // International Meshing Roundtable: Proceedings of the 13th International Conference (Williamburg, U.S.A., 19–22 September 2004). Sandia: Sandia National Laboratories, 2004. P. 109–120.
227. Chen F., Cox D., Liu Y. The μ -basis and implicitization of a rational parametric surface // Journal of Symbolic Computation. 2005. Vol. 39. P. 689–706.
228. Chen H., Bishop J. Delaunay Triangulation for Curved Surfaces // International Meshing Roundtable: Proceedings of the 6th International Conference (Park City, UT, USA, 13–15 October 1997). Sandia: Sandia National Laboratories, 1997. P. 115–127.
229. Chen M., Tucker J. V. Constructive Volume Geometry // Computer Graphics Forum. 2000. Vol. 19, Iss. 4. P. 281–293. DOI: 10.1111/1467-8659.00464
230. Chen Z. Combined Laplacian and Optimization-based Smoothing for Quadratic Mixed Surface Meshes // International Meshing Roundtable: Proceedings of the 12th International Conference (Santa Fe, U.S.A., 14–17 September 2003). Sandia: Sandia National Laboratories, 2003. P. 201–213.
231. Cheng S.-W. On the sizes of Delaunay meshes // Computational Geometry – Theory and Applications. 2006. Vol. 33, Iss. 3. P. 130–138.
232. Cheng S.-W., Dey T. K., Shewchuk J. R. Delaunay mesh generation. Boca Raton, FL: CRC Press, 2013. 387 p.
233. Chernyaev E. V. Marching Cubes 33: Construction of Topologically Correct Isosurfaces. Technical Report CERN CN 95-17. CERN, 1995. 8 p.
234. Chew L. P. Constrained Delaunay Triangulations // Algorithmica. 1989. Vol. 4, Iss. 1. P. 97–108.

235. Chew L. P. Guaranteed-Quality Triangular Meshes. Technical Report TR-89-983. Ithaca, NY: Cornell University, 1989. 20 p.

236. Chew L. P. Guaranteed-Quality Mesh Generation for Curved Surfaces // The 9th Annual Symposium on Computational Geometry: Proceedings of the International Conference (New York, NY, USA, 18–21 May 1993). New York: ACM, 1993. P. 274–280.

237. Chinnaswamy C., Amadei B., Illangasekare T. H. A new method for finite element transitional mesh generation // International Journal for Numerical Methods in Engineering. 1991. Vol. 31, Iss. 7. P. 1253–1270. DOI: 10.1002/nme.1620310703

238. Choporov S., Homenyuk S, Grebenyuk S. Optimized Smoothing of Discrete Models of the Implicitly Defined Geometrical Objects' Surfaces // Eastern-European Journal of Enterprise Technologies. 2018. Vol. 3/4 (93). P. 52–60.

239. Choporov S. V., Lisnyak A. O., Akimov D. V. Discrete models generation for layered shells of a spacecraft // Aviation in the XXI-st Century. Safety in Aviation and Space Technologies: Proceedings The Sixth World Congress (Kyiv, Ukraine, 23–25 September, 2014). Ukraine, Kyiv: NAU, 2014. V. 1. P. 1.14.45–1.14.48.

240. Choporov S. Parallel Computing Technologies in the Finite Element Method // High Performance Computing: Proceedings of Third International Conference HPC-UA 2013 (Kyiv, Ukraine, 7–11 October, 2013). Kyiv: National Technical University of Ukraine "Kyiv Polytechnic Institute", 2013. P. 85–91.

241. Cickovski Tr., Matthey Th., Izaguirre J. A. Design Patterns for Generic Object-Oriented Scientific Software. Technical Report TR 2004-29. Notre Dame: University of Notre Dame, 2004. 9 P.

242. Circular incident edge list: a data structure for rendering complex unstructured grids / B. Levy, G. Caumon, S. Conreux, X. Cavin // Conference on Visualization: Proceedings of the Conference on Visualization (Washington, DC, USA, 2001). IEEE Computer Society, 2001. P. 191–198.

243. Cohel-Steiner D., de Verdière É. C., Yvinec M. Conforming Delaunay Triangulations in 3D // Computational Geometry: Theory and Applications. 2004. Vol. 28. P. 217–233.

244. Combinatorial configurations in balance layout optimization problems / I. V. Grebennik, A. A. Kovalenko, T. E. Romanova [and others] // Cybernetics and Systems Analysis. 2018. Vol. 54, Iss. 2. P. 221–231. DOI: 10.1007/s10559-018-0023-2

245. Compact representation of simplicial meshes in two and three dimensions / D. K. Blandford, G. E. Blueloch, D. E. Cardoze, C. Kadow // International Journal of Computational Geometry and Applications. 2005. Vol. 15, Iss. 1. P. 3–24.
246. Composite mappings for planar mesh generation / R. H. Crawford, W. N. Waggenspack Jr., D. C. Anderson, M. J. Bailey // International Journal for Numerical Methods in Engineering. 1987. Vol. 24, Iss. 12. P. 2241–2252. DOI: 10.1002/nme.1620241202
247. Compression Strength of Stitched Laminates after Low-velocity Impact / C. Xiaoquan, A. M. Al-Mansour, L. Zhengneng, K. Chenghe // Journal of Reinforced Plastics and Composites. 2005. Vol. 24, No 9. P. 935–947.
248. Coplien J., Hoffman D., Weiss D. Commonality and variability in software engineering // IEEE Software. 1998. Vol. 15, Iss. 6. P. 37–45.
249. Cook W. A. Body oriented (natural) co-ordinates for generating three dimensional meshes // International Journal for Numerical Methods in Engineering. 1974. Vol. 8, Iss. 1. P. 27–43. DOI: 10.1002/nme.1620080104
250. Crespin B., Guitton P., Schlick C. Efficient and Accurate Tessellation of Implicit Sweep Objects // Constructive Solid Geometry. 1998. P. 49–63.
251. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. USA: Addison-Wesley, 1994. 395 p.
252. Design Patterns for Object-Oriented Scientific Software / S. Choporov, S. Gomenyuk, O. Kudin, A. Lisnyak // CEUR Workshop Proceedings. 2018. Vol. 2105. P. 441–444.
253. Dey T. K., Levine J. A. Delaunay meshing of isosurfaces // The Visual Computer. 2008. Vol. 24, Iss. 6. P. 411–422. DOI: 10.1007/s00371-008-0224-1
254. Du Q., Wang D. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations // International Journal for Numerical Methods in Engineering. 2003. Vol. 56, Iss. 9. P. 1355–1373. DOI: 10.1002/nme.616
255. Du Q., Wang D. Recent progress in robust and quality Delaunay mesh generation // Journal of Computational and Applied Mathematics. 2006. Vol. 195, Iss. 1-2. P. 8–23.

256. Dynamic modeling and simulation of a two-stage series-parallel vibration isolation system / R. Guo, S.-Qi. Zhou, M.-Jia. Wang, S. Qui // *Advances in Mechanical Engineering*. 2016. Vol. 8(8). P. 1–19.

257. ECMAScript Language Specification. URL: <http://www.ecma-international.org/ecma-262/5.1/> (дата звернення 15.08.2016).

258. Efficient Computation of 3D Clipped Voronoi Diagram / D.-M. Yan, W. Wang, B. Lévy, Y. Liu // *Lecture Notes in Computer Science*. 2010. Vol. 6130. P. 269–282.

259. Efficient Implementation of Marching Cubes' Cases with Topological Guarantees / T. Lewiner, H. Lopes, A. W. Vieira, G. Tavares // *Journal of Graphics Tools*. 2003. Vol. 8, Iss. 2. P. 1–15. DOI: 10.1080/10867651.2003.10487582

260. Efficient 3D modeling of damage in composite materials / El-Sisi A. E.-D. A., H. M. El-Emam, H. A. Salim, H. E.-D. M. Sallam // *Journal of Composite Materials*. 2015. Vol. 49, No 7. P. 817–828.

261. Eiseman P. R. A multi-surface method of coordinate generation // *Journal of Computational Physics*. 1979. Vol. 33, Iss. 1. P. 118–150.

262. El-Hamalawi A. A 2D combined advancing front-Delaunay mesh generation scheme // *Finite Elements in Analysis and Design*. 2004. Vol. 40, No 9-10. P. 967–989.

263. Fayolle P. A., Pasko A. Optimized surface discretization of functionally defined multi-material objects // *Advances in Engineering Software*. 2012. Vol. 45, Iss. 1. P. 301–312. DOI: 10.1016/j.advengsoft.2011.10.007

264. Fazil J., Jayakumar V. Investigation of Airfoil Profile Design using Reverse Engineering Bezier Curve // *ARPN Journal of Engineering and Applied Sciences*. 2011. Vol. 6, No 7. P. 43–52.

265. Feature Sensitive Surface Extraction from Volume Data / L. P. Kobbelt, M. Botsch, U. Schwannecke, H.-P. Seidel // *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques (Los Angeles, CA, USA, 12–17 August 2001)*. ACM Press, 2001. P. 57–66. DOI: 10.1145/383259.383265

266. Field D. A. Laplacian Smoothing and Delaunay Triangulation // *Communications in Applied Numerical Methods*. 1988. Vol. 4. P. 709–712.

267. Fix G., Hsu C.-P., Luo T. Implicitization of Rational Parametric Surfaces // *Journal of Symbolic Computation*. 1996. Vol. 21. P. 329–326.

268. Floater M. S. Parametrization and smooth approximation of surface triangulations // *Computer Aided Geometric Design*. 1997. Vol. 14, Iss. 3. P. 231–250.
269. Floater M. S. Mean value coordinates // *Computer Aided Geometric Design*. 2003. Vol. 20, Iss. 1. P. 19–27.
270. Floater M. S., Hormann K., Kós G. A general construction of barycentric coordinates over convex polygons // *Advances in Computational Mathematics*. 2006. Vol. 24, Iss. 1. P. 311–331.
271. Folwell N. T., Mitchell S. A. Reliable Whisker Weaving via Curve Contraction // *International Meshing Roundtable: Proceedings of the 7th International Conference*, Dearborn (Michigan, U.S.A., 26–28 October 1998). Sandia: Sandia National Laboratories, 1998. P. 365–378.
272. Forming and Resolving Wedges in the Spatial Twist Continuum / T. D. Blacker, S. A. Mitchell, T. J. Tautges [and others] // *Engineering with Computers*. 1997. Vol. 13, Iss. 1. P. 35–47.
273. Freitag L. A., Ollivier-Gooch C. A Comparison of Tetrahedral Mesh Improvement Techniques // *International Meshing Roundtable: Proceedings of the 5th International Conference* (Pittsburgh, Pennsylvania, U.S.A., 10–11 October 1996). Sandia: Sandia National Labs, 1996. P. 87–100.
274. Freitag L. A. On Combining Laplacian and Optimization-based Mesh Smoothing Techniques // *AMD Trends in Unstructured Mesh Generation*. 1997. Vol. 220. P. 37–43.
275. Freitag L. A., Plassmann P. E. Local Optimization-based Untangling Algorithms For Quadrilateral Meshes // *International Meshing Roundtable: Proceedings of the 10th International Conference* (Newport Beach, U.S.A., 7–10 October 2001). Sandia: Sandia National Laboratories, 2001. P. 397–406.
276. Frey P., Borouchaki H., Geouge P.-L. Delaunay tetrahedralization using an advancing-front approach // *International Meshing Roundtable: Proceedings of the 5th International Conference* (Pittsburgh, Pennsylvania, U.S.A., 10–11 October 1996). Sandia: Sandia National Laboratories, 1996. P. 31–48.
277. Frey P., Borouchaki H., Geouge P.-L. 3D Delaunay mesh generation coupled with an advancing-front approach // *Computer methods in applied mechanics and engineering*. 1998. Vol. 157, Iss. 1. P. 115–131.

278. Frey W. H. Selective Refinement: A New Strategy for Automatic Node Placement in Graded Triangular Meshes // International Journal for Numerical Methods in Engineering. 1987. Vol. 24, Iss. 11. P. 2183–2200. DOI: 10.1002/nme.1620241111

279. Frey W. H., Field D. A. Mesh Relaxation: A New Technique for Improving Triangulations // Computer Methods in Applied Mechanics and Engineering. 1991. Vol. 92, Iss. 3. P. 269–288.

280. Fully automatic adaptive mesh refinement integrated into the solution process / J. R. Tristano, Z. Chen, D. A. Hancq, W. Kwok // International Meshing Roundtable: Proceedings of the 12th International Conference (Santa Fe, New Mexico, U.S.A., 14–17 September 2003). Sandia National Laboratories, 2003. P. 307–314.

281. Gabriel-constrained Parametric Surface Triangulation / O. E. Ruiz, C. Cadavid, J. G. Lalinde [and others] // International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering. 2008. Vol. 2, No 8. P. 568–575. DOI: 10.5772/7401

282. George P. L., Hecht F., Saltel E. Automatic Mesh Generator with Specified Boundary // Computer Methods in Applied Mechanics and Engineering. 1991. Vol. 92, Iss. 3. P. 269–288.

283. Geuzaine C., Remacle J.-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities // International Journal for Numerical Methods in Engineering. 2009. Vol. 79, Iss. 11. P. 1309–1331.

284. Goerigk N., Si H. On Indecomposable Polyhedra and the Number of Steiner Points // Procedia Engineering. 2015. Vol. 124. P. 343–355. DOI: 10.1016/j.proeng.2015.10.144

285. Gonzalez-Vega L. Implicitization of Parametric Curves and Surfaces by using Multidimensional Newton Formulae // Journal of Symbolic Computation. 1997. Vol. 23, Iss. 2-3. P. 137–151.

286. Gordon W. N., Hall C. A. Construction of curvilinear coordinate systems and application to mesh generation // International Journal for Numerical Methods in Engineering. 1973. Vol. 7, Iss. 4. P. 461–477. DOI: 10.1002/nme.1620070405

287. Guoy D., Erickson J. Automatic Blocking Scheme for Structured Meshing in 2D Multiphase Flow Simulation // International Meshing Roundtable: Proceedings of the

13th International Conference (Willimasburg, Virginia, USA, 19–22 September 2004). Sandia: Sandia National Laboratories, 2004. P. 121–132.

288. Günel M., Kayran A. Non-linear progressive failure analysis of open-hole composite laminates under combined loading // *Journal of Sandwich Structures and Materials*. 2016. Vol. 15(3). P. 309–339.

289. Haber R., Shephard M. S., Abel J. F. A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mapping // *International Journal for Numerical Methods in Engineering*. 1981. Vol. 17, Iss. 7. P. 1015–1044. DOI: 10.1002/nme.1620170706

290. Haber R., Abel J. F. Discrete transfinite mappings for description and meshing of three-dimensional surfaces using interactive computer graphics // *International Journal for Numerical Methods in Engineering*. 1982. Vol. 18, Iss. 1. P. 41–66. DOI: 10.1002/nme.1620180105

291. Hable J., Rossignac J. CST: Constructive solid trimming for rendering BReps and CSG // *IEEE Transactions on Visualization and Computer Graphics*. 2007. Vol. 13, Iss. 5. P. 1004–1014. DOI: 10.1109/TVCG.2007.70411

292. Hansbo P. Generalized Laplacian Smoothing of Unstructured Grids // *Communications in Numerical Methods in Engineering*. 1995. Vol. 11. P. 455–464.

293. Hartmann E. A Marching Method for the Triangulation of Surfaces // *The Visual Computer*. 1998. Vol. 14. P. 95–108.

294. Hazelwood C. Approximating Constrained Tetrahedrizations // *Computer Aided Geometric Design*. 1993. Vol. 10. P. 67–87.

295. Heikes R., Randall D. Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part I: Basic design and results of tests // *Monthly Weather Review*. 1995. Vol. 123. P. 1862–1880.

296. Heikes R., Randall D. Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part II: A detailed description of the grid and an analysis of numerical accuracy // *Monthly Weather Review*. 1995. Vol. 123. P. 1881–1887.

297. Hilton A., Illingworth J. Marching Triangles: Delaunay Implicit Surface Triangulation. Technical Report EPSRC-GR/K04569. University of Surrey, Guildford, UK, 1997. 12 P.

298. Hoppe H., DeRose T., Duchamp T. Mesh Optimization // SIGGRAPH '93: Proceedings the 20th annual conference on Computer graphics and interactive techniques (Anaheim, CA, USA, 02–06 August 1993). ACM Press, 1993. P. 19–26. DOI: 10.1145/166117.166119

299. How to Subdivide Pyramids, Prisms and Hexahedra into Tetrahedra / J. Dompierre, P. Labbé, M.-G. Vallet, R. Camarero // International Meshing Roundtable: Proceedings of the 8th International Conference (Lake Tahoe, California, USA, 10–13 October 1999). Sandia: Sandia National Laboratories, 1999. P. 195–204.

300. Huang H., Ascher U. Surface Mesh Smoothing, Regularization, and Feature Detection // SIAM Journal on Scientific Computing. 2008. Vol. 31, Iss. 1. P. 74–93. DOI: 10.1137/060676684

301. IEEE Standard for Floating-Point Arithmetic. URL: <http://ieeexplore.ieee.org/servlet/opac?punumber=4610933> (дата звернення: 24.09.2016).

302. Igarashi T., Hughes J. F. Smooth Meshes for Sketch-based Freeform Modeling // SIGGRAPH'06: The 33rd International conference on Computer graphics and interactive techniques (Boston, Massachusetts, USA, 1–3 August 2006). 2006. Article No. 13. DOI: 10.1145/1185657.1185774

303. Isenburg M., Alliez P. Compressing Hexahedral Volume Meshes // Graphical Models. 2003. Vol. 65, Iss. 4. P. 239–257.

304. Ito Y., Shih A. M., Soni B. K. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates // International Journal for Numerical Methods in Engineering. 2009. Vol. 77, Iss. 13. P. 1809–1833. DOI: 10.1002/nme.2470

305. Ives D. Geometric grid generation. NASA CP-3291. Cleveland, 1995. P. 535–546.

306. Jablonowski C., Oehmke R. C., Stout Q. F. Block-structured adaptive meshes and reduced grids for atmospheric general circulation models // Philosophical Transactions of the Royal Society A. 2009. Volume 367. P. 4497–4522.

307. Joe B. Delaunay Triangular Meshes in Convex Polygons // SIAM Journal on Scientific and Statistical Computing. 1986. Vol. 7, Iss. 2. P. 514–539. DOI: 10.1137/0907035

308. Joe B. Three-Dimensional Triangulations from Local Transformations // SIAM Journal on Scientific and Statistical Computing. 1989. Vol. 10. P. 718–741. DOI: 10.1137/0910044
309. Joe B. Construction of Three-Dimensional Triangulations using Local Transformations // Computer Aided Geometric Design. 1991. Vol. 8. P. 123–142.
310. Joe B. Construction of k-Dimensional Delaunay Triangulations using Local Transformations // SIAM Journal on Scientific Computing. 1993. Vol. 14, Iss. 6. P. 1415–1436. DOI: 10.1137/0914083
311. Joe B. Construction of Three-Dimensional Improved-Quality Triangulations Using Local Transformations // SIAM Journal on Scientific Computing. 1995. Vol. 16, Iss. 6. P. 1292–1307. DOI: 10.1137/0916075
312. Johnston B. P., Sullivan J. M., Kwasnik A. Automatic conversion of triangular finite element meshes to quadrilateral elements // International Journal for Numerical Methods in Engineering. 1991. Vol. 31, Iss. 1. P. 67–84. DOI: 10.1002/nme.1620310105
313. Ju T., Losasso F., Schaefer S., Warren J. Dual Contouring of Hermite Data // SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (San Antonio, TX, USA, 21–26 July 2002). ACM Press, 2002. P. 339–346. DOI: 10.1145/566570.566586
314. Kettenis D. L. COSMOS: A simulation language for continuous, discrete and combined models // Simulation. 1992. Vol. 58. P. 32–41.
315. Kettner L. Using generic programming for designing a data structure for polyhedral surfaces // Computational Geometry: Theory and Applications. 1999. Vol. 13, Iss. 1. P. 65–90.
316. Khattri S. K. Computationally Efficient Technique for Nonlinear Poisson-Boltzmann Equation // Lecture Notes in Computer Science. 2006. Vol. 3991. P. 860–883.
317. Khattri S. K. An effective quadrilateral mesh adaptation // Journal of Zhejiang University Science A. 2006. Vol. 7, Iss. 12. P. 2018–2021.
318. Khattri S. K. An adaptive quadrilateral mesh in curved domains // Serdica Journal of Computing. 2009. Vol. 3. P. 248–268.
319. Kiani S., Chan W. S., Haji-Sheikh A. Measurement of Thermal Conductivity of Triaxial Braided Composites // Journal of Composite Materials. 2008. Vol. 42, No 12. P. 1159–1177.

320. Knyazev A. V., Lashuk I. Steepest Descent and Conjugate Gradient Methods with Variable Preconditioning // *SIAM Journal on Matrix Analysis and Applications*. 2008. Vol. 29, Iss. 4. P. 1267–1280.

321. Kolšek T., Šubelj M., Duhovnik J. Generation of block-structured grids in complex computational domains using templates // *Finite Elements in Analysis and Design*. 2003. Vol. 39. P. 1139–1154.

322. Krysl P., Ortiz M. Extraction of boundary representation from surface triangulations // *International Journal for Numerical Methods in Engineering*. 2001. Vol. 50, Iss. 7. P. 1737–1758. DOI: 10.1002/nme.94

323. Kucwaj J., Borowik B. An Algorithm of Combining of Advancing Front and Delaunay Triangulation to 3-D Domain Tessellation // *Annales UMCS Informatica AI IX*. 2009. No 1. P. 15–26.

324. Kudin A. V., Choporov S. V., Gomenyuk S. I. Axisymmetric bending of circular and annular sandwich plates with nonlinear elastic core material // *Mathematical models and computer simulations*. 2017. Vol. 9, No 5. P. 601–612.

325. Kumar Pr., Saxena A. On Embedded Recursive Boundary Smoothing in Topology Optimization With Polygonal Mesh and Negative Masks // *Conference on Machines and Mechanism: Proceedings of the 1st International and 16th National Conference (IIT Roorkee, India, 18–20 December 2013)*. 2013. P. 568–576.

326. Kumar Pr., Saxena A. On topology optimization with embedded boundary resolution and smoothing // *Structural and Multidisciplinary Optimization*. 2015. Vol. 52, Iss. 6. P. 1135–1159.

327. Langelaar M. The use of convex uniform honeycomb tessellations in structural topology optimization // *Structural and Multidisciplinary Optimization: Proceedings of the 7th world congress (Seoul, South Korea, 21–25 May 2007)*. Seoul: 2007.

328. Lau T. S., Lo S. H. Finite Element Mesh Generation over Analytical Curved Surfaces // *Computers & Structures*. 1996. Vol. 59, Iss. 2. P. 301–309. DOI: 10.1016/0045-7949(95)00261-8

329. Lau T. S., Lo S. H., Lee C. K. Generation of quadrilateral mesh over analytical curved surfaces // *Finite Elements in Analysis and Design*. 1997. Vol. 27, Iss. 3. P. 251–272. DOI: 10.1016/S0168-874X(97)00015-2

330. Lawrence J., Li L. Finite element analysis of temperature distribution using ABAQUS for a laser-based tile grout sealing process // *Journal of Engineering Manufacture*. 2000. Vol. 214, No 6. P. 451–461.

331. Lawson C. L. Transforming Triangulations // *Discrete Mathematics*. 1972. Vol. 3. P. 365–372.

332. Lee Ch.-S., Lee J.-M. Anisotropic elasto-viscoplastic damage model for glass-fiber-reinforced polyurethane foam // *Journal of Composite Materials*. 2014. Vol. 48. P. 3367–3380.

333. Lee C. K., Lo S. H. A new scheme for the generation of a graded quadrilateral mesh // *Computers & Structures*. 1994. Vol. 52, Iss. 5. P. 847–857. DOI: 10.1016/0045-7949(94)90070-1

334. Lee H., Lee K. O. Multi-layered mesh generation and user subroutine development based on an Eulerian–Eulerian approach for soot filtration visualization in a three-dimensional particulate filter model // *International Journal of Engine Research*. 2014. Vol. 15, Iss. 8. P. 980–992.

335. Lehtimäki R. An Algebraic Boundary Orthogonalization Procedure for Structured Grids // *International Journal for Numerical Methods in Fluids*. 2000. Vol. 32, Iss. 5. P. 605–618.

336. Li D., Guo S., Xiang J. Modeling and nonlinear aeroelastic analysis of a wing with morphing trailing edge // *Journal of Aerospace Engineering*. 2013. Vol. 225, No 4. P. 619–631.

337. Li X.-Y., Teng S.-H. Generating well-shaped Delaunay meshed in 3D // *Discrete Algorithms: Proceedings of The Twelfth Annual ACM-SIAM Symposium (Washington, DC, USA, 7–9 January 2001)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. P. 28–37.

338. Liang X., Ebeida M. S., Zhang Y. Guaranteed-quality all-quadrilateral mesh generation with feature preservation // *Computer Methods in Applied Mechanics and Engineering*. 2010. Vol. 199. P. 2072–2083.

339. Liu G. R., Gu Y. T. An Introduction to Meshfree Methods and Their Programming. Dordrecht, The Netherlands: Springer, 2005. 496 P.

340. Liu G. R. Mesh free methods: Moving Beyond the Finite Element Method. Boca Raton, London, New York, Washington, D.C.: CRC Press, 2003. 693 P.
341. Liu Y., Snoeyink J. A Comparison of Five Implementations of 3D Delaunay Tessellation // Combinatorial and Computational Geometry. 2005. Vol. 52. P. 439–458.
342. Liu Y. C., Elmaraghy H. A., Zhang K. An expert system for forming quadrilateral finite elements // Engineering Computations. 1990. Vol. 7, Iss. 3. P. 249–257. DOI: 10.1108/eb023812
343. Lo S. H. Generating quadrilateral elements on plane and over curved surfaces // Computers and Structures. 1989. Vol. 31. P. 421–426.
344. Löhner R. Generation of Three-Dimensional Unstructured Grids by the Advancing Front Method // International Journal for Numerical Methods in Fluids. 1988. Vol. 8. P. 1135–1149.
345. Lorensen W. E., Cline H. E. Marching Cubes: A high resolution 3D surface construction algorithm // Computer Graphics. 1987. Vol. 21, Iss. 4. P. 163–169.
346. MacDonald A. E., Middlecoff J., Henderson T. A general method for modeling on irregular grids // The International Journal of High Performance Computing Applications. 2010. Vol. 25, Iss. 4. P. 392–403.
347. Madenci E., Guven I. The Finite Element Method and Applications in Engineering Using ANSYS. New York: Springer, 2015. 656 P.
348. Maple C. Geometric Design and Space Planning using the Marching Squares and Marching Cube Algorithms // Geometric Modeling and Graphics: Proceedings of the International Conference (London, UK, England, 16–18 July, 2003). IEEE, 2003. P. 90–95. DOI: 10.1109/GMAG.2003.1219671
349. March L., Steadman Ph. The Geometry of Environment: An Introduction to Spatial Organization in Design. Massachusetts: The MIT Press, 1974. 360 P.
350. Marechal L. Advances in Octree-Based All-Hexahedral Mesh Generation: Handling Sharp Features // International Meshing Roundtable: Proceedings of the 18th International Conference (Salt Lake City, Utah, USA, 25–28 October 2009). Springer-Verlag, 2009. P. 64–84.
351. Marcum D. L., Weatherill N. P. Unstructured grid generation using iterative point insertion and local reconnection // AIAA Journal. 1995. Vol. 33, No 9. P. 1619–1625.

352. Mark W., Dukes-Schlossberg J. Cosmos: A System for Supporting Engineering Negotiation // *Concurrent Engineering*. 1994. Volume 2. P. 173–182.
353. Markowsky G., Wesley M. A. Fleshing out wire frames // *IBM Journal of Research and Development*. 1980. Vol. 24, Iss. 5. P. 582–597. DOI: 10.1147/rd.245.0582
354. Mavriplis D. J. An advancing front Delaunay triangulation algorithm designed for robustness. *Journal of Computational Physics*. 1995. Vol. 117, Iss. 1. P. 90–101.
355. Meng Q., Jin Z., Fisher J. Comparison between FEBio and Abaqus for biphasic contact problems // *Journal of Engineering in Medicine*. 2013. Volume 227, No 9. P. 1009–1019.
356. Meyer M., Barr A., Lee H. Generalized barycentric coordinates on irregular polygons // *Journal of Graphics Tools*. 2002. Vol. 7, Iss. 1. P. 13–22.
357. Miller G. L., Pav S. E., Walkington N. J. Fully Incremental 3D Delaunay Refinement Mesh Generation // *International Meshing Roundtable: Proceedings of the 11th International Conference (Ithaca, New York, 15–18 September 2002)*. Ithaca: Springer-Verlag, 2002. P. 75–86.
358. Miranda A. C. O., Martha L. F. Mesh Generation on High-curvature Surfaces Based on a Background Quadtree Structure // *International Meshing Roundtable: Proceedings of the 11th International Conference (Ithaca, New York, 15–18 September 2002)*. Ithaca: Springer-Verlag, 2002. P. 333–342.
359. Miranda A. C. O., Lira W. W. M., Cavalcante-Neto J. B. A Three-Dimensional Adaptive Mesh Generation Approach using Geometric Modeling with Multi-Regions and Parametric Surfaces // *Journal of Computing and Information Science in Engineering*. 2013. Vol. 13, Iss. 2. P. 021002-1–021002-13. DOI: 10.1115/1.4024106
360. Miranda A. C. O., Lira W. W. M., Marques R. C. Finite Element Mesh Generation for Subsurface Simulation Models // *Engineering with Computers*. 2015. Vol. 31, Iss. 2. P. 305–324. DOI: 10.1007/s00366-014-0352-3
361. Miranda A. C. O., Martha L. F. Hierarchical template-based quadrilateral mesh generation // *Engineering with Computers*. 2015. Vol. 31. P. 1–15.
362. Mitchell S. A., Vavasis S. A. Quality Mesh Generation in Three Dimensions. *Computational Geometry: Proceedings of the Eighth Annual Symposium (Berlin, Germany, 10–12 June 1992)*. New York: Cornell University, 1992. P. 92–104.

363. Mitchell S. A., Vavasis S. A. Quality Mesh Generation in Higher Dimensions // SIAM Journal on Computing. 2000. Vol. 29, Iss. 4. P. 1334–1370.

364. MSC Nastran. Расчет и оптимизация конструкций. URL: <http://www.mscsoftware.ru/products/msc-nastran> (дата звернения 15.08.2016).

365. MSC Nastran – Multidisciplinary Structural Analysis. URL: <http://www.mscsoftware.com/product/msc-nastran> (дата звернения 15.08.2016).

366. Multi-view Stereo with Single-View Semantic Mesh Refinement / A. Romanoni, M. Ciccone, F. Visin, M. Matteucci // IEEE International Conference Computer Vision Workshop: Proceedings of the International Conference, Venice, Italy, 22–29 October 2017. 2017. P. 706–715. DOI: 10.1109/ICCVW.2017.89

367. Multilevel Mesh Partitioning for Optimizing Domain Shape / C. Walshaw, M. Cross, R. Diekmann, F. Schlimbach // The International Journal of High Performance Computing Applications. 1999. Vol. 13, No 4. P. 334–353.

368. Murdoch P., Benzley S. E. The Spatial Twist Continuum // International Meshing Roundtable: Proceedings of The 4th International Conference (Albuquerque, New Mexico, U.S.A., 16–17 October 1995). Sandia: Sandia National Laboratories, 1995. P. 243–251.

369. Muuss M. J., Butler L. A. Combinatorial solid geometry, boundary representations, and non-manifold geometry // State of the Art in Computer Graphics: Visualization and Modeling. 1991. Vol. 1. P. 185–223.

370. Müller J.-D., Poe P. L., Deconinck H. A Frontal Approach for Internal Node Generation in Delaunay Triangulations // International Journal for Numerical Methods in Fluids. 1993. Vol. 17. P. 241–255.

371. Müller-Hannemann M. High quality quadrilateral surface meshing without template restrictions: a new approach based on network flow techniques // The International Journal of Computational Geometry and Applications. Vol. 10, Iss. 3. P. 285–307.

372. Möller T., Trumbore B. Fast, minimum storage ray-triangle intersection // Journal of Graphics Tools. 1997. Vol. 2, Iss. 1. P. 21–28. DOI: 10.1080/10867651.1997.10487468

373. Naumann U., Schenk O. Combinatorial Scientific Computing. Boca Raton, FL: CRC Press, 2012. 600 p.

374. Neugebauer F., Diekmann R. Improved Mesh Generation: Not Simple but Good // International Meshing Roundtable: Proceedings of the 5th International Conference (Pittsburgh, Pennsylvania, U.S.A., 10–11 October 1996). Sandia: Sandia National Laboratories, 1996. P. 257–272.

375. Nielson G. M., Hamman B. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes // Visualization: Proceedings of the IEEE Conference (San Diego, California, USA, 22–25 October 1991). IEEE Computer Society Press, 1991. P. 83–91. DOI: 10.1109/VISUAL.1991.175782

376. Nigel P. W. Delaunay Triangulation in Computational Fluid Dynamics // Computers and Mathematics with Applications. 1992. Vol. 24, No 5/6. P. 129–150.

377. Nigel P. W., Hassan O. Efficient Three-Dimensional Grid Generation using the Delaunay Triangulation // Computational Fluid Dynamics: Proceedings of the First European Conference (Brussels, 7–11 Belgium September 1992). Amsterdam, New York: Elsevier, 1992. P. 961–968.

378. Notay Y. Flexible Conjugate Gradient // SIAM Journal on Scientific Computing. 2001. Vol. 22, Iss. 4. P. 1444–1460.

379. Ohtake Y., Belyaev A. Dual/Primal Mesh Optimization for Polygonized Implicit Surfaces // Solid modeling and applications: Proceedings of The seventh ACM symposium (Saarbrücken, Germany, 17–21 June 2002). ACM Press, 2002. P. 171–178. DOI: 10.1145/566282.566308

380. Ohtake Y., Belyaev A., Pasko A. Dynamic Mesh Optimization for Polygonized Implicit Surfaces with Sharp Features // The Visual Computer. 2003. Vol. 19, No 2-3. P. 115–126. DOI: 10.1007/s00371-002-0181-z116

381. Organick E. I. A Fortran IV Primer. Addison-Wesley, 1967. 263 P.

382. Owen S. J., Staten M. L., Canann S. A., Saigal S. Quad-Morphing: Advancing Front Quad Meshing Using Triangle Transformations. URL: <http://www.andrew.cmu.edu/user/sowen/qm/rt98quad.htm>.

383. Owen S. J., Staten M. L., Canann S. A. Q-Morph: An Indirect Approach to Advancing Front Quad Meshing // International Journal for Numerical Methods in Engineering. 1999. Vol. 44, Iss. 9. P. 1317–1340. DOI: 10.1002/(SICI)1097-0207(19990330)44:9<1317::AID-NME532>3.0.CO;2-N

384. Owen S. J., Saigal S. H-Morph: An Indirect Approach to Advancing Front Hex Meshing // International Journal for Numerical Methods in Engineering. 2000. Vol. 49, Iss. 1–2. P. 289–312. DOI: 10.1002/1097-0207(20000910/20)49:1/2<289::AID-NME934>3.0.CO;2-L

385. Owen S. J. Hex-domain mesh generation using 3D constrained triangulation // Computer Aided Design. 2001. Vol. 33, Iss. 3. P. 211–220.

386. Owen S. J., White D. R. Mesh-based geometry // International Journal for Numerical Methods in Engineering. 2003. Vol. 58, Iss. 2. P. 375–395. DOI: 10.1002/nme.773

387. Paoluzzi A. Geometric programming for computer aided design. Wiley, 2003. 800 p.

388. Paoluzzi A. Informatica grafica en CAD. Hoepli Informatica, 2003. 424 p.

389. Packing of concave polyhedra with continuous rotations using nonlinear optimisation / T. Romanova, J. Bennell, Y. Stoyan, A. Pankratov // European Journal of Operational Research. 2018. Vol. 268, Iss. 1. P. 37–53. DOI: 10.1016/j.ejor.2018.01.025

390. Parasaram R. K. N., Charyulu T. N. Airfoil Profile Design by Reverse Engineering Bezier Curve // International Journal of Mechanical Engineering and Robotics Research. 2012. Vol. 1, No 3. P. 410–420.

391. Parthasarathy V. N., Kodiyalam S. A constrained optimization approach to finite element mesh smoothing // Finite Elements in Analysis and Design. 1991. Vol. 9, Iss. 4. P. 309–320.

392. Pasko A., Adzhiev V., Sourin A., Savchenko V. Function representation in geometric modeling: concepts, implementation and applications // The Visual Computer. Vol. 11, No 8. P. 429–446.

393. Pasko A., Adzhiev V. Function-Based Shape Modeling: Mathematical Framework and Specialized Language // Lecture Notes in Computer Science. 2004. Vol. 2930. P. 132–160.

394. Patran – Интегрирующая среда для систем анализа, моделирования и проектирования на основе универсального графического пользовательского интерфейса. URL: <http://www.mscsoftware.ru/products/patran> (дата звернения 15.08.2016).

395. Pedoe D. Circles: A Mathematical View. The Mathematical Association of America, 1997. 138 p.

396. Pointer J. Understanding Accuracy and Discretization Error in an FEA Model. URL: <https://www.ansys.com/-/media/Ansys/corporate/resourcelibrary/conference-paper/2004-Int-ANSYS-Conf-54.PDF> (дата звернення: 19.11.2018).

397. Practical Considerations on Marching Cubes 33 Topological Correctness / L. Custodio, T. Etienne, S. Pesco, C. Silva. // Computers & Graphics. 2013. Vol. 37, Iss. 7. P. 840–850. DOI: 10.1016/j.cag.2013.04.004

398. Predicting permeability based on flow simulations and textile modelling techniques: Comparison with experimental values and verification of FlowTex solver using Ansys CFX / E. E. Swery, R. Meier, St. V. Lomov [and others] // Journal of Composite Materials. 2016. Vol. 50, No 5. P. 601–614.

399. Puri G. Python Scripts for Abaqus. Learn by Example. USA: 2011. 725 P.

400. Quadros W. R., Ramaswami K., Prinz F. B. LayTracks: a new approach to automated geometry adaptive quadrilateral mesh generation using medial axis transform // International Journal for Numerical Methods in Engineering. 2004. Vol. 61, Iss. 2. P. 209–237. DOI: 10.1002/nme.1063

401. Quadros W. R., Owen S. J. Defeaturing CAD models using a geometry-based size field and facet-based reduction operators // Engineering with Computers. 2012. Vol. 28, Iss. 3. P. 211–224. DOI: 10.1007/s00366-011-0252-8

402. Quadtree grid generation: information handling, boundary fitting and CFD applications / K. F. C. Yiu, D. M. Greaves, S. Cruz [and others] // Computers & Fluids. 1996. Vol. 25, Iss. 8. P. 759–769.

403. R-функции в компьютерном моделировании дизайна 3D-поверхности автомобиля / Д. А. Лисин, К. В. Максименко-Шейко, А. В. Толоч, Т. И. Шейко. // Прикладная информатика. 2011. № 6(36). С. 78–85.

404. R-функции и обратная задача аналитической геометрии в трехмерном пространстве / К. В. Максименко-Шейко, А. М. Мацевитый, А. В. Толоч, Т. И. Шейко // Информационные технологии. 2007. № 10. С. 23–32.

405. Rahman S., Rao B. N. An element-free Galerkin method for probabilistic mechanics and reliability // International Journal of Solids and Structures. 2001. Vol. 38. P. 9313–9330.

406. Rajan V. T. Optimality of the Delaunay triangulation in R^d // Discrete & Computational Geometry. 1994. Vol. 12, Iss. 2. P. 189–202.

407. Rappoport A. Interactive Boolean operations for conceptual design of 3-D solids // SIGGRAPH'97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques (Los Angeles, USA, 3–8 August 1997). New York, USA: ACM Press/Addison-Wesley Publishing Co, 1997. P. 269–278. DOI: 10.1145/258734.258865

408. Rees M. Combining Quadrilateral and Triangular Meshing Using the Advancing Front Approach // International Meshing Roundtable: Proceedings of the 6th International Conference (Park City, Utah, USA, 13–15 October 1997). Sandia: Sandia National Laboratories, 1997. P. 337–348.

409. Requicha A. A. G., Tilove R. B. Mathematical Foundations of Constructive Solid Geometry: General Topology of Closed Regular Sets [Technical Report TM-27]. Rochester, NY, USA: University of Rochester, 1978. 33 p.

410. Requicha A. A. G. Representations for Rigid Solids: Theory, Methods, and Systems // Computing Surveys. 1980. Vol. 12, No 4. P. 437–464. DOI: 10.1145/356827.356833

411. Requicha A. A. G., Voelcker H. B. Solid Modeling: A Historical Summary and Contemporary Assessment // IEEE Computer Graphics and Applications. 1982. Vol. 2, Iss. 2. P. 9–24. DOI: 10.1109/MCG.1982.1674149

412. Requicha A. A. G., Voelcker H. B. Solid Modeling: Current Status and Research Directions // IEEE Computer Graphics and Applications. 1983. Vol. 3, Iss. 7. P. 25–37. DOI: 10.1109/MCG.1983.263271

413. Requicha A. A. G., Rossignac J. R. Solid Modeling and Beyond // IEEE Computer Graphics and Applications. 1992. Vol. 12, Iss. 5. P. 31–44. DOI: 10.1109/38.156011

414. Ricci A. A constructive geometry for computer graphics // The computer journal. 1973. Vol. 16, No 2. P. 157–163.

415. Robust, geometrically based, automatic two-dimensional mesh generation / P. L. Baehmann, S. L. Wittchen, M. S. Shephard [and others] // International Journal for Numerical Methods in Engineering. 1987. Vol. 24, Iss. 6. P. 1043–1078. DOI: 10.1002/nme.1620240603

416. Rossignac J. R. Constraints in constructive solid geometry. *Interactive 3D graphics*: Proceedings of the 1986 workshop on, Chapel Hill, NC, USA, 23–24 October 1986. New York: ACM, 1987. P. 93–110. DOI: 10.1145/319120.319129
417. Ruppert J., Seidel R. On the difficulty of triangulating three-dimensional nonconvex polyhedra // *Discrete & Computational Geometry*. 1992. Vol. 7, Iss. 3. P. 227–253.
418. Ruppert J. A Delaunay refinement algorithm for quality 2-dimensional mesh generation // *Journal of Algorithms*. 1995. Vol. 18, Iss. 3. P. 548–585.
419. Salunke N. P., Ahamad R. A., Channiwala S. A. Airfoil Parameterization Techniques: A Review // *American Journal of Mechanical Engineering*. 2014. Vol. 2, No 4. P. 99–102. DOI: 10.12691/ajme-2-4-1
420. Samet H. The Quadtree and Related Hierarchical Data Structures // *ACM Computing Surveys*. 1984. Vol. 16, Iss. 2. P. 187–260.
421. Saxena A. A Material-Mask Overlay Strategy for Continuum Topology Optimization of Compliant Mechanisms using Honeycomb Discretization // *ASME Journal of Mechanism Design*. 2009. Vol. 130, Iss. 8. P. 082304:1–9.
422. Saxena A., Jain C. An Improved Material-Mask Overlay Strategy for Topology Optimization of Structures and Compliant Mechanisms // *ASME Journal of Mechanism Design*. 2010. Vol. 132, Iss. 6. P. 061006:1–10.
423. Saxena A. An Adaptive Material Mask Overlay Method: Modifications and Investigations on Binary, Well Connected Robust Compliant Continua // *ASME Journal of Mechanism Design*. 2009. Vol. 133, Iss. 4. P. 041004:1–11.
424. Saxena A. Topology Optimization with Negative Masks Using Gradient Search // *Structural and Multidisciplinary Optimization*. 2011. Vol. 44, Iss. 5. P. 629–649.
425. Saxena A. Combined Gradient-Stochastic Optimization with Negative Circular Masks for Large Deformation Topologies // *International Journal for Numerical Methods in Engineering*. 2013. Vol. 93, Iss. 6. P. 635–663. DOI: 10.1002/nme.4401
426. Saxena R., Saxena A. On Honeycomb Representation and SIGMOID Material Assignment in Optimal Topology Synthesis of Compliant Mechanisms // *Finite Elements in Analysis and Design*. 2007. Vol. 43, Iss. 14. P. 1082–1098.

427. Schneider Ph., Eberly D. H. Geometric Tools for Computer Graphics (The Morgan Kaufmann Series in Computer Graphics). San Francisco: Morgan Kaufmann Publishers, Elsevier Science, 2003. 1056 p.

428. Schneiders R. A Grid-based Algorithm for the Generation of Hexahedral Element Meshes // Engineering with Computers. 1996. Vol. 12, Iss. 3. P. 168–177.

429. Schneiders R., Schindler R., Weiler F. Octree-based Generation of Hexahedral Element Meshes // International Meshing Roundtable: Proceedings of the 5th International Conference (Pittsburgh, Pennsylvania, U.S.A., 10–11 October 1996). Sandia: Sandia National Laboratories, 1996. P. 205–216.

430. Schneiders R. Octree-Based Hexahedral Mesh Generation // International Journal of Computational Geometry & Applications. 2000. Vol. 10, Iss. 4. P. 383–398.

431. Shalloway A., Trott J. R. Design Patterns Explained: A New Perspective on Object Oriented Design. USA: Addison-Wesley, 2004. 480 p.

432. Shapiro V. A Convex Deficiency Tree Algorithm for Curved Polygons // Internal Journal of Computational Geometry and Applications. 2001. Vol. 11. No. 2. P. 215–238.

433. Shapiro V. Real functions for representation of rigid solids // Computer-Aided Geometric Design. 1994. Vol. 11. No. 2. P. 153–175.

434. Shapiro V. Semi-analytic geometry with R-functions // Acta Numerica. 2007. Vol. 16. P. 239–303. DOI: 10.1017/S096249290631001X

435. Shapiro V., Tsukanov I., Grishin A. Geometric Issues in Computer Aided Design/Computer Aided Engineering Integration // Journal of Computing and Information Science in Engineering. 2011. Vol. 11. Iss. 2. P. 021005.1–021005.13. DOI: 10.1115/1.3593416

436. Shekhar R., Fayyad E., Yagel R. Octree-Based Decimation of Marching Cubes Surfaces // Visualization'96: Proceedings of the IEEE International Conference (San Francisco, CA, USA, 27 Oct.–1 Nov. 1996). IEEE, 1996. P. 335–342. DOI: 10.1109/VISUAL.1996.568127

437. Shephard M. S., Georges M. K. Automatic three-dimensional mesh generation by the finite octree technique // International Journal for Numerical Methods in Engineering. 1991. Vol. 32, Iss. 4. P. 709–749. DOI: 10.1002/nme.1620320406

438. Shewchuk J. R. Robust adaptive floating-point geometric predicates. Computational Geometry: Proceedings of the 12th Annual ACM Symposium (Philadelphia, Pennsylvania, 24–26 May 1996). Philadelphia: ACM, 1996. P. 141–150.
439. Shewchuk J. R. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations // Computational Geometry: Proceedings of The 14th Annual ACM Symposium (Minneapolis, Minnesota, USA, 7–10 June 1998). Minneapolis: ACM, 1998. P. 76–85. DOI: 10.1145/276884.276893
440. Shewchuk J. R. Tetrahedral mesh generation by Delaunay refinement // Computational Geometry: Proceedings of The 14th Annual ACM Symposium (Minneapolis, Minnesota, USA, 7–10 June 1998). Minneapolis: ACM, 1998. P. 86–95.
441. Shewchuk J. R. Delaunay Refinement Algorithms for Triangular Mesh Generation. Berkeley, CA: University of California at Berkeley, 2001. 58 P.
442. Shimada K., Liao J.-H., Itoh T. Quadrilateral Meshing with Directionality Control though the Packing of Square Control // International Meshing Roundtable: Proceedings of the 7th International Conference (Dearborn, Michigan, U.S.A., 26–28 October 1998). Sandia: Sandia National Laboratories, 1998. P. 61–76.
443. Shimer M. Algorithm 112: Position of Point Relative to Polygon // Communications of the ACM. 1962. Vol. 5, Iss. 8. P. 434. DOI: 10.1145/368637.368653
444. Shu R., Zhou C., Kankanhalli M. S. Adaptive marching cubes // The Visual Computer. 1995. Vol. 11, Iss. 4. P. 202–217. DOI: 10.1007/BF01901516
445. Sieger D., Botsch M. Design, implementation, and evaluation of the surface_mesh data structure // International Meshing Roundtable: Proceedings of the 20th International Conference (Paris, France, 23–26 October 2011). Sandia: Sandia National Laboratories, 2011. P. 533–550.
446. Silva C. T., Mitchell J. S. B. Greedy Cuts: An Advancing Front Terrain Triangulation Algorithm // Advances in Geographic Information Systems: Proceedings of The 6th ACM International Symposium (Washington, 2–7 November 1998). Washington: ACM, 1998. P. 137–144. DOI: 10.1145/288692.288717
447. Sokolovskyy Ya., Levkovich M. Two-Dimensional Mathematical Models of Visco-Elastic Deformation Using a Fractional Differentiation Apparatus // International Journal of Modern Education and Computer Science. 2018. Vol. 10. Iss. 4. P. 34–43.

448. Sokolovsky Ya., Nechepurenko A., Zdolbytsky A. Software simulation of heat mass transfer using parallel computing technologies // Вісник НУЛП: Комп'ютерні системи. Теорія і практика. Львів: Видавництво НУЛП, 2017. №923. С. 34–43.

449. Sparse implicitization by interpolation: Characterizing non-exactness and an application to computing discriminants / I. Z. Emiris, T. Kalinka, C. Konaxis, T. L. Ba // Computer-Aided Design. 2013. Vol. 45, Iss. 2. P. 252–261.

450. Staten M. L., Owen S. J., Blacker T. D. Unconstrained Paving & Plastering: A New Idea for All Hexahedral Mesh Generation // International Meshing Roundtable: Proceedings of the 14th International Conference (San Diego, California, USA, 11–14 September 2005). Sandia National Laboratories: Springer-Verlag, 2005. P. 399–416. DOI: 10.1007/3-540-29090-7_24

451. Staten M. L., Owen S. J., Blacker T. D. Unconstrained plastering – Hexahedral mesh generation via advancing-front geometry decomposition // International Journal for Numerical Methods in Engineering. 2010. Vol. 81, Iss. 2. P. 135–171. DOI: 10.1002/nme.2679

452. Steensland J., Ray J. A partitioner-centric model for structured adaptive mesh refinement partitioning trade-off optimization: Part I // The International Journal of High Performance Computing Applications. 2005. Vol. 19, No 4. P. 409–422.

453. Steensland J., Ray J. A partitioner-centric model for structured adaptive mesh refinement partitioning trade-off optimization: Part II // Parallel Processing Workshops: Proceedings of the International Conference (Montreal, QC, Canada, 15–18 August 2004). 2004. P. 231–238.

454. Stroud I. Boundary Representation Modelling Techniques. London: Springer, 2006. 808 P. ISSN: 1846283124.

455. Stuhne G. R., Peltier W. R. New Icosahedral Grid-Point Discretizations of the Shallow Water Equations on the Sphere // Journal of Computational Physics. 1999. Vol. 148. P. 23–58.

456. Su P., Drysdale R. L. S. A comparison of sequential Delaunay triangulation algorithms // Computational Geometry. 1997. Vol. 7, Iss. 5-6. P. 361–385.

457. Taghavi R. Automatic, parallel and fault tolerant mesh generation from CAD // Engineering with Computers. 1996. Vol. 12, Iss. 3. P. 178–185.

458. Taghavi R. Automatic block decomposition using fuzzy logic analysis // International Meshing Roundtable: Proceedings of the 9th International Conference, New Orleans, Louisiana, USA, 2–5 October 2000. Sandia: Sandia National Laboratories, 2000. P. 187–192.

459. Takahashi H., Shimizu H. A general purpose automatic mesh generation using shape recognition technique // ASME Computers in Engineering. 1991. Vol. 1. P. 519–526.

460. Takahashi H., Shimizu H., Moriyama H. A Three-Dimensional Automatic Mesh Generation System Using Shape Recognition Technique // Transactions of the Japan Society of Mechanical Engineers. 1993. Vol. 59, No 560, Series A. P. 1161–1167.

461. Talbert J. A., Parkinson A. R. Development of an automatic two-dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definition // International Journal for Numerical Methods in Engineering. 1990. Vol. 29, Iss. 7. P. 1551–1567. DOI: 10.1002/nme.1620290712

462. Talischi C., Paulino Gl. H., Le Ch. H. Honeycomb wachspress finite elements for structural topology optimization // Structural and Multidisciplinary Optimization. 2009. Vol. 37, Iss. 6. P. 569–583.

463. Talischi C., Paulino Gl. H., Pereira A. PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab // Structural and Multidisciplinary Optimization. 2012. Vol. 45, Iss. 3. P. 309–328.

464. Talischi C., Pereira A., Paulino G. H. Polygonal finite elements for incompressible fluid flow // International Journal for Numerical Methods in Fluids. 2014. Vol. 74. P. 134–151.

465. Tam T. K. H., Armstrong C. G. 2D finite element mesh generation by medial axis subdivision // Advances in Engineering Software and Workstations. 1991. Vol. 13, No 5-6. P. 313–324.

466. Tautges T. J., Mitchell S. A. Progress Report on the Whisker Weaving All-haxahedral Meshing Algorithm // Numerical Grid Generation in Computational Field Simulations: Proceedings of the 5th International Conference (Mississippi State University, 1–5 April 1996). Mississippi: Mississippi State University, 1996. P. 659–670.

467. Tautges T. J., Blacker T. D., Mitchell S. A. The Whisker Weaving Algorithm: a connectivity based method for constructing all-hexahedral finite element meshes // International Journal for Numerical Methods in Engineering. 1996. Vol. 39, Iss. 19. P. 3327–3350. DOI: 10.1002/(SICI)1097-0207(19961015)39:19<3327::AID-NME2>3.0.CO;2-H

468. The Non-hydrostatic Icosahedral Atmospheric Model: description and development / M. Satoh, H. Tomita, H. Yashiro [and others] // Progress in Earth and Planetary Science. 2014. Vol. 18, Iss. 1. P. 1–32.

469. The Spatial Twist Continuum Captures the Global Connectivity of an All-Hexahedral Finite Element Mesh / P. Murdoch, S. E. Benzley, T. D. Blacker, S. A. Mitchell. URL: <http://www.sandia.gov/~samitch/papers/MurdochBenzleyBlackerMitchellfixed.pdf> (дата звернення 15.08.2016).

470. The spatial twist continuum: A connectivity based method for representing all-hexahedral finite element meshes / P. Murdoch, S. E. Benzley, T. D. Blacker, S. A. Mitchell // Finite Elements in Analysis and Design. 1997. Vol. 28, Iss. 2. P. 137–149.

471. Tinoco-Ruiz J.-G., Barrera-Sánchez P., Cortes-Médina A. Some Properties of Area Functionals in Numerical Grid Generation // International Meshing Roundtable: Proceedings of the 10th International Conference (Newport Beach, U.S.A., 7–10 October 2001). Sandia: Sandia National Laboratories, 2001. P. 43–54.

472. Thompson J. F., Sony Bh., Weatherill N. Handbook of grid generation. New York: CRC Press, 1999. 1136 p.

473. Tournois J., Alliez P., Devillers O. Interleaving Delaunay Refinement and Optimization for 2D Triangle Mesh Generation // International Meshing Roundtable: Proceedings of the 16th International Conference (Washington, 14–17 October 2007). Washington: Springer, 2008. P. 83–101.

474. Tournois J., Wormser C., Alliez P., Desbrun M. Interleaving Delaunay refinement and optimization for practical isotropic tetrahedron mesh generation // ACM Transactions on Graphics. 2009. Vol. 28, Iss. 3. Article No. 75.

475. Two Approaches to Modeling and Solving the Packing Problem for Convex Polytopes / Y. E. Stoian, A. M. Chugay, A. V. Pankratov, T. E. Romanova // Cybernetics and Systems Analysis. 2018. Vol. 54, Iss. 4. P. 585–593. DOI: 10.1007/s10559-018-0059-3

476. Turner T. L., Patel H. D. Analysis of SMA Hybrid Composite Structures in MSC.Nastran and ABAQUS // Journal of Intelligent Material Systems and Structures. 2007. Vol. 18, Iss. 5. P. 435–447.
477. Validation of a Laminated Beam Model of LIPCA Piezoelectric Actuators / N. S. Goo, A. Haris, H. C. Park, K. J. Yoon // Journal of Intelligent Material Systems and Structures. 2005. Vol. 16, No 3. P. 189–195.
478. Vaněček G. Brep-index: A Multidimensional Space Partitioning Tree // International Journal of Computational Geometry & Applications. 1991. Vol. 1, Iss. 3. P. 243–261. DOI: 10.1142/S0218195991000189
479. Voelcker H. B., Requicha A. A. G. Geometric Modeling of Mechanical Parts and Processes // Computer. 1977. Vol. 10, Iss. 12. P. 48–57. DOI: 10.1109/C-M.1977.217601
480. Wang D. A simple method for implicitizing rational curves and surfaces // Journal of Symbolic Computation. 2004. Vol. 38, Iss. 1. P. 899–914.
481. Wang N., Lee J.-L. Geometric Properties of the Icosahedral-Hexagonal Grid on the Two-Sphere // SIAM Journal on Scientific Computing. 2011. Vol. 33, No 5. P. 2536–2559.
482. Warren J. Barycentric coordinates for convex polytopes // Advances in Computational Mathematics. 1996. Vol. 6, Iss. 1. P. 97–108.
483. Watson D. F. Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes // The Computer Journal. 1981. Vol. 24, Iss. 2. P. 167–172.
484. Watson D. F. Contouring: A guide to the analysis and display of spatial data. Oxford: Pergamon Press, 1992. 321 p.
485. Web-based shape modeling with HyperFun / R. Cartwright, V. Adzhiev, A. A. Pasko, Y. Goto, T. L. Kunii // IEEE Computer Graphics and Applications. 2005. Vol. 25, Iss. 2. P. 60–69.
486. Wellford L. C., Gorman M. R. A finite element transitional mesh generation procedure using sweeping functions // International Journal for Numerical Methods in Engineering. 1988. Vol. 26, Iss. 12. P. 2623–2643. DOI: 10.1002/nme.1620261204
487. Wilhelms J., van Gelder A. Octrees for faster isosurface generation // ACM Transactions on Graphics. 1992. Vol. 11, Iss. 3. P. 201–22. DOI: 10.1145/130881.130882
488. Yalcin H., Unel M., Wolovich W. Implicitization of Parametric Curves by Matrix Annihilation // International Journal of Computer Vision. 2003. Vol. 54, Iss. 1. P. 105–115.

489. Yates D. N., Sable W. W., Vinson T. J. The DAISY code // Numerical and Computer Methods in Structural Mechanics. New-York, 1973. P. 175–210.
490. Yerry M. A., Shephard M. S. A modified quadtree approach to finite element mesh generation // IEEE Computer Graphics and Applications. 1983. Vol. 3, Iss. 1. P. 39–46.
491. Yerry M. A., Shephard M. S. Automatic three-dimensional mesh generation by the modified-octree technique // International Journal for Numerical Methods in Engineering. 1984. Vol. 20, Iss. 11. P. 1965–1990. DOI: 10.1002/nme.1620201103
492. Zhang H., Zhao G. Adaptive hexahedral mesh generation based on local domain curvature and thickness using a modified grid-based method // Finite Elements in Analysis and Design. 2007. Vol. 43, Iss. 9. P. 691–704.
493. Zhang Y., Jia Y., Wang S. S. Y. 2D Nearly Orthogonal Mesh Generation // International Journal for Numerical Methods in Fluids. 2004. Vol. 46, Iss. 7. P. 685–707.
494. Zhang Y., Jia Y., Wang S. S. Y. 2D Nearly Orthogonal Mesh Generation with Controls of Distortion Function // Journal of Computational Physics. 2006. Vol. 218, Iss. 2. P. 549–571.
495. Zhang Y., Jia Y., Wang S. S. Y. Structured Mesh Generation with Smoothness Controls // International Journal for Numerical Methods in Fluids. 2006. Vol. 51, Iss. 11. P. 1255–1276.
496. Zhao Y., Yu B. A modified paving algorithm for quadrilateral mesh generation // Kung Cheng Je Wu Li Hsueh Pao/Journal of Engineering Thermophysics. 2013. Vol. 34, Iss. 4. P. 728–732.
497. Zheng J. Z. X., Groth C. P. T. Block-Based Adaptive Mesh Refinement Finite-Volume Scheme for Hybrid Multi-Block Meshes // Computational Fluid Dynamics (ICCFD7): Proceedings of the 7th International Conference (Big Island, Hawaii, 9–13 July 2012). Hawaii: 2012. P. 1–19.
498. Zheng Y., Lewis R. W., Gethin D. T. Three-dimensional unstructured mesh generation: Part 1. Fundamental aspects of triangulation and point creation // Computer Methods in Applied Mechanics and Engineering. 1996. Vol. 134. P. 249–268.
499. Zheng Y., Lewis R. W., Gethin D. T. Three-dimensional unstructured mesh generation: Part 2. Surface Meshes // Computer Methods in Applied Mechanics and Engineering. 1996. Vol. 134. P. 269–284.

500. Zhou Q. A Simple Grid Generation Method // International Journal for Numerical Methods in Fluids. 1998. Vol. 26, Iss. 6. P. 713–724.

501. Zhou T., Shimada K. An Angle-based Approach to Two-Dimensional Mesh Smoothing // International Meshing Roundtable: Proceedings of the 9th International Conference (New Orleans, U.S.A., 2–5 October 2000). 2000. Sandia: Sandia National Laboratories. P. 373–384

502. Zienkiewicz O. C., Phillips D. V. An automatic mesh generation scheme for plane and curved surfaces by ‘isoparametric’ co-ordinates // International Journal For Numerical Methods in Engineering. 1971. Vol. 3, Iss. 4. P. 519–528. DOI: 10.1002/nme.1620030407

503. Zienkiewicz O. C., Taylor R. L., Zhu J. Z. The Finite Element Method: Its Basis and Fundamentals, Seventh Edition. London: Butterworth-Heinemann, 2013. 756 P.

504. Zienkiewicz O. C., Taylor R. L., Fox D. D. The Finite Element Method for Solid and Structural Mechanics, Seventh Edition. London: Butterworth-Heinemann, 2013. 672 P.

505. Zienkiewicz O. C., Taylor R. L., Nithiarasu P. The Finite Element Method for Fluid Dynamics, Seventh Edition. London: Butterworth-Heinemann, 2013. 584 P.

ДОДАТОК А

Довідка про використання результатів дисертаційної роботи державним підприємством «Конструкторське бюро «Південне»



Державне підприємство
«Конструкторське бюро «Південне»
 вул. Криворізька, 3, м. Дніпро, Україна, 49008
 тел.: (056) 372-00-22, факс: (056) 792-50-41, (0562) 34-03-83
 E-mail: info@yuzhnoye.com, www.yuzhnoye.com

Вих. № _____ від _____

«ЗАТВЕРДЖУЮ»

Головний конструктор і начальник КБ-1
 ДП «Конструкторське бюро «Південне»

М.О. Дегтярев

_____ 2018 р.

ДОВІДКА

про використання результатів дисертаційної роботи Чопорова Сергія Вікторовича, «Математичне моделювання та аналіз форм об'єктів у САПР машинобудування», виконаної у Запорізькому національному університеті.

Комісія у складі: В.М. Сіренко, Д.В. Клименко, І.Ф. Ларіонова засвідчує, що результати дисертаційної роботи: метод подання математичних моделей оболонкових конструкцій, методи побудови дискретних моделей оболонкових конструкцій, проблемно-орієнтована мова моделювання форм об'єктів, САПР «qzCAD», використані у науково-виробничій діяльності Державного підприємства «Конструкторське бюро «Південне» ім. М. К. Янгеля» для визначення напружено-деформованого стану оболонкових конструкцій ракетної техніки, зокрема в дослідженні паливних відсіків під дією внутрішнього тиску та композитних головних обтічників, з урахуванням особливостей геометричної форми, характеристик жорсткості і характеру зовнішнього навантаження.

Методи побудови дискретних моделей оболонкових конструкцій, проблемно-орієнтована мова моделювання форм об'єктів, САПР «qzCAD» застосовані при виконанні робіт за договорами:

1. Розробка методики визначення руйнівних навантажень оболонок під дією внутрішнього тиску з урахуванням пластичних деформацій матеріалу і чисельного прогнозування зон руйнування оболонок під дією внутрішнього надлишкового тиску з урахуванням нагріву;

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л. В. Колесник

2. Розробка методики визначення напружено - деформованого стану головного обтічника тришарової стільникової конструкції з раціональним вибором характеристик анізотропії матеріалу в залежності від характеру зовнішнього навантаження конструкції та розробки розрахункової моделі для визначення критичних значень навантажень.

Дана довідка не є підставою для пред'явлення фінансових претензій до ДП КБ «Південне» ім. М.К. Янгеля.

Начальник розрахунково-теоретичного комплексу,
кандидат технічних наук



В.М. Сіренко

Головний науковий співробітник,
кандидат технічних наук



І.Ф. Ларіонов

Начальник відділу
міцності і навантажень,
кандидат технічних наук



Д.В. Клименко

Вчений секретар
ДП «КБ «Південне»,
кандидат технічних наук



Л.П. Потапович

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л. В. Колесник

ДОДАТОК Б

Довідка про використання результатів дисертаційної роботи у навчальному процесі Запорізького національного університету



ДОВІДКА

про використання результатів дисертаційної роботи **Чопорова Сергія Вікторовича**, «Математичне моделювання та аналіз форм об'єктів у САПР машинобудування», виконаної у Запорізькому національному університеті.

Комісія у складі:

Голова комісії: завідувач кафедри програмної інженерії, кандидат фізико-математичних наук, доцент, Лісняк Андрій Олександрович

Члени комісії: доцент кафедри програмної інженерії, к.ф.-м.н., доцент Горбенко В. І.; доцент кафедри програмної інженерії, к.т.н., доцент Мухін В. В.

Засвідчує, що результати дисертаційної роботи: методи подання математичних моделей форм об'єктів, методи побудови дискретних моделей форм об'єктів, методи генерації та обробки дискретних моделей форм об'єктів на базі технологій паралельних обчислень, шаблони проектування об'єктно-орієнтованої САПР, використані у навчальній діяльності кафедри програмної інженерії Запорізького національного університету при викладанні дисциплін «Паралельні та розподілені обчислення», «Програмне забезпечення наукових досліджень» для магістрів спеціальності 121 – інженерія програмного забезпечення, а також у підготовці бакалаврських і магістерських кваліфікаційних робіт.

Голова комісії: Лісняк Андрій Олександрович

Члени комісії: Горбенко Віталій Іванович

Мухін Віталій Вікторович

Підпис
Підпис
Підпис

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л. В. Колесник