

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ
(м. Сєвєродонецьк)

Факультет Інформаційних технологій та електроніки
(повне найменування факультету)

Кафедра Програмування та математики
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
до дипломної роботи

освітньо-кваліфікаційного рівня магістр
(бакалавр, спеціаліст, магістр)

напряму підготовки 122
(шифр і назва напряму підготовки)

спеціальності 122 Комп'ютерні науки та інформаційні технології (спеціалізація "Інформатика")
(шифр і назва спеціальності)

на тему Моделювання складних географічних ландшафтів

Виконав: студент групи ІНФ-16дм

Салова А.І.
(прізвище, та ініціали) (підпис)

Керівник Фесенко Т.М.
(прізвище та ініціали) (підпис)

Завідувач кафедри Лифар В. О.
(прізвище та ініціали) (підпис)

Рецензент Лорія М.Г.
(прізвище та ініціали) (підпис)

3. Вихідні дані до проекту (роботи) Матеріали науково-дослідної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Актуальність теми і постановка завдань дослідження. Систематизація відомостей про побудову різних типів поверхонь;

2) Формулювання основних принципів представлення даних для зберігання інформації про ландшафти ;

3) Огляд та аналіз алгоритмів генерації ландшафтів;

4) Створення комп'ютерних моделей розглянутих алгоритмів генерації складних ландшафтних форм;

5. Перелік графічного матеріалу .

6. Дата видачі завдання 08.11.2017

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Актуальність теми і постановка завдань дослідження.	08.11.17-15.11.17	
2	Систематизація відомостей про побудову різних типів поверхонь	16.11.17-17.12.17	
3	Формулювання основних принципів представлення даних для зберігання інформації про ландшафти	18.12.17-25.12.17	
4	Огляд та аналіз алгоритмів генерації ландшафтів	26.12.17-01.01.18	
5	Створення комп'ютерних моделей розглянутих алгоритмів генерації складних ландшафтних форм	02.01.18-10.01.18	
6	Оформлення пояснювальної записки	11.01.18-15.01.18	

Студент Салова А.І.

(підпис)

(прізвище та ініціали)

Керівник проекту (роботи) Фесенко Т.М

(підпис)

(прізвище та ініціали)

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1	
ЕТАПИ РОЗВИТКУ МАШИННОЇ ГРАФІКИ	5
1.1. Загальні відомості.....	5
1.2. Інтерактивна комп'ютерна графіка.....	7
1.3. Комп'ютерна графіка й глобальна мережа Internet.....	11
РОЗДІЛ 2	
МОДЕЛЮВАННЯ ПОВЕРХОНЬ	13
2.1. Математична модель поверхонь.....	13
2.2. Аналітичні поверхні	15
2.3. Поверхні руху	16
2.4. Лінійчаті поверхні	23
2.5. Поверхні Кунса.....	25
2.6. Сплайнові поверхні	30
2.7. Поверхні Безьє	37
2.8. Раціональні поверхні.....	39
2.9. NURBS поверхні.....	42
2.10. Поверхні, що базуються на поверхнях.....	48
2.11. Обмежені контурами поверхні	53
2.12. Способи побудови поверхонь.....	56
РОЗДІЛ 3	
ГЕНЕРАЦІЯ ЛАНДШАФТІВ.....	59
3.1. Регулярна сітка висот.....	59
3.2. Іррегулярна сітка	60
3.3. Посегментна карта висот	61
3.4. Алгоритми генерації ландшафтів	62

РОЗДІЛ 4

КОМП'ЮТЕРНА МОДЕЛЬ СКЛАДНИХ ЛАНДШАФТНИХ ФОРМ	68
4.1. Комп'ютерне моделювання рельєфних поверхонь, що описуються функціональними залежностями.....	68
4.2. Генерація ландшафту за допомогою стандартного алгоритму.....	70
4.3. Генерація ландшафту за допомогою пагорбового алгоритму	73
4.4. Генерація ландшафту за допомогою фрактального алгоритму	76
ВИСНОВОК	78
ЛІТЕРАТУРА	79
ДОДАТОК	81

ВСТУП

З виникненням досить потужної комп'ютерної техніки з'явилася можливість використати її для досліджень у різних областях наукового й практичного знання. Сьогодні питання автоматизації різних галузей людської життєдіяльності дуже актуальне. Практично скрізь можна спостерігати застосування комп'ютерів: від звичайних персональних до потужних кластерів, здатних зберігати й обробляти терабайти інформації. Не є виключенням і проблема дослідження й моделювання ландшафтів. У роботі сучасних дизайнерів, архітекторів, розроблювачів комп'ютерних ігор та дистанційних керуючих систем й багатьох інших фахівців це одне з основних завдань. Передбачати зміну існуючого ландшафту або митецьки вписати в нього деякі декоративні деталі, урахувати рельєф місцевості при дистанційному керуванні польотом безпілотною апарату для прокладки оптимального курсу – ось короткий перелік питань, які можуть бути вирішені за допомогою дослідження складних географічних ландшафтів.

На сьогодні існує безліч пакетів, що дозволяють моделювати ландшафт у тім або іншому вигляді, але в них є ряд властивостей, які роблять їх не кращими варіантами для використання. Наприклад, це:

- висока вартість ліцензійної продукції;
- необхідність для користувача мати достатні навички для роботи із програмним продуктом;
- програмний продукт не призначений конкретно для моделювання ландшафтів;
- недостатня наочність результатів для непрофесіоналів та ін.

Таким чином, метою магістерської атестаційної роботи є аналіз ефективності та систематизація існуючих методів моделювання ландшафтів, їхня можлива модифікація.

Для досягнення поставленої мети, в магістерській атестаційній роботі сформульовані та вирішені наступні **задачі**:

- 1) систематизація відомостей про побудову різних типів поверхонь;

- 2) формулювання основних принципів представлення даних для зберігання інформації про ландшафти;
- 3) аналіз алгоритмів генерації ландшафтів;
- 4) створення комп'ютерних моделей розглянутих алгоритмів генерації складних ландшафтних форм.

Об'єктом дослідження магістерської роботи є моделювання та генерація складних ландшафтних форм.

Предметом дослідження є алгоритми генерації складних ландшафтних форм.

РОЗДІЛ 1

ЕТАПИ РОЗВИТКУ МАШИННОЇ ГРАФІКИ

1.1. Загальні відомості

Швидкий розвиток обчислювальних засобів, розширення їхніх можливостей є головним чинником усе більш широкого впровадження їх у різні сфери діяльності людини. Винятково інтенсивно розвивається напрямок комп'ютерного синтезу зображень. Можна вважати, що в цей час в основному сформувалася нова галузь інформатики – машинна графіка. Її можна визначити як науку про математичне моделювання геометричних форм і вигляду об'єктів, а також методів їхньої візуалізації.

Інтерес до синтезу зображень пояснюється високою інформативністю останніх. Інформація, що втримується в зображенні, представлена в найбільш концентрованій формі, і ця інформація, як правило, більш доступна для аналізу: для її сприйняття одержувачеві досить мати невеликий обсяг спеціальних знань.

Прагнення візуалізувати інформацію спостерігається практично у всіх сферах діяльності людини. І з початку використання ЕОМ виникла проблема подання одержаних даних у вигляді зображення. На початковому етапі програмними засобами формувалися різні символні зображення: діаграми, графіки, умовні схеми, плани й таке інше. Вивід такої інформації здійснювався безпосередньо через алфавітно-цифровий друкувальний пристрій.

Формування машинної графіки як самостійного напрямку відноситься до початку 60-х років, коли Сазерлендом був створений перший спеціалізований пакет програмного забезпечення машинної графіки. В 60-і роки були сформульовані принципи малювання відрізками, видалення невидимих ліній, методи відображення складних поверхонь, визначені методи формування тіней, обліку освітленості сюжету. Перші роботи були в основному спрямовані на розвиток векторної графіки, тобто малювання відрізками. З перших робіт цього напрямку найбільший вплив на розвиток векторної графіки зробили алгоритми Брезенхема.

В 70-і роки значне число теоретичних і прикладних робіт було спрямовано на розвиток методів відображення просторових форм і об'єктів. Цей напрямок прийнято називати тривимірною машинною графікою. Математичне моделювання тривимірних сюжетів вимагає обліку тривимірності простору предметів, розташування в ньому джерел висвітлення й спостерігача. Потребами тривимірної машинної графіки обумовлена велика кількість робіт з методів апроксимації й подання складних поверхонь, відображенню візерунків з них, генеруванню текстур, рельєфу, моделюванню умов висвітлення. Одночасно значна увага приділялася поліпшенню якості синтезованих зображень, підвищенню рівня їхньої реалістичності; згладжуванню погрішностей, що виникають у результаті апроксимації геометричної форми реальних тіл і просторової дискретизації зображення. Методи тривимірної машинної графіки дозволяють візуалізувати складні функціональні залежності, одержати зображення проєктованих, ще не створених об'єктів, оцінити вигляд предмета з недоступної для спостереження позиції й вирішити ряд інших аналогічних завдань.

В 80-і роки сфера застосування машинної графіки істотно розширилася. З'явилася база для широкого використання методів машинної графіки – персональні комп'ютери. У результаті машинна графіка стала інструментом не тільки інженерів-дослідників, але й фахівців багатьох інших галузей, безпосередньо не зв'язаних ні з технікою, ні із програмуванням. Збільшення пам'яті й швидкості обробки інформації в персональних ЕОМ, створення відеокомплексів із широким набором програм машинної графіки, можливість керування ними в діалоговому режимі сприяють подальшому розширенню застосування машинної графіки.

Одним з найбільш нових напрямків машинної графіки є розробка принципів і методів формування реалістичних зображень, тобто зображень, які могли б спостерігатися візуально або реєструвалися б оптичними, фотографічними або оптикоелектронними пристроями. Потреба в створенні реалістичних зображень виникає в таких областях, як дизайн, машинобудівне й

архітектурне проектування, реклама й т.п. У ряді застосувань реалістичність використовується як засіб підвищити емоційний вплив зображення, як, наприклад, у рекламі, в інших же галузях – як засіб оцінки якості обраних рішень (в архітектурному плануванні, дизайні й ін.), у третіх – як засіб створення реалістичної обстановки (відеотренажери, системи розпізнавання образів). У системах формування реалістичних зображень повинна забезпечуватися передача всієї сукупності образотворчих властивостей: об'ємність, розташування предметів у сюжеті, півтони, кольори, текстура поверхні. Теоретичні дослідження останніх років значною мірою були спрямовані на розробку методів і засобів відображення всієї сукупності характеристик сюжетів у синтезованому зображенні.

Швидке розширення функціональних можливостей сучасної обчислювальної техніки створило базу для розвитку систем машинної графіки, що забезпечують відображення динамічних сюжетів, у яких зображення послідовно змінюють одне інше. З таких систем можна відзначити три групи: системи графічного моделювання для наочного подання процесів у хімії, медицині, астрономії та ін.; системи імітації динамічних ситуацій (наприклад, тренажери); системи одержання двомірних і тривимірних зображень для телебачення й кіно. Саме в розвитку цих систем найбільше рельєфно проявляються труднощі й проблеми тривимірної машинної графіки. Для них потрібно не тільки висока точність моделей, але й надзвичайно висока продуктивність обчислювальних засобів. Створення й впровадження в практику трансп'ютерних обчислювальних систем створюють у цей час реальні перспективи подальшого розвитку можливостей машинної графіки.

1.2. Інтерактивна комп'ютерна графіка

Донедавна досить популярним було словосполучення інтерактивна комп'ютерна графіка. Їм підкреслювалася здатність комп'ютерної системи створювати зображення й вести діалог з людиною. Раніше системи працювали в пакетному режимі – способи діалогу були не розвинені. Сьогодні майже будь-яку програму можна вважати системою інтерактивної комп'ютерної графіки.

Історично першими інтерактивними системами вважаються системи автоматизованого проектування (САПР), які з'явилися в 60-х роках [1, 2, 3]. Вони являють собою значний етап в еволюції комп'ютерів і програмного забезпечення. У системі інтерактивної комп'ютерної графіки користувач сприймає на дисплеї зображення, що представляє деякий складний об'єкт, і може вносити зміни в опис (модель) об'єкта. Такими змінами можуть бути як введення й редагування окремих елементів, так і завдання числових значень для будь-яких параметрів, а також інші операції по введенню інформації на основі сприйняття зображень.

Системи типу САПР активно використовуються в багатьох областях, наприклад у машинобудуванні й електроніці. Одними з перших були створені САПР для проектування літаків, автомобілів, системи для розробки мікроелектронних інтегральних схем, архітектурні системи. Такі системи спочатку функціонували на досить великих комп'ютерах. Потім поширилося використання швидкодіючих комп'ютерів середнього класу з розвиненими графічними можливостями – графічних робочих станцій. З ростом потужностей персональних комп'ютерів все частіше САПР використовували на дешевих масових комп'ютерах, які зараз мають достатню швидкодію й обсяги пам'яті для рішення багатьох завдань. Це призвело до широкого поширення САПР.

Як тільки графічні системи стали мати достатню продуктивність для створення складних динамічних зображень, виникла ідея застосувати їх як засіб моделювання реальної обстановки (симулятора) на різного роду тренажерах. Першими такі системи освоїли авіатори й використали для навчання пілотів на землі. Це дозволило значно знизити вартість навчання, гарантуючи при цьому його високу якість і безпеку. Використання в сучасних системах спеціальних БІС (бойових інформаційних систем) дозволило настільки знизити вартість подібного роду пристроїв, що вони дійшли до рівня дитячих іграшок. Наприклад, за допомогою засобів комп'ютерної графіки можливе створення графічної моделі робота, що використовується при плануванні технологічних

процесів і підготовці програм керування роботами, що працюють в оточення безлічі інших елементів виробничої системи.

Стають усе більше популярними геоінформаційні системи (ГІС). Це відносно нова для масових користувачів різновидність систем інтерактивної комп'ютерної графіки. Вона акумулює у собі методи й алгоритми багатьох наук і інформаційних технологій. Такі системи використовують останні досягнення технологій баз даних, у них закладені багато методів і алгоритми математики, фізики, геодезії, топології, картографії, навігації й, звичайно ж, комп'ютерної графіки. Системи типу ГІС найчастіше вимагають значних потужностей комп'ютера як у плані роботи з базами даних, так і для візуалізації об'єктів, які перебувають на поверхні Землі. Причому, візуалізацію необхідно робити з різним ступенем деталізації – як для Землі в цілому, так і в границях окремих ділянок (рис. 1.1). Помітне прагнення розроблювачів ГІС підвищити реалістичність зображень просторових об'єктів і територій.

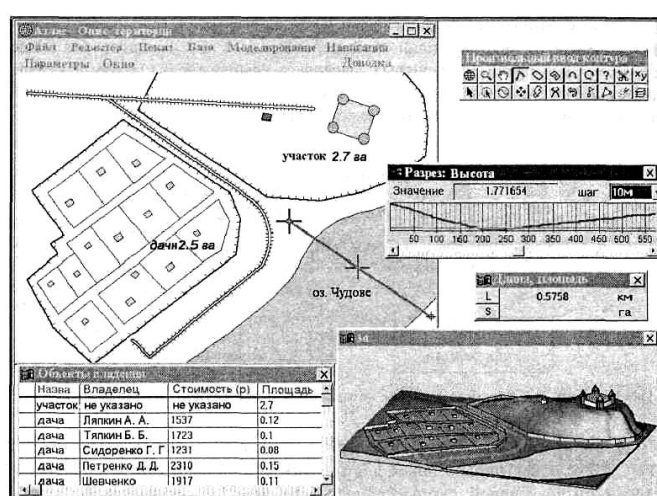


Рис. 1.1. Работа с геоинформационной системой (ГИС «ОКО»)

Типовими для будь-якій ГІС є такі операції – уведення й редагування об'єктів з урахуванням їх розташування на поверхні Землі, формування різноманітних цифрових моделей, запис у бази даних, виконання різноманітних запитів до баз даних. Важливою операцією є аналіз із обліком просторових, топологічних відносин безлічі об'єктів, розташованих на деякій території та ін.

Важливим етапом розвитку систем комп'ютерної графіки є так звані системи віртуальної реальності (virtual reality). Нарощування потужності

комп'ютерів, підвищення реалістичності тривимірної графіки й удосконалювання способів діалогу з людиною дозволяють створювати ілюзію входження людини у віртуальний простір. Цей простір може бути моделлю або існуючого простору, або вигаданого. В VR-системах людина-спостерігач користується спеціальним шоломом з парою мініатюрних дисплеїв, на екранах яких формуються різні зображення для правого й лівого ока. У результаті створюється стереоефект. Крім того, положення й орієнтація голови спостерігача постійно аналізуються й відповідно змінюється зображення на екранах дисплеїв. Якщо додати до цього ще й засоби «впливу» на це середовище такі як рукавички із силомоментними датчиками, а також звуковий супровід, то створюється повна ілюзія занурення у віртуальне середовище. Спостерігач відчуває себе учасником подій, що відбуваються. Це вже не проста розвага, а інструмент для практичного застосування. Наприклад, за допомогою такої системи хірург може відробити методику проведення операції, астронавт може підготуватися до виходу у відкритий космос і проведення ремонтних робіт.

Зразки комп'ютерної графіки відомі кожному. Набули поширення, наприклад, різноманітні комп'ютерні ігри. Значну роль у них грає анімація, реалістичність зображень, досконалість способів вводу-виводу інформації. Тут слід зазначити, що в багатьох комп'ютерних іграх реалізовані ідеї й методи, які раніше були втілені в професійних дорогих системах, наприклад, у тренажерах для льотчиків.

У телебаченні, кінематографії й рекламній справі останнім часом також широко використовуються засоби комп'ютерної графіки, що дозволяють створювати динамічні зображення, практично не відмінні від створених класичними засобами. Саме їх ми часто бачимо на екранах телевізорів і в кінотеатрах. Подібного роду зображення (правда, статичні) заповнили й сторінки популярних періодичних видань. Вартість створення повнометражного кінофільму за допомогою комп'ютера порівнянна з вартістю

звичайної зйомки такого ж фільму, але при цьому можна створювати такі спецефекти й трюки, які недоступні для «живої» зйомки.

Одним з перших відомих фільмів, знятих подібним чином, були «Зоряні війни». Він був створений за допомогою суперкомп'ютера Cray. Етапи подальшого розвитку комп'ютерного кінематографа можна простежити по таких фільмах, як «Термінатор 2», «Вавилон 5», «Лекс» і десяткам (якщо вже не сотням) інших. Донедавна технології комп'ютерної графіки використовувалися для створення спецефектів, зображень екзотичних чудовиськ, імітації стихійних лих і інших елементів, які були лише фоном для гри живих акторів. В 2001 році вийшов на екрани повнометражний кінофільм «Фінальна фантазія», у якому все, включаючи зображення людей, синтезовано комп'ютером – живі актори тільки озвучили ролі за кадром.

1.3. Комп'ютерна графіка й глобальна мережа Internet

Важливою подією в житті суспільства стала поява глобальної мережі Internet. Зараз відбувається її бурхливий розвиток. Накопичуються потужності каналів передачі даних, удосконалюються способи обміну й обробки інформації. Мережу Internet використовує усе більш людей у всіх країнах. Це спосіб спілкування людей, обміну інформацією, зближення мов, поширення ідей, новий простір для бізнесу тощо. Можна ставитися до Internet по-різному – наприклад, одні вважають її важливим фактором демократизації, а інші називають знаряддям чийогось світового панування. Імовірно, що обидві ці думки справедливі, як і багато інших. Одне безсумнівно – створення мережі Internet є видатним досягненням людства. Важливе місце в Internet займає комп'ютерна графіка. Усе більше вдосконалюються способи передачі візуальної інформації, розробляються більш досконалі графічні формати, відчутно бажання використати тривимірну графіку, анімацію, весь спектр мультимедіа.

Таким чином, необхідність застосування комп'ютерної графіки при різного роду дослідженнях очевидна. Вона обумовлена відносно невеликими витратами на проведення робіт, наочністю результатів, простотою використання отриманих програмних продуктів та ін. Варто відзначити, також,

що в розробці подібних додатків беруть участь висококваліфіковані фахівці в різних областях математики, таких як: аналітична геометрія, алгебра, диференціальна геометрія, геометричне моделювання, чисельні методи й ін. Такий підхід до створення різних графічних пакетів дозволяє одержувати програмні продукти високої якості, які застосовуються не тільки для малювання, але й для різних досліджень, за допомогою отриманих зображень.

РОЗДІЛ 2

МОДЕЛЮВАННЯ ПОВЕРХОНЬ

2.1. Математична модель поверхонь

Будь-яка складна поверхня може бути представлена як якийсь конгломерат більш простих поверхонь, які є її складовими частинами. Маючи інформацію про моделі цих частин і їхніх розташувань друг щодо друга, можна одержувати моделі поверхонь, що описують складні ландшафтні форми.

Поверхні можна описувати за допомогою скалярних величин, векторів, ліній і інших поверхонь. У даній главі ми розглянемо способи побудови різних поверхонь, якщо відомо інформацію, достатню для їхнього моделювання.

Нехай у просторі задана деяка глобальна декартова прямокутна система координат з базисними векторами $\bar{e}_1, \bar{e}_2, \bar{e}_3$. Представимо поверхню векторною функцією

$$\bar{r}(u, v) = \sum_{i=1}^3 r_i(u, v) e_i, \quad u, v \in \Omega \quad (2.1)$$

двох скалярних аргументів u і v . Будемо вважати, що компоненти $r_1(u, v)$, $r_2(u, v)$, $r_3(u, v)$ радіуса-вектора точки поверхні $\bar{r}(u, v)$ є однозначними безперервними функціями параметрів u і v . Такий опис поверхонь називається параметричним.

Область зміни параметрів поверхні Ω в загальному випадку являє собою довільну зв'язну двомірну область на площині uv , обмежену двомірними контурами $c_i(t_i)$, $t_{i\min} \leq t_i \leq t_{i\max}$, $i = 1, 2, \dots, n$. Параметри поверхні є координатами деякої двомірної точки $\bar{p} = [u \ v]^T$ на параметричній площині, що належить області Ω . Область визначення параметрів поверхні має зовнішню границю й може мати вирізи усередині, але вона повинна бути зв'язною, так щоб з кожної точки області можна було потрапити в будь-яку іншу точку, не перетинаючи границі області.

В окремому випадку область визначення параметрів поверхні може являти собою прямокутник, коли параметри приймають значення в межах

$u_{\min} \leq u \leq u_{\max}$, $v_{\min} \leq v \leq v_{\max}$ Прямокутні області визначення параметрів зручні для практичного застосування й часто використовуються, тому що їхня математична модель досить проста. Прямокутні області визначення параметрів мають вихідні або базові поверхні, які служать заготівлями для більш складних поверхонь. Приклад області визначення параметрів поверхні наведений на рис. 2.1.

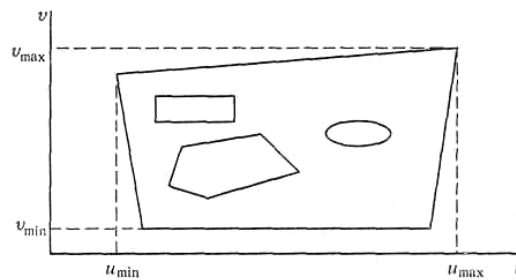


Рис. 2.1. Область визначення поверхні Ω на площині параметрів

Поверхні можуть бути замкнутими або розімкнутими. Замкнуті поверхні будуть мати прямокутні області визначення параметрів. Для замкнутої по u лінії поверхні радіус-вектор $\bar{r}(u \pm (u_{\max} - u_{\min}), v) = \bar{r}(u, v)$; для замкнутої по v лінії поверхні радіус-вектор $\bar{r}(u, v \pm (v_{\max} - v_{\min})) = \bar{r}(u, v)$. Поверхня може бути замкнута по обох координатних лініях.

Деякі поверхні можна описати за допомогою рівняння $F(r_1, r_2, r_3) = 0$, якому задовольняють компоненти радіуса-вектора її точок. Такий опис поверхні називається неявним. Для деякої точки рівняння поверхні можна переписати у вигляді, розв'язаному щодо однієї з координат, наприклад $r_3 = f_3(r_1, r_2)$. Неявний опис поверхні на відміну від параметричного опису не завжди однозначний, що викликає певні труднощі. Крім того, при переході в іншу систему координат рівняння, що описує поверхню, змінюється. Для моделювання поверхонь будемо використовувати їхнє параметричне представлення, як найбільш просте при програмній реалізації.

Математична модель поверхні містить у собі функціональну залежність радіуса-вектора $\bar{r}(u, v)$ від параметрів і двомірну область Ω зміни параметрів u

і v . Таким чином, математична модель поверхні в загальному випадку повинна містити й математичну модель області визначення своїх параметрів.

Спілкування з математичною моделлю поверхні відбувається аналогічно спілкуванню із кривою [5]. Звертаємося до функцій поверхні з деякими значеннями двох її параметрів і як відповідь одержуємо геометричну інформацію про поверхні в точці, що відповідає даним значенням параметрів. Якщо деяка функція поверхні допускає вихід значень параметрів за область визначення, то дана функція повинна видавати геометричну інформацію для продовження поверхні.

2.2. Аналітичні поверхні

Координати радіуса-вектора деяких поверхонь можуть бути представлені аналітичними функціями двох параметрів. Такі поверхні назвемо аналітичними. До них відносяться площина, поверхня тора, поверхні другого порядку й ряд інших.

Опис аналітичних поверхонь зручно представляти у вигляді

$$\bar{r}(u, v) = \bar{p} + x(u, v)\bar{i}_x + y(u, v)\bar{i}_y + z(u, v)\bar{i}_z,$$

де \bar{p} – деяка точка прив'язки характерної точки поверхні, \bar{i}_x , \bar{i}_y , \bar{i}_z – взаємно ортогональні вектори одиничної довжини, $x(u, v)$, $y(u, v)$ і $z(u, v)$ – координатні функції. Точка \bar{p} й вектори \bar{i}_x , \bar{i}_y , \bar{i}_z представляють місцеву декартову систему координат, а функції $x(u, v)$, $y(u, v)$ і $z(u, v)$ можна розглядати як координати радіуса-вектора поверхні в місцевій системі координат. Орти \bar{i}_x , \bar{i}_y , \bar{i}_z можуть бути представлені у вигляді розкладання у базисі \bar{e}_1 , \bar{e}_2 , \bar{e}_3 глобальної декартової системи координат

$$\bar{i}_x = a_{11}\bar{e}_1 + a_{12}\bar{e}_2 + a_{13}\bar{e}_3,$$

$$\bar{i}_y = a_{21}\bar{e}_1 + a_{22}\bar{e}_2 + a_{23}\bar{e}_3,$$

$$\bar{i}_z = a_{31}\bar{e}_1 + a_{32}\bar{e}_2 + a_{33}\bar{e}_3.$$

Точка \bar{p} також може бути представлена у вигляді розкладання у базисі \bar{e}_1 , \bar{e}_2 , \bar{e}_3 глобальної декартової системи координат $\bar{p} = p_1\bar{e}_1 + p_2\bar{e}_2 + p_3\bar{e}_3$.

Положення й орієнтація місцевої декартової системи координат становлять частину структури даних аналітичних поверхонь. Перетворення з місцевої системи координат у глобальну систему координат і назад описуються відомими співвідношеннями аналітичної геометрії. Для опису аналітичних поверхонь зручно використовувати параметричні залежності, що дають однозначні функції радіуса-вектора. Описують їх, як правило, за допомогою місцевої декартової системи координат, у якій запис має найбільш простий вид, що називається канонічним.

Математичний опис аналітичних поверхонь (як у параметричній формі, так і в явній та неявній) можна знайти в будь-якій літературі, присвяченій геометричному моделюванню [5, 8, 9, 10]. Тому в даній роботі він розглядатися не буде.

2.3. Поверхні руху

Поверхню можна одержати шляхом руху кривої по заданій траєкторії. Такі поверхні називаються поверхнями руху. Область зміни параметрів таких поверхонь являє собою прямокутник. Крива, рухом якої утворюється поверхня, називається утворюючою, а траєкторія руху деякої її точки називається напрямною. Поверхня руху у якості даних містить утворюючу криву й у тім або іншому виді напрямну лінію. У загальному випадку напрямною може бути довільна крива. Серед усіляких напрямних виділяють дві найбільш прості – відрізки прямої лінії й дугу окружності. Якщо напрямною служить відрізок прямої, то поверхня називається поверхнею видавлювання. Якщо напрямною служить дуга окружності або вся окружність, то поверхня називається поверхнею обертання. У всіх інших випадках поверхню будемо називати кінематичною поверхнею. Для побудов не будемо використовувати поверхні, що перетинають самі себе.

2.3.1. Поверхня видавлювання

Поверхня видавлювання, отримана рухом кривій $\bar{c}(t)$, $t_{\min} \leq t \leq t_{\max}$, уздовж вектора \bar{d} , описується радіусом-вектором

$$\bar{r}(u, v) = \bar{c}(u) + v\bar{d}, \quad t_{\min} \leq u \leq t_{\max}, \quad 0 \leq v \leq 1. \quad (2.2)$$

Довжина поверхні видавлювання визначається довжиною вектора \bar{d} . Залежно від замкнутості кривій $\bar{c}(t)$ поверхня видавлювання може бути замкнутою по параметру u чи незамкнутою. Напрямний вектор $\bar{d} = \bar{i}d$ може бути представлений у вигляді добутку вектора одиничної довжини на скаляр. Поверхня видавлювання наведена на рис. 2.2.



Рис. 2.2. Поверхня видавлювання

2.3.2. Поверхня обертання

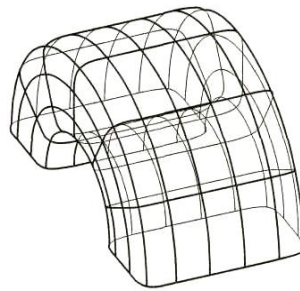


Рис. 2.3. Поверхня обертання

Поверхня, отримана обертанням кривої $\bar{c}(t)$, $t_{\min} \leq t \leq t_{\max}$ на кут α навколо осі, заданої одиничним вектором \bar{i} і точкою \bar{p} , описується радіусом-вектором

$$\begin{aligned} \bar{r}(u, v) = & \bar{p} + ((\bar{c}(u) - \bar{p}) \cdot \bar{i}) \bar{i} + (\bar{c}(u) - \bar{p} - ((\bar{c}(u) - \bar{p}) \cdot \bar{i}) \bar{i}) \cos v + \\ & + \bar{i} \times (\bar{c}(u) - \bar{p}) \sin v = \bar{p} + \bar{r}_3(u) + \bar{r}_1(u) \cos v + \bar{r}_2(u) \sin v, \end{aligned} \quad (2.3)$$

$$t_{\min} \leq u \leq t_{\max}, \quad 0 \leq v \leq \alpha.$$

де $\bar{r}_3(u) = ((\bar{c}(u) - \bar{p}) \cdot \bar{i}) \bar{i}$ – складова вектора $\bar{c}(u) - \bar{p}$, паралельна осі обертання, $\bar{r}_1(u) = \bar{c}(u) - \bar{p} - \bar{r}_3(u)$ – складова вектора $\bar{c}(u) - \bar{p}$, перпендикулярна осі обертання, $\bar{r}_2(u) = \bar{i} \times (\bar{c}(u) - \bar{p})$ – ортогональний першим двом векторам вектор, довжина якого дорівнює довжині вектора $\bar{r}_1(u)$. Замкнутість поверхні обертання по параметру u збігається із замкнутістю утворюючої кривої $\bar{c}(t)$. Якщо кут

обертання $\alpha = 2\pi$, то поверхня (2.3) замкнута по параметру v , якщо $\alpha < 2\pi$, то поверхня обертання не замкнута по параметру v . Як правило, розглядають поверхні обертання, у яких кут обертання не перевищує 2π . Поверхня обертання наведена на рис. 2.3.

2.3.3. Кінематичні поверхні

Розглянемо кінематичну поверхню, що є загальним випадком поверхні руху. Нехай утворююча описується кривою $\bar{c}(t)$, $t_{\min} \leq t \leq t_{\max}$, а напрямною є крива $\bar{g}(v)$, $v_{\min} \leq v \leq v_{\max}$. Нехай при побудові кінематичної поверхні параметр u збігається з параметром утворюючої кривій $\bar{c}(u)$, а параметр v збігається з параметром напрямної кривій $\bar{g}(v)$. При русі утворюючої кривій уздовж напрямної кривій орієнтація першої щодо другої може мінятися або може залишатися незмінною. Якщо утворююча виконує площинопаралельний рух, тобто залишається «паралельною» своєму початковому положенню, то таку поверхню будемо називати поверхнею зрушення, у протилежному випадку поверхню будемо називати поверхнею замітання.

2.3.3.1. Поверхня зрушення



Рис. 2.4. Поверхня зрушення

Поверхня зрушення простіше поверхні замітання. Радіус-вектор поверхні зрушення визначається як

$$\bar{r}(u, v) = \bar{g}(v) + (\bar{c}(u) - \bar{g}(v_{\min}) - \bar{h}), \quad t_{\min} \leq u \leq t_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \quad (2.4)$$

де \bar{h} – вектор прив'язки утворюючої до напрямної. Вектор \bar{h} зрушує криву $\bar{c}(u)$ на деяку величину з її початкового положення щодо напрямної й зберігає це зрушення під час руху. Якщо $\bar{h} = \bar{0}$, то зберігається вихідне положення утворюючої кривій щодо початкової точки напрямної кривої. Радіус-вектор

поверхні зрушення будується як сума двох векторів: вектора точки на напрямній кривій $\bar{g}(v)$ й вектора положення точки утворюючої щодо початкової точки напрямної зі зрушенням $\bar{c}(u) - \bar{g}(v_{\min}) - \bar{h}$. Поверхня зрушення наведена на рис. 2.4.

2.3.3.2. Поверхня замітання

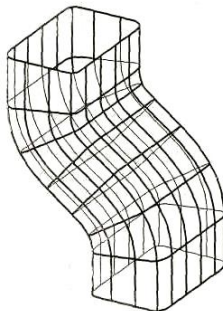


Рис. 2.5. Поверхня замітання

Для побудови кінематичної поверхні замітання (рис. 2.5) будемо використовувати рухливий дотичний базис, пов'язаний з напрямною кривою $\bar{g}(v)$. Рухливий дотичний базис являє собою місцеву декартову прямокутну систему координат. Її початок перебуває в поточній точці напрямної кривої й визначається радіусом-вектором $\bar{p}_0 = \bar{g}(v)$. Перший базисний орт $\bar{i}_1 = \frac{\bar{g}'}{|\bar{g}'|}$ місцевої системи координат збігається з дотичної до напрямній кривій, другий базисний орт \bar{i}_2 є ортогональним першому, а третій $\bar{i}_3 = \bar{i}_1 \times \bar{i}_2$ утворює із першим й другим праву трійку векторів. Зі сказаного не ясно, у яку сторону спрямований вектор \bar{i}_2 . На перший погляд, його можна було б направити по вектору головної нормалі до напрямної, але в точках «розпрямлення» кривій нормаль не визначена, а в точках перегину її напрямок може різко помінятися на протилежний. Тому вектор \bar{i}_2 будемо будувати в такий спосіб: нехай \bar{d} є деякий вектор, що не змінюється, який не є колінеарним дотичній до напрямній $\bar{g}(v)$ у жодній її точці. Тоді вектор \bar{i}_2 приймемо паралельним вектору

$$\bar{d}_2 = \bar{d} - (\bar{i}_1 \cdot \bar{d}) \bar{i}_1 \quad (2.5)$$

Вектор \bar{d}_2 являє собою складову вектора \bar{d} , перпендикулярну орту \bar{i}_1 (дотичній до напрямній у даній точці). Складова \bar{d}_2 ніколи не дорівнює нулю, тому що ми припускаємо, що \bar{d} ніколи не є колінеарним \bar{i}_1 . Будемо вважати, що для будь-якої кривої існує такий вектор \bar{d} , що ніколи не є колінеарним дотичній до кривої. Рухливий дотичний базис пов'язаний із кривою $\bar{g}(v)$ й тому є функцією параметра v

$$\bar{p}_0(v) = \bar{g}(v), \quad (2.6)$$

$$\bar{i}_1(v) = \frac{\bar{g}'}{|\bar{g}'|}, \quad \bar{i}_2(v) = \frac{\bar{d}_2}{|\bar{d}_2|}, \quad \bar{i}_3(v) = \bar{i}_1 \times \bar{i}_2,$$

де \bar{d}_2 визначається рівністю (2.5), а $\bar{g}' = \frac{d\bar{g}}{dv}$.

Запам'ятаємо положення утворюючої кривої $\bar{c}(u)$ в рухливому дотичному базисі на початку напрямної (при початковому значенні параметра $v = v_{\min}$) і будемо зберігати його при русі уздовж напрямної. При русі уздовж напрямної рухливий дотичний базис міняє своє положення й орієнтацію в просторі й захоплює за собою жорстко пов'язану з ним утворюючу криву. Таким чином, у даних кінематичної поверхні перебувають дві криві лінії, вектор \bar{d} і положення дотичного базису (2.6) на початку напрямної: $\bar{g}(v_{\min})$, $\bar{i}_1(v_{\min})$, $\bar{i}_2(v_{\min})$, $\bar{i}_3(v_{\min})$. У загальному випадку можна запам'ятати положення утворюючій кривій щодо початкової точки напрямної, зрушивши першу на деякий вектор \bar{h} (наприклад, щоб по напрямній рухався центр ваги утворюючої), і зберегти його при русі уздовж напрямної. Якщо $\bar{h} = \bar{0}$, то зберігається вихідне положення утворюючої кривої щодо початкової точки напрямної кривої й дотичній до неї. Радіус-вектор поверхні для довільних параметрів u і v одержимо в такий спосіб. Обчислимо положення точки на утворюючій кривій у місцевій системі координат при $v = v_{\min}$ по формулі

$$\bar{x}(u, v_{\min}) = A(v_{\min}) \cdot (\bar{c}(u) - \bar{g}(v_{\min}) - \bar{h}),$$

де через $A(v_{\min})$ позначена матриця, обчислена по матричній функції $A(v)$ при початковому параметру v . Матрична функція $A(v)$ є матрицею перетворення

координат радіуса-вектора точки при переході із глобальної в місцеву систему координат [5], її рядки складені з компонентів базисних ортів місцевої системи координат

$$A(v) = \begin{bmatrix} \bar{i}_1(v) \\ \bar{i}_2(v) \\ \bar{i}_3(v) \end{bmatrix}.$$

Вектор $\bar{x}(u, v_{\min})$ виражає положення точки утворюючої щодо точки на напрямній кривій у рухливому дотичному базисі, що зберігається для довільного параметра v . Переходячи з місцевої системи координат у глобальну систему координат уже при поточному параметру v відповідно одержимо радіус-вектор точки на поверхні

$$\bar{r}(u, v) = \bar{g}(v) + A^T(v) \bar{x}(u, v_{\min}).$$

Таким чином, радіус-вектор кінематичної поверхні замітання описується співвідношенням

$$\bar{r}(u, v) = \bar{g}(v) + M(v) \cdot (\bar{c}(u) - \bar{g}(v_{\min}) - \bar{h}), \quad t_{\min} \leq u \leq t_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \quad (2.7)$$

де $M(v)$ – матрична функція, що дає матрицю повороту поточного рухливого дотичного базису щодо його початкового положення. Ця матриця обчислюється по формулі

$$M(v) = A^T(v) \cdot A(v_{\min}) = \begin{bmatrix} \bar{i}_1(v) \\ \bar{i}_2(v) \\ \bar{i}_3(v) \end{bmatrix}^T \cdot \begin{bmatrix} \bar{i}_1(v_{\min}) \\ \bar{i}_2(v_{\min}) \\ \bar{i}_3(v_{\min}) \end{bmatrix}. \quad (2.8)$$

На початку напрямної кривої матриця $M(v_{\min})$ дорівнює одиничній матриці, тому що $A^T = A^{-1}$.

Для обчислення похідних радіуса-вектора кінематичної поверхні замітання потрібно обчислювати похідні матриці $M(v)$, що у свою чергу зводиться до обчислення похідних ортів $\bar{i}_1(v)$, $\bar{i}_2(v)$, $\bar{i}_3(v)$. Розглянемо обчислення похідних векторів одиничної довжини на прикладі обчислення похідних орта \bar{i}_1 . Цей орт отриманий нормуванням вектора $\bar{g}(v)$. Обчислимо першу, другу й третю похідні цього вектора – \bar{g}' , \bar{g}'' і \bar{g}''' . По цих похідних сам

орт \bar{i}_1 і його перша й друга похідні (\bar{i}_1' і \bar{i}_1'') визначаються з наступних співвідношень:

$$\bar{i}_1 = \frac{\bar{g}'}{|\bar{g}'|},$$

$$\frac{d\bar{i}_1}{dv} = \frac{\bar{g}''}{|\bar{g}'|} - \left(\bar{i}_1 \cdot \frac{\bar{g}''}{|\bar{g}'|} \right) \bar{i}_1, \quad (2.9)$$

$$\frac{d^2\bar{i}_1}{dv^2} = \frac{\bar{g}'''}{|\bar{g}'|} - 2 \left(\bar{i}_1 \cdot \frac{\bar{g}''}{|\bar{g}'|} \right) \frac{d\bar{i}_1}{dv} - \left(\bar{i}_1 \cdot \frac{\bar{g}'''}{|\bar{g}'|} + \frac{d\bar{i}_1}{dv} \cdot \frac{\bar{g}''}{|\bar{g}'|} \right) \bar{i}_1,$$

де друга й третя формули отримані шляхом диференціювання першої з урахуванням того, що довжина похідної напрямної кривої $|\bar{g}'| \equiv \left| \frac{d\bar{g}}{dv} \right| = \sqrt{\bar{g}' \cdot \bar{g}'}$.

Аналогічно одержимо похідні інших ортів. Орт \bar{i}_2 отриманий нормуванням вектора $\bar{d}_2 = \bar{d} - (\bar{i}_1 \cdot \bar{d}) \bar{i}_1$, а орт \bar{i}_3 отриманий нормуванням вектора $\bar{d}_3 = \bar{g}' \times \bar{d}_2$. Обчисливши перші, другі й треті похідні \bar{d}_2 й \bar{d}_3 по (2.9), одержимо похідні відповідних ортів. Далі обчислимо похідні матричних функцій $A(v)$ і $M(v)$, що дозволить обчислити похідні радіуса-вектора поверхні (2.7).

Замкнутість кінематичної поверхні по параметру u збігається із замкнутістю утворюючої кривій $\bar{c}(u)$, а замкнутість поверхні по параметру v збігається із замкнутістю напрямної кривої $\bar{g}(v)$. На рис. 2.6 показана кінематична поверхня, отримана рухом еліпса уздовж спіральної лінії.

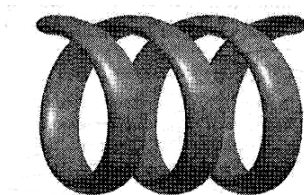


Рис. 2.6. Кінематична поверхня

При невеликому сполученні параметрів кривих кінематична поверхня може перетинати сама себе.

З (2.9) виходить, що для обчислення часток похідних другого порядку радіуса-вектора поверхні потрібні похідні радіуса-вектора кривій третього порядку. Для одержання всієї геометричної інформації про криву досить знати

векторну функцію її радіуса-вектора і її першу й другу похідні, але для побудови деяких поверхонь потрібна третя похідна кривої. Кінематична поверхня є однією з них.

2.4. Лінійчаті поверхні

Лінійчата поверхня являє собою геометричне місце відрізків прямих, що з'єднують відповідні точки заданих двох ліній. Її можна одержати шляхом руху прямої лінії по двох напрямних кривих лініях, при цьому кожній точці однієї кривої повинна відповідати цілком певна точка іншої кривої. Нехай задані дві напрямні криві $\bar{a}(t)$ й $\bar{d}(\omega)$, де параметри кривих t і ω змінюються в межах $t_{\min} \leq t \leq t_{\max}$ і $\omega_{\min} \leq \omega \leq \omega_{\max}$. Радіус-вектор лінійчатої поверхні визначається співвідношенням

$$\bar{r}(u, v) = (1 - v)\bar{a}(t) + v\bar{d}(\omega), \quad (2.10)$$

$$t = t_{\min}(1 - u) + t_{\max}u, \quad \omega = \omega_{\min}(1 - u) + \omega_{\max}u,$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

Для напрямних кривих неявно виробляється репараметризація – приведення області зміни параметрів до відрізка від 0 до 1. Можна було б репараметризувати тільки одну криву – привести область зміни однієї з них до іншої. Якщо параметризація кривих $\bar{a}(u)$ і $\bar{d}(u)$ збігається, то радіус-вектор лінійчатої поверхні може бути описаний функцією

$$\bar{r}(u, v) = (1 - v)\bar{a}(u) + v\bar{d}(u), \quad u_{\min} \leq u \leq u_{\max}, \quad 0 \leq v \leq 1.$$

На рис. 2.7 наведений приклад лінійчатої поверхні.

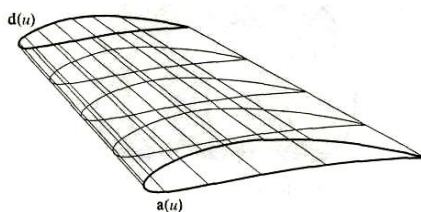


Рис. 2.7. Лінійчата поверхня

Якщо обидві напрямні замкнуті, то лінійчата поверхня замкнута по параметру u . По іншому параметру лінійчата поверхня завжди не замкнута.

Якщо напрямними лініями є відрізки прямих ліній, то поверхня є лінійчатою по обох параметрах і для її побудови досить знати радіуси-вектори кінцевих точок відрізків.

Нехай один напрямний відрізок у формі $\bar{r}(t) = v\bar{p}_1 + t\bar{p}_2$, $0 \leq t \leq 1$, $0 \leq v \leq 1$, $t + v = 1$ проведений із точки \bar{p}_1 в точку \bar{p}_2 , а інший напрямний відрізок проведений із точки \bar{p}_3 в точку \bar{p}_4 . Тоді лінійчата по двох параметрах поверхня визначиться векторною функцією

$$\begin{aligned} \bar{r}(u, v) &= (1-v)(\bar{p}_1(1-u) + \bar{p}_2u) + v(\bar{p}_3(1-u) + \bar{p}_4u) = \\ &= (1-u)(1-v)\bar{p}_1 + u(1-v)\bar{p}_2 + (1-u)v\bar{p}_3 + uv\bar{p}_4 \end{aligned} \quad (2.11)$$

$$0 \leq u \leq 1, 0 \leq v \leq 1.$$

Така поверхня називається білінійною. Вона завжди не замкнута. Якщо всі чотири точки лежать в одній площині, то поверхня являє собою частину площини. У деяких випадках зручно описувати площину по чотирьох точках, що лежать у кутах квадрата, у вигляді залежності (2.11). Описана в такий спосіб площина відрізняється тим, що її параметричні розміри не залежать від розмірів самої площини. При виконанні перетинань і інших операцій над поверхнями бажано, щоб у поверхонь області визначення параметрів не залежали від розмірів поверхонь. Якщо точки \bar{p}_1 й \bar{p}_3 або \bar{p}_2 й \bar{p}_4 збігаються, то ми одержимо трикутну поверхню, що завжди плоска.

Частковим випадком лінійчатої поверхні є секторіальна поверхня. Вона утворюється із лінійчатої поверхні (2.10), якщо одну із кривих, наприклад $\bar{d}(\omega)$, замінити точкою \bar{p} . Радіус-вектор секторіальній поверхні описується вираженням

$$\bar{r}(u, v) = (1-v)\bar{a}(u) + v\bar{p}, \quad t_{\min} \leq u \leq t_{\max}, \quad 0 \leq v \leq 1 \quad (2.12)$$

Якщо напрямна $\bar{a}(t)$ замкнута, то секторіальна поверхня також замкнута по параметру u . По іншому параметру секторіальна поверхня завжди не замкнута. Трикутна й секторіальна поверхні мають точку, у якій одна із частинних похідних її радіуса-вектора дорівнює нулю. Аналогічні точки мають сферична й конічна поверхні. Ці точки не є особливими, тому що інша частинна похідна

радіуса-вектора поверхні не дорівнює нулю. Якщо розглядати криві на поверхнях, то в таких точках перебувають відрізки двомірних координатних ліній.

2.5. Поверхні Кунса

Візьмемо векторну функцію лінійчатої поверхні (2.10), додамо до неї й віднімемо від неї векторну функцію білінійної поверхні (2.11), де \bar{p}_1 й \bar{p}_2 – кінцеві точки напрямної кривої $\bar{a}(t)$, а \bar{p}_3 й \bar{p}_4 – кінцеві точки напрямної кривої $\bar{d}(\omega)$. У результаті цих дій векторна функція не зміниться, але буде мати вигляд

$$\begin{aligned} \bar{r}(u, v) = & (1-v)\bar{a}(u) + v\bar{d}(u) + (1-u)(\bar{p}_1(1-v) + \bar{p}_3v) + \\ & + u(\bar{p}_2(1-v) + \bar{p}_4v) - (1-u)(1-v)\bar{p}_1 - u(1-v)\bar{p}_2 - \\ & - (1-u)v\bar{p}_3 - uv\bar{p}_4 = (1-v)\bar{a}(u) + v\bar{d}(u) + (1-u)\bar{b}(v) + u\bar{c}(v) - \\ & - (1-u)(1-v)\bar{p}_1 - u(1-v)\bar{p}_2 - (1-u)v\bar{p}_3 - uv\bar{p}_4, \end{aligned} \quad (2.13)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

У позначеннях функцій $\bar{a}(u)$ і $\bar{d}(u)$ ми використали їхній реальний аргумент u , а також увели позначення для відрізків прямих $\bar{b}(v) = \bar{p}_1(1-v) + \bar{p}_3v$, $\bar{c}(v) = \bar{p}_2(1-v) + \bar{p}_4v$, що з'єднують кінці кривих $\bar{a}(u)$ і $\bar{d}(u)$. Якщо допустити, що в якості $\bar{b}(v)$ й $\bar{c}(v)$ можуть використовуватися довільні криві, що починаються й кінчаються в тих же точках, то одержимо варіант поверхні Кунса.

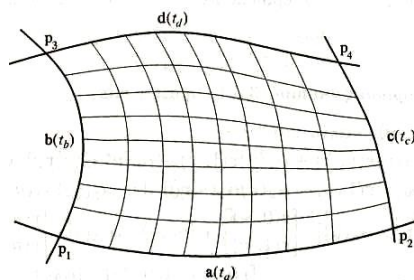


Рис. 2.8. Поверхня Кунса

Поверхня Кунса будується по чотирьох рівноправних кривих. Розглянемо чотири криві $\bar{a}(t_a)$, $\bar{d}(t_d)$, $\bar{b}(t_b)$, $\bar{c}(t_c)$, що попарно перетинаються в точках \bar{p}_1 , \bar{p}_3 , \bar{p}_2 , \bar{p}_4 , як показано на рис. 2.8. Побудуємо поверхню усередині «чотирикутника», утвореного кривими. Нехай точкам \bar{p}_1 , і \bar{p}_2 на кривій $\bar{a}(t_a)$

відповідають параметри $t_{a \min}$ й $t_{a \max}$ точкам \bar{p}_1 і \bar{p}_3 на кривій $\bar{b}(t_b)$ відповідають параметри $t_{b \min}$ й $t_{b \max}$ точкам \bar{p}_3 і \bar{p}_4 на кривій $\bar{d}(t_d)$ відповідають параметри $t_{d \min}$ й $t_{d \max}$ точкам \bar{p}_2 і \bar{p}_4 на кривій $\bar{c}(t_c)$ відповідають параметри $t_{c \min}$ й $t_{c \max}$. Поверхню, побудовану по даним чотирьом кривим, можна описати радіусом-вектором

$$\begin{aligned}
\bar{r}(u, v) &= (1-v)(\bar{a}(t_a) - (1-u)\bar{p}_1) + (1-u)(\bar{b}(t_b) - v\bar{p}_3) + \\
&\quad + v(\bar{d}(t_d) - u\bar{p}_4) + u(\bar{c}(t_c) - (1-v)\bar{p}_2) = \\
&= (1-v)\bar{a}(t_a) + (1-u)\bar{b}(t_b) + v\bar{d}(t_d) + u\bar{c}(t_c) - \\
&\quad - (1-u)(1-v)\bar{p}_1 - u(1-v)\bar{p}_2 - (1-u)v\bar{p}_3 - uv\bar{p}_4, \tag{2.14} \\
t_a &= t_{a \min}(1-u) + t_{a \max}u, \quad t_b = t_{b \min}(1-v) + t_{b \max}v, \\
t_c &= t_{c \min}(1-v) + t_{c \max}v, \quad t_d = t_{d \min}(1-u) + t_{d \max}u, \\
0 &\leq u \leq 1, \quad 0 \leq v \leq 1.
\end{aligned}$$

Легко перевірити, що вектор $\bar{r}(u, 0)$ описує ділянку кривій $\bar{a}(t_a)$, вектор $\bar{r}(u, 1)$ описує ділянку кривій $\bar{d}(t_d)$, вектор $\bar{r}(0, v)$ описує ділянку кривій $\bar{b}(t_b)$, вектор $\bar{r}(1, v)$ описує ділянку кривої $\bar{c}(t_c)$. Зовсім не обов'язково, щоб $t_{k \min}$ було менше $t_{k \max}$, $k = a, b, c, d$.

Для того щоб співвідношення (2.14) легше сприймалося, перепозначимо криві й точки, що входять у структуру даних описуваної нею поверхні: криву $\bar{a}(t_a)$ позначимо через $\bar{r}(u, 0)$, криву $\bar{d}(t_d)$ позначимо через $\bar{r}(u, 1)$, криву $\bar{b}(t_b)$ позначимо через $\bar{r}(0, v)$, криву $\bar{c}(t_c)$ позначимо через $\bar{r}(1, v)$, точку \bar{p}_1 позначимо через $\bar{r}(0, 0)$, точку \bar{p}_2 позначимо через $\bar{r}(1, 0)$, точку \bar{p}_3 позначимо через $\bar{r}(0, 1)$, точку \bar{p}_4 позначимо через $\bar{r}(1, 1)$, і будемо вважати, що для кривих виконано перехід до параметрів u і v відповідно до (2.14). Для коефіцієнтів, з якими граничні криві входять в (2.14), уведемо позначення

$$\begin{aligned}
\alpha_0(u) &= 1-u, \quad \alpha_1(u) = u, \tag{2.15} \\
\alpha_0(v) &= 1-v, \quad \alpha_1(v) = v.
\end{aligned}$$

У результаті векторна функція (2.13) прийме вид

$$\begin{aligned}
\bar{r}(u, v) &= \alpha_0(v)\bar{r}(u, 0) + \alpha_0(u)\bar{r}(0, v) + \alpha_1(v)\bar{r}(u, 1) + \alpha_1(u)\bar{r}(1, v) - \\
&- \alpha_0(u)\alpha_0(v)\bar{r}(0, 0) - \alpha_1(u)\alpha_0(v)\bar{r}(1, 0) - \alpha_0(u)\alpha_1(v)\bar{r}(0, 1) - \alpha_1(u)\alpha_1(v)\bar{r}(1, 1) = \\
&= [\alpha_0(u) \quad \alpha_1(u)] \cdot \begin{bmatrix} \bar{r}(0, v) \\ \bar{r}(1, v) \end{bmatrix} + [\alpha_0(v) \quad \alpha_1(v)] \cdot \begin{bmatrix} \bar{r}(u, 0) \\ \bar{r}(u, 1) \end{bmatrix} - \\
&- [\alpha_0(u) \quad \alpha_1(u)] \cdot \begin{bmatrix} \bar{r}(0, 0) & \bar{r}(0, 1) \\ \bar{r}(1, 0) & \bar{r}(1, 1) \end{bmatrix} \cdot \begin{bmatrix} \alpha_0(v) \\ \alpha_1(v) \end{bmatrix} = \\
&= \sum_{i=0}^1 \alpha_i(u)\bar{r}(i, v) + \sum_{j=0}^1 \alpha_j(v)\bar{r}(u, j) + \sum_{i=0}^1 \sum_{j=0}^1 \alpha_i(u)\alpha_j(v)\bar{r}(i, j), \tag{2.16} \\
&0 \leq u \leq 1, \quad 0 \leq v \leq 1.
\end{aligned}$$

Поверхня (2.16) побудована на чотирьох граничних кривих, коефіцієнти при яких є лінійними функціями параметрів u і v . Таке лінійне об'єднання чотирьох кривих називається лінійною поверхнею Кунса. Функції $\alpha_i(u)$ й $\alpha_j(v)$ називаються функціями зсуву. Якщо граничні лінії є відрізками прямих ліній, то формула (2.16) дасть білінійну поверхню (2.11).

Подібною поверхнею можна описати довільну частину площини, границю якої можна представити у вигляді чотирьох попарно пересічних ліній на ній. Наприклад, частину площини, що лежить усередині еліпса, можна описати поверхнею

$$\begin{aligned}
\bar{r}(u, v) &= \bar{p} + a \left(\cos u + \cos v - 1 + \frac{2u(1 - \cos v - \sin v)}{\pi} + \frac{2v(1 - \cos u - \sin u)}{\pi} \right) \bar{i}_1 + \\
&+ b \left(\sin u + \sin v - \frac{2u(1 - \cos v - \sin v)}{\pi} + \frac{2v(1 - \cos u - \sin u)}{\pi} \right) \bar{i}_2 \tag{2.17} \\
&0 \leq u \leq \frac{\pi}{2}, \quad 0 \leq v \leq \frac{\pi}{2},
\end{aligned}$$

де \bar{p} – радіус-вектор центра, a і b – півосі еліпса, \bar{i}_1 , \bar{i}_2 – орти, що лежать у площині еліпса (рис. 2.9).

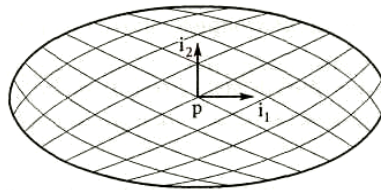


Рис. 2.9. Окремий випадок поверхні Кунса

Виразження (2.17) отримане як окремий випадок співвідношення (2.16) шляхом підстановки в нього конкретних ліній і точок. Поверхня (2.17) побудована по чотирьох дугах еліпса. Дана поверхня відрізняється від свого аналога у вигляді площини з областю зміни параметрів, обмеженої еліпсом, тим, що має прямокутну область зміни параметрів. Варто помітити, що поверхня (2.17) має особливості в кутових точках: похідна по першому параметру є колінеарною похідній по другому параметру. Дуги еліпса в (2.17) представлені в явному виді. Для зручності верхні границі області зміни параметрів замінені з 1 на $\frac{\pi}{2}$. Поверхня (2.17) має чотири особливі точки, у яких частинні похідні радіуса-вектора по параметрах u і v є колінеарними. Це точки стикування дуг еліпса.

З поверхонь (2.16) можна сконструювати складену поверхню, стикуючи їх по граничних кривих (роблячи граничні криві загальними). Якщо поверхня (2.16) стикується по своїх краях з іншою такою ж поверхнею, то на границі похідна в трансверсальному напрямку буде перетерплювати злам. Для того щоб на граничних кривих похідні радіуса-вектора поверхні в трансверсальному до граничних кривих напрямку були заданими функціями, потрібно додати їх до її опису. Тобто необхідно додати до опису поверхні її похідні уздовж граничних кривих $\bar{r}_u(i, v)$, $\bar{r}_v(u, j)$ і похідні в кутах поверхні $\bar{r}_u(i, j)$, $\bar{r}_v(i, j)$, $\bar{r}_{uv}(i, j)$, з коефіцієнтами $\beta_i(u)$, $\beta_j(v)$, де $i=0,1$, $j=0,1$. Лінійні функції зсуву (2.15) замінимо кубічними функціями [9, 12].

Побудована в такий спосіб поверхня описується векторною функцією

$$\begin{aligned} \bar{r}(u,v) = & [\alpha_0(u) \quad \alpha_1(u) \quad \beta_0(u) \quad \beta_1(u)] \cdot \begin{bmatrix} \bar{r}(0,v) \\ \bar{r}(1,v) \\ \bar{r}_u(0,v) \\ \bar{r}_u(1,v) \end{bmatrix} + \\ & + [\alpha_0(v) \quad \alpha_1(v) \quad \beta_0(v) \quad \beta_1(v)] \cdot \begin{bmatrix} \bar{r}(u,0) \\ \bar{r}(u,1) \\ \bar{r}_u(u,0) \\ \bar{r}_u(u,1) \end{bmatrix} - \\ & - [\alpha_0(u) \quad \alpha_1(u) \quad \beta_0(u) \quad \beta_1(u)] \begin{bmatrix} \bar{r}(0,0) & \bar{r}(0,1) & \bar{r}_v(0,0) & \bar{r}_v(0,1) \\ \bar{r}(1,0) & \bar{r}(1,1) & \bar{r}_v(1,0) & \bar{r}_v(1,1) \\ \bar{r}_u(0,0) & \bar{r}_u(0,1) & \bar{r}_{uv}(0,0) & \bar{r}_{uv}(0,1) \\ \bar{r}_u(1,0) & \bar{r}_u(1,1) & \bar{r}_{uv}(1,0) & \bar{r}_{uv}(1,1) \end{bmatrix} \cdot \begin{bmatrix} \alpha_0(v) \\ \alpha_1(v) \\ \beta_0(v) \\ \beta_1(v) \end{bmatrix}, \quad (2.18) \\ & 0 \leq u \leq 1, \quad 0 \leq v \leq 1, \end{aligned}$$

де кубічні функції зсуву визначаються формулами

$$\begin{aligned} \alpha_0(\omega) &= 1 - 3\omega^2 + 2\omega^3, \quad \alpha_1(\omega) = 3\omega^2 - 2\omega^3, \\ \beta_0(\omega) &= \omega - 2\omega^2 + \omega^3, \quad \beta_1(\omega) = -\omega^2 + \omega^3. \end{aligned} \quad (2.19)$$

Поверхня (2.18) називається кубічною поверхнею Кунса. Співвідношення (2.18) описує поверхню Кунса, побудовану на тих же чотирьох кривих, показаних на рис. 2.8, що й поверхня (2.16). Але на відміну від останньої вона має задані похідні в трансверсальному напрямку уздовж граничних кривих.

Форрест запропонував узагальнити поверхні (2.16) і (2.18) і використати замість лінійних або кубічних функцій зсуву деякі узагальнені функції зсуву $\alpha_{ki}(u)$. Індекс k указує на те, що узагальнена функція зсуву множиться на векторну функцію, що представляє собою похідну k -го порядку від граничної кривої. Похідною нульового порядку функції будемо називати саму функцію. Для граничних кривих $k=0$, для дотичних векторів $k=1$ або, інакше кажучи, $\alpha_{0i}(u) = \alpha_i(u)$, $\alpha_{1i}(u) = \beta_i(u)$, $i=0,1$. Узагальнені функції зсуву повинні задовольняти рівностям

$$\alpha_{ki}^{(n)}(j) \equiv \left. \frac{d^n \alpha_{ki}}{du} \right|_{u=j} = \delta_i^j \delta_k^n = \begin{cases} 1, & \text{если } i = j \text{ и } k = n, \\ 0, & \text{если } i \neq j \text{ или } k \neq n, \end{cases} \quad (2.20)$$

що являються узагальненням умов, яким задовольняють лінійні й кубічні функції зсуву. Радіус-вектор узагальненої поверхні Кунса визначається співвідношенням

$$\begin{aligned} \bar{r}(u, v) = & \sum_{i=0}^1 \sum_{k=0}^n \alpha_{ki}(u) \bar{r}_u^{(k)}(i, v) + \sum_{j=0}^1 \sum_{l=0}^m \alpha_{lj}(v) \bar{r}_v^{(l)}(u, j) - \\ & - \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^n \sum_{l=0}^m \alpha_{ki}(u) \alpha_{lj}(v) \bar{r}_{uv}^{(k,l)}(i, j) \end{aligned} \quad (2.21)$$

$$0 \leq u \leq 1, 0 \leq v \leq 1,$$

де використовуються задані значення похідних на краях поверхні

$$\bar{r}_u^{(k)}(i, v) = \left. \frac{\partial^k \bar{r}(u, v)}{\partial u^k} \right|_{u=i}, \quad i = 0, 1,$$

$$\bar{r}_v^{(l)}(u, j) = \left. \frac{\partial^l \bar{r}(u, v)}{\partial v^l} \right|_{v=j}, \quad j = 0, 1,$$

і задані значення похідних у кутових точках параметричної області поверхні

$$\bar{r}_{uv}^{(k,l)}(i, j) = \left. \frac{\partial^{k+l} \bar{r}(u, v)}{\partial u^k \partial v^l} \right|_{\substack{u=i \\ v=j}}, \quad i, j = 0, 1.$$

У якості узагальнених функцій зсуву можуть бути використані поліноми. Узагальнені поверхні Кунса мають задані похідні k -го порядку радіуса-вектора по параметру u уздовж граничних кривих $u = const$ і задані похідні l -го порядку радіуса-вектора по параметру v уздовж граничних кривих $v = const$.

2.6. Сплайнові поверхні

Корпуса судів, фюзеляжі й крила літаків і інші об'єкти складної форми часто проектуються за допомогою поперечних або поздовжніх перетинів, що проходять у заданих місцях. Ці перетини з'єднуються плавними кривими, у результаті чого виходить єдина тривимірна форма. Це виконувалося в натуральну величину, для чого використовувалися плази. За традицією процес побудови поверхні по її перетинах називається плазирювання, а поверхні, побудовані таким способом, називаються сплайновими поверхнями. Розглянемо деякі сплайнові поверхні.

2.6.1. Поверхня Ерміта

Нехай ϵ сімейство кривих $\bar{r}_j(t_j)$, $j=0,1,2,\dots,n$, показаних на рис. 2.10. Криві сімейства не повинні перетинати одна одну. На цьому сімействі можна побудувати гладку поверхню, що має безперервні похідні в усіх напрямках. Приведемо всі криві до однієї параметричної довжини, так щоб область зміни параметра була $u_{\min} \leq u \leq u_{\max}$. Криві перепозначимо відповідно до їх місця на поверхні: $\bar{r}(u, j) \equiv \bar{r}_j(t_j)$, $j=0,1,2,\dots,n$. Нехай параметр поверхні v між кривими $\bar{r}(u, i)$ й $\bar{r}(u, k)$ змінюється від i до k .

Радиус-вектор поверхні, що згладжує, побудованої по сімейству перетинів, що є аналогом складеного сплайна Ерміта [5], описується співвідношенням

$$\begin{aligned} \bar{r}(u, v) &= [\alpha_0(\omega) \quad \alpha_1(\omega) \quad \beta_0(\omega) \quad \beta_1(\omega)] \cdot \begin{bmatrix} \bar{r}(u, j) \\ \bar{r}(u, j+1) \\ \bar{r}_v(u, j) \\ \bar{r}_v(u, j+1) \end{bmatrix} = \\ &= \bar{r}(u, j)(1 - 3\omega^2 + 2\omega^3) + \bar{r}(u, j+1)(3\omega^2 - 2\omega^3) + \bar{r}_v(u, j)(\omega - 2\omega^2 + \omega^3) + \\ &\quad + \bar{r}_v(u, j+1)(-\omega^2 + \omega^3) \end{aligned} \quad (2.22)$$

$$\omega = \omega(v) = v - j, \quad u_{\min} \leq u \leq u_{\max}, \quad 0 \leq v \leq n,$$

де j дорівнює цілої частини параметра v й номеру кривій, $n+1$ – число кривих. Дана поверхня є кубічною по v напрямку. Параметр ω є місцевим параметром даної частини поверхні в v напрямку, він змінюється від 0 до 1.

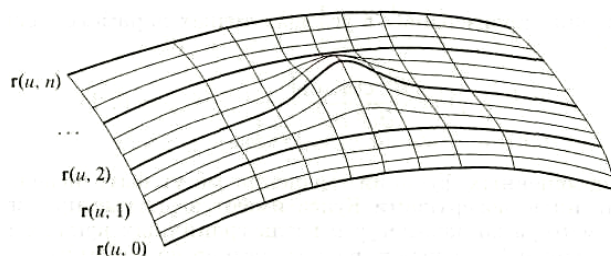


Рис. 2.10. Поверхня на сімействі кривих

Похідні на лініях сітки $\bar{r}_v(u, j)$ можуть бути обчислені по сусідніх лініях, аналогічно тому, як обчислюються похідні в характеристичних точках складеного сплайна Ерміта,

наприклад,

$$\bar{r}_v(u, j) = \frac{\bar{r}(u, j+1) - \bar{r}(u, j-1)}{2}$$

для внутрішніх ліній і

$$\bar{r}_v(u, 0) = 2\bar{r}(u, 1) - 2\bar{r}(u, 0) - \bar{r}_v(u, 1),$$

$$\bar{r}_v(u, n) = 2\bar{r}(u, n) - 2\bar{r}(u, n-1) - \bar{r}_v(u, n-1)$$

для крайніх ліній.

Ці залежності забезпечують рівність нулю третіх похідних на крайніх кривих поверхні в трансверсальних до краю напрямках. Якщо всі криві сімейства є замкнутими, то поверхня (2.22) виходить замкнутою у відповідному напрямку. Поверхня може бути замкнутою й в іншому параметричному напрямку. Похідні $\bar{r}_v(u, j)$ можуть бути обчислені й по інших формулах, що наведені для сплайна Ерміта [5]. Поверхня Ерміта, побудована на сімействі NURBS кривих, наведена на рис. 2.11.

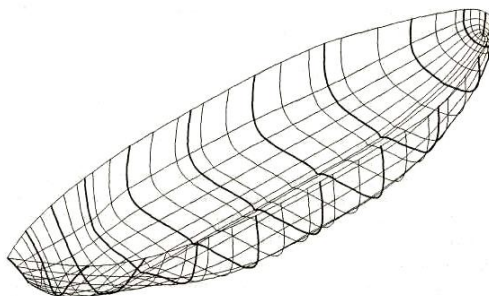


Рис. 2.11. Поверхня Ерміта на сімействі кривих

2.6.2. Поверхня Лагранжа

По сімейству кривих, показаних на рис. 2.10, можна побудувати поверхню, що є аналогом кривої Лагранжа [5]. Радіус-вектор такої поверхні, побудованої по сімейству перетинів $\bar{r}(u, i) \equiv \bar{r}_i(t_i)$, $i = 0, 1, 2, \dots, n$ описується вираженням

$$\bar{r}(u, v) = \sum_{i=0}^n L_i(v) \bar{r}(u, i), \quad u_{\min} \leq u \leq u_{\max}, \quad 0 \leq v \leq n, \quad (2.23)$$

де $L_i(v)$ – коефіцієнти Лагранжа при $t_j = j$:

$$L_i(v) = \frac{\prod_{j=0, j \neq i}^n (v - j)}{\prod_{j=0, j \neq i}^n (i - j)}. \quad (2.24)$$

Кожна v -лінія такої поверхні являє собою криву Лагранжа, побудовану по точках $\bar{r}(u, i)$, $i = 0, 1, 2, \dots, n$, $u = const$. Існують і інші способи побудови поверхні, що згладжує, по сімейству кривих.

2.6.3. Поверхня переходу

Поверхня типу (2.22) може служити поверхнею переходу від однієї поверхні до іншої. Нехай, наприклад, потрібно плавно сполучити край однієї поверхні із краєм іншої. Позначимо край першої поверхні, що сполучає, через $\bar{a}(t)$, а край другої поверхні – через $\bar{b}(t)$, де t – деякий загальний параметр. Через $\bar{a}_\omega(t)$ позначимо частинну похідну першої з поверхонь, що сполучають, у трансверсальному до її краю напрямку, а через $\bar{b}_\omega(t)$ позначимо частинну похідну другої поверхні в трансверсальному до її краю напрямку. Під параметрично трансверсальним напрямком будемо мати на увазі наступне. Якщо перша поверхня описується радіусом-вектором $\bar{p}(u, v)$, то $\bar{a}(t) = \bar{p}(u_{\max}, t)$ й $\bar{a}_\omega(t) = \bar{p}_v(u_{\max}, t)$ описують край поверхні. Аналогічно, якщо друга поверхня описується радіусом-вектором $\bar{q}(u, v)$, то $\bar{b}(t) = \bar{q}(t, v_{\min})$ й $\bar{b}_\omega(t) = \bar{q}_u(t, v_{\min})$ описують край цієї поверхні. Поверхня

$$\begin{aligned} \bar{r}(t, \omega) = & \bar{a}(t)(1 - 3\omega^2 + 2\omega^3) + \bar{b}(t)(3\omega^2 - 2\omega^3) + \\ & + \bar{a}_\omega(t)(\omega - 2\omega^2 + \omega^3) + \bar{b}_\omega(t)(-\omega^2 + \omega^3), \end{aligned} \quad (2.25)$$

$$t_{\min} \leq t \leq t_{\max}, \quad 0 \leq \omega \leq 1,$$

гладко сполучає дві задані поверхні по їхніх краях. Ми розглянули випадок, поверхонь, що сполучають, коли вони мають прямокутні області визначення параметрів. На двох поверхнях, що сполучають, і поверхні переходу може бути побудована єдина гладка поверхня.

2.6.4. Поверхня Гордона

Перейдемо до побудови поверхонь на двох сімействах кривих. Нехай є сітка кривих, показаних на рис. 2.12.

Сітка утворена двома сімействами кривих: $\bar{a}_j(t_j)$, $j = 0, 1, 2, \dots, n$, $\bar{b}_i(\omega_i)$, $i = 0, 1, 2, \dots, m$. Кожна крива одного сімейства перетинає всі криві іншого сімейства й не перетинає ні однієї кривій свого сімейства.

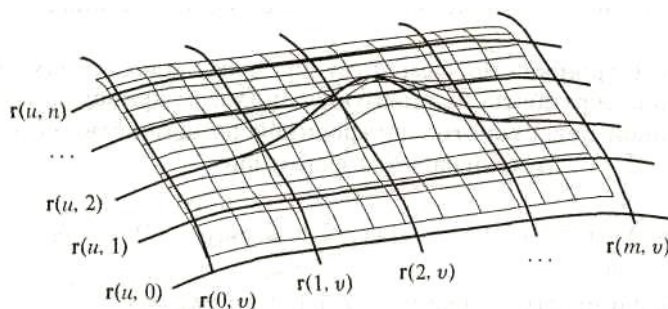


Рис. 2.12. Поверхня Гордона на сітці із кривих

На цій сітці можна побудувати складену поверхню, кожний осередок якої являє собою кубічну поверхню Кунса. Для цього криві перепозначимо відповідно до їх місця на поверхні: $\bar{r}(u, j) \equiv \bar{a}_j(t_j)$, $j = 0, 1, 2, \dots, n$, $\bar{r}(i, v) \equiv \bar{b}_i(\omega_i)$, $i = 0, 1, 2, \dots, m$. Точки перетинання кривих $\bar{r}(t_i, j)$ і $\bar{r}(i, t_j)$ позначимо через $\bar{r}(i, j)$, $j = 0, 1, 2, \dots, n$, $i = 0, 1, 2, \dots, m$. На кожній ділянці кривих уведемо параметризацію аналогічну (2.14) так, щоб між точками $\bar{r}(i, j)$ й $\bar{r}(k, l)$ параметр u змінювався від i до k , а параметр v змінювався від j до l . Радіус-вектор поверхні, що інтерполює всю сітку й задовольняє певні умови гладкості, описується співвідношенням

$$\bar{r}(u, v) = \sum_{j=0}^m L_j(v) \bar{r}(u, j) + \sum_{i=0}^n L_i(u) \bar{r}(i, v) - \sum_{i=0}^n \sum_{j=0}^m L_i(u) L_j(v) \bar{r}(i, j), \quad (2.26)$$

$$0 \leq u \leq m, \quad 0 \leq v \leq n,$$

де $L_j(v)$ й $L_i(u)$ – коефіцієнти Лагранжа (2.24). Ця формула аналогічна (2.23), тільки інтерполяція виконується по двох напрямках.

У формулах (2.23) і (2.26) замість коефіцієнтів Лагранжа можуть використатися довільні функції $a_k(\omega)$, які диференціюються, що задовольняють рівності

$$a_k(\omega) = \begin{cases} 1, & \text{если } \omega = k, \\ 0, & \text{если } \omega \neq k. \end{cases}$$

У якості функцій зсуву $a_k(\omega)$ можуть використатися кубічні сплайни. Поверхня (2.26) називається поверхнею Гордона. Гладкість такої поверхні залежить від гладкості функцій зсуву.

Для побудови поверхні на сітці, утвореної двома сімействами кривих, можуть бути використані поверхні Кунса (2.18) у вигляді її порцій. Тому що в цьому випадку утвориться поверхня, складена з декількох поверхонь, то вона називається складовою. Наявність у складі даних поверхні (2.18) похідних на стиках окремих порцій дозволяє одержати гладку складену поверхню. При використанні у якості кожній порції поверхні (2.18) складова поверхня буде гладкою на заданій сітці кривих. Її радіус-вектор описується вираженням

$$\begin{aligned} \bar{r}(u, v) = & [\alpha_0(\omega) \quad \alpha_1(\omega) \quad \beta_0(\omega) \quad \beta_1(\omega)] \cdot \begin{bmatrix} \bar{r}(u, j) \\ \bar{r}(u, j+1) \\ \bar{r}_v(u, j) \\ \bar{r}_v(u, j+1) \end{bmatrix} + \\ & + [\alpha_0(t) \quad \alpha_1(t) \quad \beta_0(t) \quad \beta_1(t)] \cdot \begin{bmatrix} \bar{r}(i, v) \\ \bar{r}(i+1, v) \\ \bar{r}_u(i, v) \\ \bar{r}_u(i+1, v) \end{bmatrix} - [\alpha_0(t) \quad \alpha_1(t) \quad \beta_0(t) \quad \beta_1(t)] \times \\ & \times \begin{bmatrix} \bar{r}(i, j) & \bar{r}(i, j+1) & \bar{r}_v(i, j) & \bar{r}_v(i, j+1) \\ \bar{r}(i+1, j) & \bar{r}(i+1, j+1) & \bar{r}_v(i+1, j) & \bar{r}_v(i+1, j+1) \\ \bar{r}_u(i, j) & \bar{r}_u(i, j+1) & \bar{r}_{uv}(i, j) & \bar{r}_{uv}(i, j+1) \\ \bar{r}_u(i+1, j) & \bar{r}_u(i+1, j+1) & \bar{r}_{uv}(i+1, j) & \bar{r}_{uv}(i+1, j+1) \end{bmatrix} \cdot \begin{bmatrix} \alpha_0(\omega) \\ \alpha_1(\omega) \\ \beta_0(\omega) \\ \beta_1(\omega) \end{bmatrix}, \quad (2.27) \\ & t = u - i, \quad \omega = v - j, \quad 0 \leq u \leq m, \quad 0 \leq v \leq n, \end{aligned}$$

де i дорівнює цілої частині параметра u , j дорівнює цілої частині параметра v , $m+1$ – число кривих v -напрямку, $n+1$ – число кривих u -напрямку. Кубічні функції зсуву визначаються формулами (2.19). Фактично співвідношення (2.18) описує одну порцію складеної поверхні, що перебуває між кривими $\bar{r}(u, j)$, $\bar{r}(u, j+1)$, $\bar{r}(i, v)$, $\bar{r}(i+1, v)$, $i=0,1,2,\dots,m-1$, $j=0,1,2,\dots,n-1$. Параметри t і ω є місцевими параметрами даної порції складеної поверхні, вони змінюються від

0 до 1. Якщо $m=1$ й $n=1$, то формула (2.26) описує одну порцію кубічної поверхні Кунса (2.18), побудованої на чотирьох кривих, показаних на рис. 2.8. Похідні на лініях сітки можуть бути обчислені по сусідніх лініях аналогічно тому, як по сусідніх точках обчислюються похідні складеного сплайна Ерміта [5] в опорних точках

$$\begin{aligned}\bar{r}_u(i, v) &= \frac{\bar{r}(i+1, v) - \bar{r}(i-1, v)}{2}, \\ \bar{r}_v(u, j) &= \frac{\bar{r}(u, j+1) - \bar{r}(u, j-1)}{2}\end{aligned}\quad (2.28)$$

для внутрішніх ліній і

$$\begin{aligned}\bar{r}_u(0, v) &= 2\bar{r}(1, v) - 2\bar{r}(0, v) - \bar{r}_u(1, v), \\ \bar{r}_u(m, v) &= 2\bar{r}(m, v) - 2\bar{r}(m-1, v) - \bar{r}_u(m-1, v), \\ \bar{r}_v(u, 0) &= 2\bar{r}(u, 1) - 2\bar{r}(u, 0) - \bar{r}_v(u, 1), \\ \bar{r}_v(u, n) &= 2\bar{r}(u, n) - 2\bar{r}(u, n-1) - \bar{r}_v(u, n-1)\end{aligned}\quad (2.29)$$

для крайніх ліній. Рівності (2.29) забезпечують рівність нулю третіх похідних на краях поверхні в трансверсальних до краю напрямках. Якщо всі криві якогонебудь сімейства є замкнутими, то поверхня (2.27) виходить замкнутою у відповідному напрямку. Помітимо, що поверхня (2.27) у точках перетинання ліній $\bar{r}(i, j)$ може мати сплющення. Припустимо, що в (2.26) сегменти кривих сітки між точками їхнього перетинання $\bar{r}(i, j)$ являють собою кубічні сплайни Ерміта [5], тобто визначаються залежностями:

$$\begin{aligned}\bar{r}(i, v) &= \alpha_0(\omega)\bar{r}(i, j) + \alpha_1(\omega)\bar{r}(i, j+1) + \beta_0(\omega)\bar{r}_v(i, j) + \beta_1(\omega)\bar{r}_v(i, j+1), \\ \bar{r}(u, j) &= \alpha_0(t)\bar{r}(i, j) + \alpha_1(t)\bar{r}(i+1, j) + \beta_0(t)\bar{r}_u(i, j) + \beta_1(t)\bar{r}_u(i+1, j), \\ \bar{r}_u(i, v) &= \alpha_0(\omega)\bar{r}_u(i, j) + \alpha_1(\omega)\bar{r}_u(i, j+1) + \beta_0(\omega)\bar{r}_{vu}(i, j) + \beta_1(\omega)\bar{r}_{vu}(i, j+1), \\ \bar{r}_v(u, j) &= \alpha_0(t)\bar{r}_v(i, j) + \alpha_1(t)\bar{r}_v(i+1, j) + \beta_0(t)\bar{r}_{uv}(i, j) + \beta_1(t)\bar{r}_{uv}(i+1, j),\end{aligned}$$

де i дорівнює цілої частині параметра u , j дорівнює цілої частині параметра v , $t = u - i$, $\omega = v - j$. Після підстановки цих залежностей в (2.26) одержимо три однакових матричних члени, два з них зі знаком плюс і один зі знаком мінус. У результаті співвідношення (2.26) прийме вид

$$\bar{r}(u, v) = [\alpha_0(t) \quad \alpha_1(t) \quad \beta_0(t) \quad \beta_1(t)] \times$$

$$\times \begin{bmatrix} \bar{r}(i, j) & \bar{r}(i, j+1) & \bar{r}_v(i, j) & \bar{r}_v(i, j+1) \\ \bar{r}(i+1, j) & \bar{r}(i+1, j+1) & \bar{r}_v(i+1, j) & \bar{r}_v(i+1, j+1) \\ \bar{r}_u(i, j) & \bar{r}_u(i, j+1) & \bar{r}_{uv}(i, j) & \bar{r}_{uv}(i, j+1) \\ \bar{r}_u(i+1, j) & \bar{r}_u(i+1, j+1) & \bar{r}_{uv}(i+1, j) & \bar{r}_{uv}(i+1, j+1) \end{bmatrix} \cdot \begin{bmatrix} \alpha_0(\omega) \\ \alpha_1(\omega) \\ \beta_0(\omega) \\ \beta_1(\omega) \end{bmatrix}, \quad (2.30)$$

$$t = u - i, \quad \omega = v - j, \quad 0 \leq u \leq m, \quad 0 \leq v \leq n,$$

де i дорівнює цілої частині параметра u , j дорівнює цілої частині параметра v . Поверхня (2.30) повністю визначається точками $\bar{r}(i, j)$ й значенням у них похідних поверхні $\bar{r}_u(i, j)$, $\bar{r}_v(i, j)$ і $\bar{r}_{uv}(i, j)$. Дану поверхню можна вважати складовою, тому що вона складається з порцій кубічних поверхонь Ерміта, які гладко стикуються між собою, тому що мають загальні похідні на границях. Поверхня (2.30) ще називають поверхнею тензорного добутку. Фактично дана поверхня будується на сітці точок із заданими похідними, а не на сітці кривих.

2.7. Поверхні Безьє

Криві Безьє характерні тим, що їхній радіус-вектор представлений у вигляді суми радіусів-векторів точок у просторі, помножених на деякі коефіцієнти, що є скалярними функціями параметра кривій [5]. Скалярними функціями є функції Бернштейна. Аналогічним образом будуються поверхні Безьє, радіус-вектор яких представляється у вигляді суми радіусів-векторів точок у просторі, помножених на коефіцієнти, що є функціями Бернштейна параметрів поверхні. Розглянемо поверхні Безьє докладніше.

Нехай є сукупність точок \bar{p}_{ij} , що утворюють сітку, тобто умовно розташованих у вигляді $n+1$ рядів по $m+1$ точок у кожному ряді. Індекси точки \bar{p}_{ij} означають, що дана точка розташована j -ю по рахунку в i -м ряді (перший індекс дорівнює номеру ряду, другий – номеру точки в ряді). Нумерувати ряди будемо від 0 до n , а точки в них будемо нумерувати від 0 до m . Поверхня Безьє, побудована на сукупності точок \bar{p}_{ij} , визначається радіусом-вектором

$$\bar{r}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_j^m(u) B_i^n(v) \bar{p}_{ij} = \sum_{i=0}^n \sum_{j=0}^m B_{ij}^{nm}(v, u) \bar{p}_{ij}, \quad (2.31)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1,$$

де $B_{ij}^{nm}(v, u) = B_i^n(v)B_j^m(u)$ – функції базису Бернштейна:

$$B_i^n(v) = \frac{n!}{(n-i)!i!} v^i (1-v)^{n-i} = C_n^i v^i (1-v)^{n-i}, \quad (2.32)$$

$$B_j^m(v) = \frac{m!}{(m-j)!j!} u^j (1-u)^{m-j} = C_m^j u^j (1-u)^{m-j}. \quad (2.33)$$

Варто відзначити, що функції $B_{ij}^{nm}(v, u)$ задовольняють рівності

$$\sum_{i=0}^n \sum_{j=0}^m B_{ij}^{nm}(v, u) = 1.$$

Будемо говорити, що уздовж u -напрямку поверхня має ступінь m , а уздовж v -напрямку поверхня має ступінь n .

Якщо представити, що на кожних чотирьох сусідніх точках \bar{p}_{ij} , \bar{p}_{ij+1} , \bar{p}_{i+1j} , \bar{p}_{i+1j+1} побудована білінійна поверхня, то ми одержимо багатогранник, що будемо називати характеристичним. Точки \bar{p}_{ij} будемо називати вершинами характеристичного багатогранника. Поверхня Безьє разом зі своїм характеристичним багатогранником показана на рис. 2.13. Тому що з коефіцієнтів Бернштейна тільки нульовий і останній приймають максимально можливі значення, то поверхня Безьє проходить тільки через кутові вершини сітки: \bar{p}_{00} , \bar{p}_{0m} , \bar{p}_{n0} , \bar{p}_{nm} . Перетини поверхні (2.31) по лініях $u = const$ або $v = const$ являють собою криві Безьє. Характеристичний багатогранник дає загальне представлення про поверхні Безьє, а сама поверхня як би згладжує кути цього багатогранника. Чим вище порядок поверхні (2.31), тим вона більш гладка. Порядок поверхні в u -напрямку жорстко пов'язаний з кількістю вершин у ряді, а порядок поверхні в v -напрямку жорстко пов'язаний з кількістю рядів вершин. Область зміни параметрів поверхні являє собою квадрат зі стороною, рівною 1.

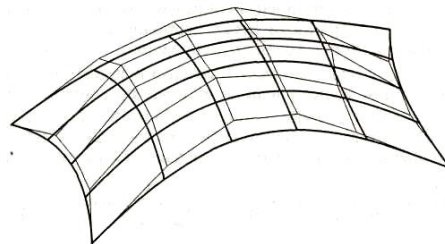


Рис. 2.13. Поверхня Безьє і її характеристичний багатогранник

Ступінь поверхні Безьє, а відповідно і її гладкість, жорстко пов'язані з кількістю вершин, по яких вона побудована. Це може створити деякі незручності при використанні поверхонь Безьє. Перебороти ці незручності можуть допомогти складені поверхні. Складена поверхня Безьє виходить у результаті стикування окремих поверхонь Безьє, що мають уздовж країв однакові ступені, що стикаються. У загальному випадку для того, щоб порції поверхні Безьє гладко стикувалися один з одним, потрібно щоб ребра, що прилягають до кутів порцій поверхонь, що стикуються, лежали в одній площині.

2.8. Раціональні поверхні

Поверхню Безьє можна узагальнити так, щоб вона могла описати необхідну частину поверхні другого порядку – еліпсоїда, гіперболоїда, параболоїда або їхніх окремих випадків.

Наділимо кожну вершину кривої (2.31) \bar{p}_{ij} вагою ω_{ij} й будемо описувати поверхню в однорідних координатах. Поверхні, характеристичні точки яких мають ваги, називаються раціональними. Сам радіус-вектор такої поверхні теж здобуває вагу. Вага радіуса-вектора обчислюється по точно таких же формулах, що і його координати, тому вагу прийнято вважати додатковою координатою точки. Розширений радіус-вектор раціональної поверхні Безьє описується формулою, по виду співпадаючою з (2.31), у якій звичайні вектори $\bar{r}(u, v)$ й \bar{p}_{ij} замінені відповідними розширеними векторами $\bar{R}(u, v)$ й \bar{P}_{ij} . Звичайний тривимірний радіус-вектор раціональної поверхні Безьє визначається співвідношенням

$$\bar{r}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m B_i^n(v) B_j^m(u) \omega_{ij} \bar{p}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m B_i^n(v) B_j^m(u) \omega_{ij}}, \quad 0 \leq u \leq 1, \quad 0 \leq v \leq 1. \quad (2.34)$$

У раціональної поверхні відіграють роль не абсолютні значення ваг вершин, а їхні відносні значення. Чим більше вага деякої вершини стосовно ваг інших вершин, тим ближче до цієї вершини проходить поверхня.

За допомогою поверхні (2.34) може бути побудована частина поверхні другого порядку або частина поверхні тора. Для цього найпростіше використати раціональну поверхню Безьє другого ступеня, побудовану по дев'яточ вершинах:

$$\bar{r}(u, v) = \frac{\begin{bmatrix} (1-v)^2 & 2v(1-v) & v^2 \end{bmatrix} \cdot \begin{bmatrix} \bar{p}_{00} & \omega_u \bar{p}_{01} & \bar{p}_{02} \\ \omega_v \bar{p}_{10} & \omega_u \omega_v \bar{p}_{11} & \omega_v \bar{p}_{12} \\ \bar{p}_{20} & \omega_u \bar{p}_{21} & \bar{p}_{22} \end{bmatrix} \cdot \begin{bmatrix} (1-u)^2 \\ 2u(1-u) \\ u^2 \end{bmatrix}}{\left((1-u)^2 + 2u(1-u)\omega_u + u^2 \right) \left((1-v)^2 + 2v(1-v)\omega_v + v^2 \right)}, \quad (2.35)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1.$$

В (2.35) ваги кутових вершин покладені рівними одиниці, а для інших вершин уведені ваги ω_u й ω_v , що мають індекси параметричних напрямків поверхні. Тип поверхні, одержуваної за допомогою (2.35), залежить від положення вершин і від ваг ω_u і ω_v . Так, наприклад, для того щоб одержати частину поверхні тора з кутами побудови α_u й α_v , потрібно, щоб вершини \bar{p}_{0i} , \bar{p}_{1i} , \bar{p}_{2i} і \bar{p}_{i0} , \bar{p}_{i1} , \bar{p}_{i2} , де $i=0,1,2$, розташовувалися друг щодо друга аналогічно тому, як показано на рис. 2.14, а ваги вершин мали значення $\omega_u = \cos\left(\frac{\alpha_u}{2}\right)$,

$$\omega_v = \cos\left(\frac{\alpha_v}{2}\right).$$

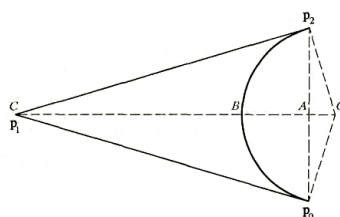


Рис. 2.14. Розташування вершин для побудови частини тора

За допомогою складеної раціональної поверхні Безьє може бути побудована повністю будь-яка поверхня другого порядку. Для цього потрібно використати кілька поверхонь (2.35), що гладко стикуються між собою, як було описано вище.

2.8.1. Поверхня округлення

Раціональною поверхнею можна описати поверхню округлення. Перетин поверхні округлення в одному з параметричних напрямків повинен являти

собою дугу окружності. Цю дугу ми опишемо за допомогою раціональній кривій Безьє, у якій вершини $\bar{p}_0, \bar{p}_1, \bar{p}_2$ і кут дуги α будемо вважати функціями одного з параметрів поверхні округлення. Радіус-вектор поверхні округлення можна представити у вигляді

$$\bar{r}(u, v) = \frac{(1-t)^2 \bar{p}_0(u) + 2t(1-t)\omega(u)\bar{p}_1(u) + t^2 \bar{p}_2(u)}{(1-t)^2 + 2t(1-t)\omega(u) + t^2}, \quad (2.36)$$

$$u_{\min} \leq u \leq u_{\max}, \quad 0 \leq t \leq 1,$$

де $\omega(u) = \cos\left(\frac{\alpha(u)}{2}\right)$ – вага лінії $\bar{p}_1(u)$, обчислена по куту дуги в даному місці.

Найпростіший варіант поверхні округлення (2.36), побудованої між двома площинами, наведений на рис. 2.15.

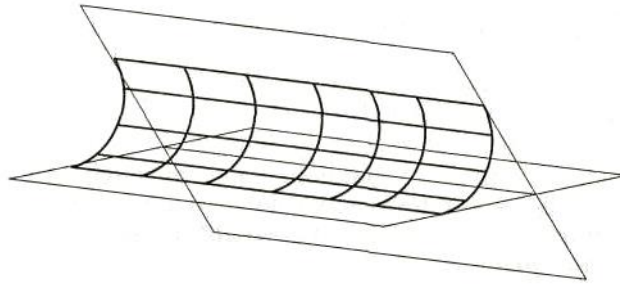


Рис. 2.15. Поверхня округлення - приклад раціональної поверхні

Будемо розглядати вираження (2.34) у більше широкому змісті, а саме, будемо вважати, що в ньому замість функцій B_{ij}^{mm} можуть стояти деякі довільні базисні функції, які позначимо через F_{ij} . У загальному виді радіус-вектор раціональної параметричної поверхні, побудованої на mm вершинах, визначається як

$$\bar{r}(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) \omega_{ij} \bar{p}_{ij}}{\sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) \omega_{ij}}, \quad u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \quad (2.37)$$

де \bar{p}_{ij} – радіуси-вектори j -ї вершини, розташованої в i -м ряді, характеристичного багатогранника поверхні, ω_{ij} – вага відповідної вершини, $F_{ij}(v, u)$ – функція j -ї вершини в i -м ряді, n – число рядів вершин, m – число вершин у кожному ряді. На відміну від (2.34) підсумовування у вираженні

(2.37) починається з 1, а не з 0. Це не має принципового значення, просто для поверхонь Безье зручно було нумерувати, починаючи з 0, а в інших випадках зручніше проводити нумерацію, починаючи з 1. Параметр u змінюється уздовж вершин, що знаходяться у деякому ряді, а параметр v змінюється уздовж рядів вершин, що мають однаковий номер у ряді.

У термінах однорідних координат розширений радіус-вектор раціональної поверхні визначається співвідношенням

$$\bar{R}(u, v) = \sum_{i=1}^n \sum_{j=1}^m F_{ij}(v, u) \bar{P}_{ij}, \quad u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \quad (2.38)$$

де $\bar{P}_{ij} = [\omega_{ij} \bar{P}_{ij} \quad \omega_{ij}]^T$ – розширені вектори вершин поверхні.

Радіус-вектор раціональної поверхні обчислюється як частка від розподілу двох функцій параметрів u і v , тому при обчисленні похідних раціональної поверхні праву частину (2.38) варто розглядати як складну функцію. Якщо умовно позначимо радіус-вектор раціональної поверхні як $\bar{r} = \frac{\omega \bar{r}}{\omega}$, то похідні радіуса-вектора раціональної поверхні визначаються як

$$\frac{\partial \bar{r}}{\partial u} = \frac{1}{\omega} \frac{\partial(\omega \bar{r})}{\partial u} - \frac{\omega \bar{r}}{\omega^2} \frac{\partial \omega}{\partial u}, \quad (2.39)$$

$$\frac{\partial \bar{r}}{\partial v} = \frac{1}{\omega} \frac{\partial(\omega \bar{r})}{\partial v} - \frac{\omega \bar{r}}{\omega^2} \frac{\partial \omega}{\partial v}, \quad (2.40)$$

$$\frac{\partial^2 \bar{r}}{\partial u^2} = \frac{1}{\omega} \frac{\partial^2(\omega \bar{r})}{\partial u^2} - \frac{2}{\omega^2} \frac{\partial(\omega \bar{r})}{\partial u} \frac{\partial \omega}{\partial u} - \frac{\omega \bar{r}}{\omega^2} \frac{\partial^2 \omega}{\partial u^2} + \frac{2\omega \bar{r}}{\omega^3} \left(\frac{\partial \omega}{\partial u} \right)^2, \quad (2.41)$$

$$\frac{\partial^2 \bar{r}}{\partial v^2} = \frac{1}{\omega} \frac{\partial^2(\omega \bar{r})}{\partial v^2} - \frac{2}{\omega^2} \frac{\partial(\omega \bar{r})}{\partial v} \frac{\partial \omega}{\partial v} - \frac{\omega \bar{r}}{\omega^2} \frac{\partial^2 \omega}{\partial v^2} + \frac{2\omega \bar{r}}{\omega^3} \left(\frac{\partial \omega}{\partial v} \right)^2, \quad (2.42)$$

$$\frac{\partial^2 \bar{r}}{\partial u \partial v} = \frac{1}{\omega} \frac{\partial^2(\omega \bar{r})}{\partial u \partial v} - \frac{1}{\omega^2} \frac{\partial(\omega \bar{r})}{\partial u} \frac{\partial \omega}{\partial v} - \frac{1}{\omega^2} \frac{\partial(\omega \bar{r})}{\partial v} \frac{\partial \omega}{\partial u} - \frac{\omega \bar{r}}{\omega^2} \frac{\partial^2 \omega}{\partial u \partial v} + \frac{2\omega \bar{r}}{\omega^3} \frac{\partial \omega}{\partial u} \frac{\partial \omega}{\partial v}. \quad (2.43)$$

Похідні більше високого порядку обчислюються аналогічно.

2.9. NURBS поверхні

У якості функцій $F_{ij}(v, u)$ у формулі (2.37) будемо використати добутки B -сплайнів, один із яких залежить від параметра u , а іншої залежить від параметра v . У результаті одержимо неоднорідну раціональну поверхню,

визначену B -сплайнами, що придбала назву NURBS поверхні (Non-Uniform Rational B-Spline surface). NURBS поверхня, як і поверхня Безьє, будується на сітці вершин – сукупності точок \bar{p}_{ij} , умовно розташованих у вигляді n рядів по m точок у кожному ряді. Індокси вершини \bar{p}_{ij} означають, що дана точка розташована j -ю по рахунку в i -м ряді (перший індекс дорівнює номеру ряду, другий – номеру точки в ряді). У цьому випадку нумерувати ряди зручніше від 1 до n , а точки в них – від 1 до m . Нехай вага вершини \bar{p}_{ij} дорівнює ω_{ij} й нехай B -сплайни $N_{jk}(u)$ u -напрямку мають порядок k , а B -сплайни $N_{iq}(v)$ v -напрямку мають порядок q . Радіус-вектор NURBS поверхні визначається рівністю

$$\bar{r}(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) \omega_{ij} \bar{p}_{ij}}{\sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) \omega_{ij}} = \frac{\sum_{i=1}^n \sum_{j=1}^m N_{ijqk}(v, u) \omega_{ij} \bar{p}_{ij}}{\sum_{i=1}^n \sum_{j=1}^m N_{ijqk}(v, u) \omega_{ij}}, \quad (2.44)$$

$$u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max},$$

де $N_{ijqk}(v, u) = N_{iq}(v) N_{jk}(u)$ – базисні функції, n – число рядів вершин, m – число вершин у кожному ряді. Похідні радіуса-вектора NURBS поверхні визначаються рівностями (2.39)-(2.43). Кожний з B -сплайнів k -го порядку $N_{jk}(u)$ побудований на послідовності з $k+1$ вузлів $u_j, u_{j+1}, u_{j+2}, \dots, u_{j+k}$; усього для побудови сукупності m B -сплайнів k -го порядку потрібно $m+k$ вузлів u_j у випадку незамкнутої кривої й $m+2k$ вузлів у випадку замкнутої кривої. Кожний з B -сплайнів q -го порядку $N_{iq}(v)$ побудований на послідовності з $q+1$ вузлів $v_i, v_{i+1}, v_{i+2}, \dots, v_{i+q}$; усього для побудови сукупності B -сплайнів q -го порядку потрібно $n+q$ вузлів v_i у випадку незамкнутої кривої й $n+2q$ вузлів у випадку замкнутої кривої. Як витікає з (2.44), NURBS поверхня має дві послідовності вузлів, одну – в u -напрямку, а іншу – в v -напрямку. Значення вузлів повинні утворювати неубутні послідовності. Будемо називати їх u -послідовністю вузлів і v -послідовністю вузлів. Пронумеруємо вузли кожної послідовності, почавши нумерацію з 1

Для незамкнутої NURBS поверхні використовуються наступні послідовності вузлів. Перші k вузлів u -послідовності мають значення, рівні нулю: $u_1 = u_2 = \dots = u_k = 0$; наступних $m-k$ вузлів приймають цілочисельні значення від 1 до $m-k$: $t_{k+j} = j$ ($j=1,2,\dots,m-k$); k вузлів, що залишилися, приймають значення $m-k+1$: $t_{m+1} = t_{m+2} = \dots = t_{m+k} = m-k+1$. Перші q вузлів v -послідовності мають значення, рівні нулю: $v_1 = v_2 = \dots = v_q = 0$; наступних $n-q$ вузлів приймають цілочисельні значення від 1 до $n-q$: $t_{q+i} = i$ ($i=1,2,\dots,n-q$); q вузлів, що залишилися, приймають значення $n-q+1$: $t_{n+1} = t_{n+2} = \dots = t_{n+q} = n-q+1$.

Для замкнутої NURBS поверхні використовуються рівномірні послідовності вузлів з одиничним кроком: $u_j = j-m$, $j=1,2,\dots,m+2k$, $v_i = i-n$, $i=1,2,\dots,n+2q$.

Наведені приклади вузлових послідовностей не є єдино можливими. Як і для будь-якої поверхні, внутрішня параметризація NURBS поверхні може бути довільною.

Область зміни параметрів NURBS поверхні являє собою прямокутник: $u_k \leq u \leq u_{m+k}$, $v_q \leq v \leq u_{n+q}$. Для зазначених послідовностей вузлів параметри NURBS поверхні змінюються в межах: $0 \leq u \leq (m-k+1)$, $0 \leq v \leq (n-q+1)$ для незамкнутої поверхні й $0 \leq u \leq m$, $0 \leq v \leq n$ для замкнутої поверхні.

За значенням параметра u з умови $u_j \leq u < u_{j+1}$ визначимо номер j єдиного відмінного від нуля B -сплайна першого порядку

$$M_{j1}(u) = \frac{1}{u_{j+1} - u_j}.$$

Далі, використовуючи рекурентне співвідношення

$$M_{rk}(u) = \frac{(u_{r+k} - u)M_{r+1k-1}(u) + (u - u_r)M_{rk-1}(u)}{u_{r+k} - u_r}, \quad r = j-k+1, j-k+2, \dots, j,$$

послідовно обчислимо всі відмінні від нуля при даному u B -сплайни до k -го порядку включно: $M_{j-k+1k}(u)$, $M_{j-k+2k}(u)$, ..., $M_{jk}(u)$... Далі B -сплайни k -го порядку нормуємо відповідно до співвідношення

$$N_{rk}(u) = (u_{r+k} - u_r)M_{rk}(u), \quad r = j-k+1, j-k+2, \dots, j.$$

Аналогічно за значенням параметра v з умови $v_i \leq v < v_{i+1}$ визначимо номер i відмінного від нуля B -сплайна першого порядку

$$M_{i1}(v) = \frac{1}{v_{i+1} - v_i}.$$

Використовуючи рекурентне співвідношення

$$M_{rq}(v) = \frac{(v_{r+q} - v)M_{r+1q-1}(v) + (v - v_r)M_{rq-1}(v)}{v_{r+q} - v_r}, \quad r = i - q + 1, i - q + 2, \dots, i,$$

послідовно обчислимо всі відмінні від нуля при даному v B -сплайни до q -го порядку включно: $M_{i-q+1q}(v)$, $M_{i-q+2q}(v)$, ..., $M_{iq}(v)$ і потім нормуємо їх

$$N_{rq}(v) = (v_{r+q} - v_r)M_{rq}(v), \quad r = i - q + 1, i - q + 2, \dots, i.$$

Нагадаємо, що порядок NURBS поверхні в тім або іншому параметричному напрямку дорівнює порядку відповідних B -сплайнів, а порядок B -сплайна на одиницю більше ступеня поліномів, з яких він побудований.

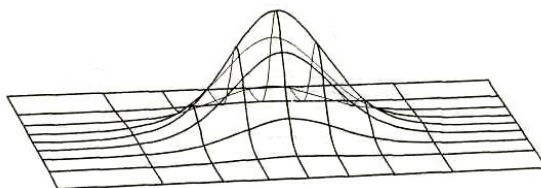


Рис. 2.16. NURBS поверхня

На рис. 2.16 показана NURBS поверхня 4-го порядку по обох параметричних напрямках, побудована на сітці, що складається з 7×7 вершин. Всі вершини, крім центральної, лежать в одній площині. Ваги всіх вершин рівні 1.

Якщо на кожних чотирьох сусідніх характеристичних точках побудувати білінійні поверхні, то ми одержимо деякий багатогранник, що називається характеристичним. Характеристичний багатогранник поверхні, наведеної на рис. 2.16, показаний на рис. 2.17.

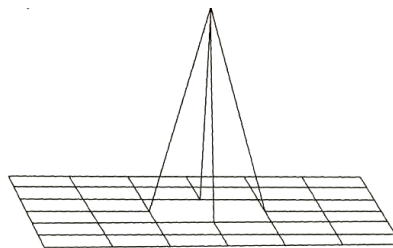


Рис. 2.17. Характеристичний багатогранник NURBS поверхні

У термінах однорідних координат розширений радіус-вектор NURBS поверхні визначається вираженням

$$\bar{R}(u, v) = \sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) \bar{P}_{ij}, \quad u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \quad (2.45)$$

де $\bar{P} = [\omega_{ij} \bar{p}_{ij} \quad \omega_{ij}]^T$ – розширені вектори вершин поверхні.

Коли всі вершини NURBS поверхні мають рівні ваги, то формула (2.44) у силу властивості B -сплайнів прийме вид

$$\bar{r}(u, v) = \sum_{i=1}^n \sum_{j=1}^m N_{iq}(v) N_{jk}(u) \bar{p}_{ij}, \quad (2.46)$$

$$u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}.$$

Ця поверхня називається B -сплайн поверхнею. Вона також може бути використана в проектуванні, тому що має певну гладкість, і для обчислення її похідних не потрібно користуватися формулами (2.39)-(2.43).

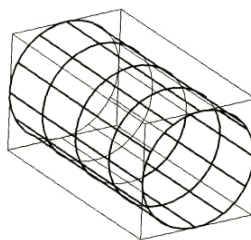


Рис. 2.18. Циліндрична NURBS поверхня

Якщо побудувати NURBS поверхню (2.46) на базі B -сплайнів другого порядку, то вона збігається зі своїм характеристичним багатогранником і являє собою сукупність білінійних поверхонь.

Якщо $n = q$ й $m = k$, тобто якщо число рядів вершин n дорівнює порядку q , B -сплайнів v -напрямку, число вершин у ряді m дорівнює порядку k B -сплайнів u -напрямку, вузли u -послідовності мають значення $u_1 = u_2 = \dots = u_m = 0$, $u_{m+1} = u_{m+2} = \dots = u_{2m} = 1$, вузли v -послідовності мають значення $v_1 = v_2 = \dots = v_n = 0$,

$v_{n+1} = v_{n+2} = \dots = v_{2n} = 1$, то NURBS поверхня (2.44) збігається з раціональною поверхнею Безьє (2.34). Це витікає з того, що функції Бернштейна є окремими випадками B -сплайнів. Область зміни параметрів NURBS поверхні в цьому випадку являє собою квадрат зі стороною, рівною 1.

За допомогою поверхні (2.44) може бути побудована будь-яка поверхня другого порядку. На рис. 2.18 наведена циліндрична NURBS поверхня і її характеристичний багатогранник. На рис. 2.19 наведена тороїдальна NURBS поверхня і її характеристичний багатогранник. Обидві поверхні мають кратні вузли.

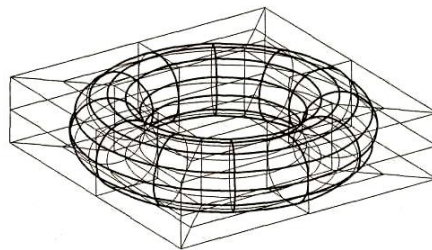


Рис. 2.19. Тороїдальна NURBS поверхня

По своєму зовнішньому вигляду NURBS поверхні й поверхні Безьє схожі. Але на відміну від останніх порядок NURBS поверхні не зв'язаний жорстко з кількістю вершин і надає можливість будувати поверхні невисокого порядку (3-го або 4-го) на великій кількості вершин. Ця можливість надає поверхні більшу гнучкість, тому що поверхні Безьє, побудовані на великій кількості вершин, мають високий порядок і є занадто гладкими.

NURBS поверхня є узагальненням більшості розглянутих вище поверхонь: багатогранної білінійної поверхні, поверхні Безьє, поверхонь другого порядку, раціональних поверхонь.

Кожна лінія $u = const$ або $v = const$ на поверхні (2.44) є NURBS кривою. Але можна побудувати раціональні неоднорідні поверхні на основі B -сплайнів, які необов'язково давали б NURBS криві в якості всіх своїх параметричних u -ліній або v -ліній. Наприклад, можна одержати поверхню, якщо замість вершин характеристичної ламаної підставити криві $\bar{p}_i = \bar{p}_i(u)$, $i = 1, 2, \dots, n$, що володіють

вагами $\omega_i = \omega_i(u)$, і мають однакові параметричні довжини $u_{\min} \leq u \leq u_{\max}$. У результаті, одержимо поверхню, описувану вираженням

$$\bar{r}(u, v) = \frac{\sum_{i=1}^n N_{im}(v) \omega_i(u) \bar{p}_i(u)}{\sum_{i=1}^n N_{im}(v) \omega_i(u)}, \quad (2.47)$$

$$u_{\min} \leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}.$$

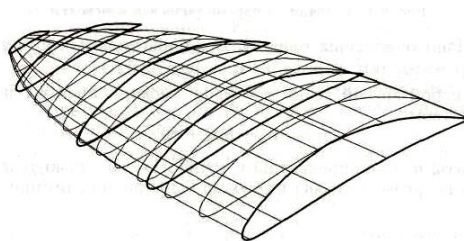


Рис. 2.20. NURBS поверхні на сімействі кривих

Тільки параметричні лінії $u = const$ на поверхні (2.47) є NURBS кривими. Таку поверхню можна назвати плазовою NURBS поверхнею. На рис. 2.20. показана плазова NURBS поверхня й п'ять кривих, по яких вона побудована. Окремим випадком цієї поверхні є поверхня округлення (2.36).

2.10. Поверхні, що базуються на поверхнях

На базі поверхні можна побудувати іншу поверхню. Розглянемо еквідистантну, паралельну й продовжену поверхні. Поверхню, на основі якої будується нова поверхня, будемо називати базовою поверхнею. Базова поверхня буде лежати в структурі даних розглянутих нижче поверхонь.

2.10.1. Еквідистантна поверхня

Для будь-якої поверхні може бути побудована еквідистантна поверхня. Кожна точка еквідистантної поверхні відстоїть від відповідної точки базової поверхні на заданій відстані h по нормалі до неї. Радіус-вектор еквідистантної поверхні описується співвідношенням

$$\bar{r}(u, v) = \bar{b}(u, v) + h\bar{m}(u, v), \quad u, v \in \Omega, \quad (2.48)$$

де $\bar{b}(u, v)$, $u, v \in \Omega$ – базова поверхня, Ω – область визначення параметрів базової поверхні, $\bar{m}(u, v)$ – нормаль до базової поверхні.

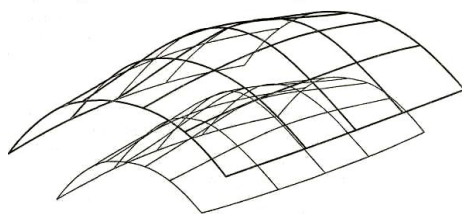


Рис. 2.21. Еквідистантна поверхня

Нормаль має одиничну довжину й напрямок вектора $\bar{b}_1 \times \bar{b}_2$. Область зміни параметрів еквідистантної поверхні збігається з областю зміни параметрів базової поверхні. Еквідистантна й базова поверхні показані на рис. 2.21. Похідні $\bar{r}_1(u, v)$ й $\bar{r}_2(u, v)$ еквідистантної поверхні по параметрах u і v описуються векторами

$$\bar{r}_1(u, v) = \bar{b}_1(u, v) + h\bar{m}_1(u, v), \quad (2.49)$$

$$\bar{r}_2(u, v) = \bar{b}_2(u, v) + h\bar{m}_2(u, v), \quad (2.50)$$

де похідні нормалі $\bar{m}_1(u, v)$ й $\bar{m}_2(u, v)$ по параметрах визначаються за допомогою дериваційних формул Вейнгартена [5].

У загальному випадку можна побудувати еквідистантну поверхню, у якої відстань $h = h(u, v)$ залежить від параметрів поверхні.

2.10.2. Посилальна поверхня

Посилальна поверхня являє собою поверхню, кожна точка якої отримана шляхом деякого перетворення відповідної точки базової поверхні. Посилальна поверхня описується радіусом-вектором

$$\bar{r}(u, v) = M_t \cdot \bar{b}(u, v), \quad u, v \in \Omega, \quad (2.51)$$

де $\bar{b}(u, v)$, $u, v \in \Omega$ – базова поверхня, Ω – область визначення параметрів базової поверхні, M_t – розширена матриця перетворення базової поверхні. Область зміни параметра посилальної поверхні збігається з областю зміни параметра базової поверхні. Посилальна поверхня може бути використана при проєктивних перетвореннях.

2.10.3. Репараметризована поверхня

Для будь-якої поверхні може бути зроблена лінійна репараметризація. Нехай область визначення параметрів поверхні $\bar{r}(u, v)$ описується нерівностями

$u_{\min} \leq u \leq u_{\max}$, $v_{\min} \leq v \leq v_{\max}$. Нехай потрібно, щоб кожний з параметрів змінювався в межах від x до y . Будемо вважати, що u , v залежать від нових параметрів p і q :

$$u(p) = u_{\min} \frac{y-p}{y-x} + u_{\max} \frac{p-x}{y-x}, \quad (2.52)$$

$$v(p) = v_{\min} \frac{y-q}{y-x} + v_{\max} \frac{q-x}{y-x}. \quad (2.53)$$

Тоді радіус-вектор поверхні буде описуватися функцією

$$\bar{r}(p, q) = \bar{r}(u(p), v(q)), \quad x \leq p \leq y, \quad x \leq q \leq y. \quad (2.54)$$

Після переходу до нових параметрів сама поверхня не зміниться, але область визначення її параметрів буде являти собою квадрат зі стороною $y-x$. Зміну області визначення параметрів можна вважати одним зі способів побудови поверхні на базі іншої поверхні.

2.10.4. Продовжена поверхня

Будь-яка поверхня може бути продовжена на задану параметричну відстань. Нехай потрібно продовжити поверхню $\bar{b}(u, v)$, $u_1 \leq u \leq u_2$, $v_1 \leq v \leq v_2$ шляхом розширення області визначення параметрів до $a_u + u_1 \leq u \leq u_2 + b_u$, $a_v + v_1 \leq v \leq v_2 + b_v$. Якщо $a_u \leq 0$, $a_v \leq 0$, $b_u \geq 0$, $b_v \geq 0$, то поверхня буде продовжена за свої межі. Якщо $a_u > 0$, $a_v > 0$, $b_u < 0$, $b_v < 0$, то поверхня буде усічена. Якщо поверхня замкнута по якому-небудь із параметрів, то при виході замкнутого параметра за кордон області визначення виконаємо його циклічне перерахування. Якщо поверхня $\bar{b}(u, v)$ замкнута по параметру u , тоді

$$\bar{r}(u, v) = \begin{cases} \bar{b}(u + u_\Delta, v), & \text{если } u < u_1, v_1 \leq v \leq v_2, \\ \bar{b}(u, v), & \text{если } u_1 \leq u \leq u_2, v_1 \leq v \leq v_2, \\ \bar{b}(u - u_\Delta, v), & \text{если } u > u_2, v_1 \leq v \leq v_2, \\ \bar{b}(u, v_1) + (v - v_1)\bar{b}_2(u, v_1), & \text{если } u_1 \leq u \leq u_2, v < v_1, \\ \bar{b}(u, v_2) + (v - v_2)\bar{b}_2(u, v_2), & \text{если } u_1 \leq u \leq u_2, v > v_2, \end{cases} \quad (2.55)$$

де $u_\Delta = u_2 - u_1$. Якщо поверхня $\bar{b}(u, v)$ замкнута по параметру v , тоді

$$\bar{r}(u, v) = \begin{cases} \bar{b}(u, v + v_\Delta), & \text{если } u_1 \leq u \leq u_2, v < v_1, \\ \bar{b}(u, v), & \text{если } u_1 \leq u \leq u_2, v_1 \leq v \leq v_2, \\ \bar{b}(u, v - v_\Delta), & \text{если } u_1 \leq u \leq u_2, v > v_2, \\ \bar{b}(u_1, v) + (u - u_1)\bar{b}_1(u_1, v), & \text{если } u < u_1, v_1 \leq v \leq v_2, \\ \bar{b}(u_2, v) + (u - u_2)\bar{b}_1(u_2, v), & \text{если } u > u_2, v_1 \leq v \leq v_2, \end{cases} \quad (2.56)$$

де $v_\Delta = v_2 - v_1$.

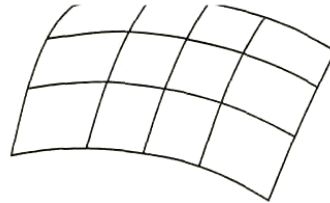


Рис. 2.22. Вихідна поверхня

Якщо ж поверхня не є замкнутою, а один з параметрів або обоє вийшли за кордон області визначення, то продовжимо поверхню по дотичній на найближчій границі, і обчислимо по продовженій поверхні необхідні геометричні характеристики. Радіус-вектор продовженої поверхні обчислимо по формулі

$$\bar{r}(u, v) = \begin{cases} \bar{b}(u, v), & \text{если } u_1 \leq u \leq u_2, v_1 \leq v \leq v_2, \\ \bar{b}(u_1, v) + (u - u_1)\bar{b}_1(u_1, v), & \text{если } u < u_1, v_1 \leq v \leq v_2, \\ \bar{b}(u_2, v) + (u - u_2)\bar{b}_1(u_2, v), & \text{если } u > u_2, v_1 \leq v \leq v_2, \\ \bar{b}(u, v_1) + (v - v_1)\bar{b}_2(u, v_1), & \text{если } u_1 \leq u \leq u_2, v < v_1, \\ \bar{b}(u, v_2) + (v - v_2)\bar{b}_2(u, v_2), & \text{если } u_1 \leq u \leq u_2, v > v_2, \\ \bar{b}(u_1, v_1) + (u - u_1)\bar{b}_1(u_1, v_1) + (v - v_1)\bar{b}_2(u_1, v_1) + \\ + (u - u_1)(v - v_1)\bar{b}_{12}(u_1, v_1), & \text{если } u < u_1, v < v_1, \\ \bar{b}(u_2, v_1) + (u - u_2)\bar{b}_1(u_2, v_1) + (v - v_1)\bar{b}_2(u_2, v_1) + \\ + (u - u_2)(v - v_1)\bar{b}_{12}(u_2, v_1), & \text{если } u > u_2, v < v_1, \\ \bar{b}(u_1, v_2) + (u - u_1)\bar{b}_1(u_1, v_2) + (v - v_2)\bar{b}_2(u_1, v_2) + \\ + (u - u_1)(v - v_2)\bar{b}_{12}(u_1, v_2), & \text{если } u < u_1, v > v_2, \\ \bar{b}(u_2, v_2) + (u - u_2)\bar{b}_1(u_2, v_2) + (v - v_2)\bar{b}_2(u_2, v_2) + \\ + (u - u_2)(v - v_2)\bar{b}_{12}(u_2, v_2), & \text{если } u > u_2, v > v_2, \end{cases} \quad (2.57)$$

де $\bar{b}_1(u,v)$ й $\bar{b}_2(u,v)$ – частинні похідні радіуса-вектора $\bar{b}(u,v)$ по параметрах u і v , відповідно, $\bar{b}_{12}(u,v)$ – змішана похідна радіуса-вектора $\bar{b}(u,v)$ по параметрах u і v . Диференціюючи формули (2.55)-(2.57), одержимо похідні радіуса-вектора продовженої поверхні. На рис. 2.22 наведена поверхня, на основі якої побудована продовжена поверхня. Продовжена поверхня наведена на рис. 2.23.

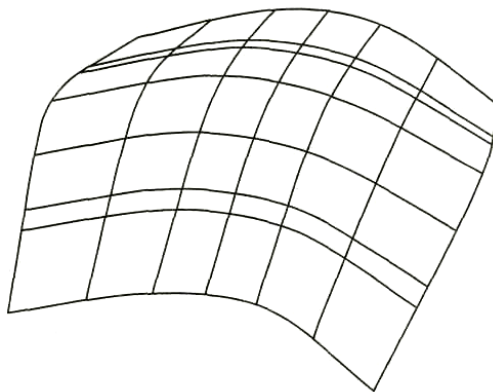


Рис. 2.23. Продовжена поверхня

Формули (2.55)-(2.57) можуть бути використані для обчислення радіуса-вектора поверхні і його похідних при виході параметрів поверхні за область їхнього визначення.

2.10.5. Загальне правило

Всі поверхні, у структурі даних яких лежить інша поверхня, не повинні допускати багаторазового спадкування свого ж типу. Наприклад, як базова поверхня для еквідистантної поверхні не повинна бути використана інша еквідистантна поверхня, а повинна бути використана базова поверхня останньої з відповідним перерахуванням довжини еквідистанти. Тобто, якщо потрібно побудувати еквідистантну поверхню на відстані h_2 від іншої еквідистантної поверхні $\bar{r}_1(u,v) = \bar{b}(u,v) + h_1 \bar{m}(t)$, то як базова поверхня повинна бути використана поверхня $\bar{b}(u,v)$, а довжина еквідистанти повинна дорівнювати $h = h_2 + h_1$. Якщо потрібно побудувати посиляльну поверхню з матрицею перетворення M_t на базі іншої посиляльної поверхні $\bar{r}_1(u,v) = N_t \bar{b}(u,v)$, то як базова поверхня повинна бути використана поверхня $\bar{b}(u,v)$, а матрицю

перетворення одержимо як добуток матриць $M_i \cdot N_i$. Аналогічні правила повинні діяти й для інших базованих поверхонь.

2.11. Обмежені контурами поверхні

Розглянемо поверхні, які виділяються серед всіх інших поверхонь і створюються «над ними» (на їхній основі).

Всі наведені вище поверхні мають «природні» прямокутні або трикутні області визначення параметрів. Прямокутні або трикутні області зручні у використанні через їхню простоту. Прямокутна область параметрів визначається нерівностями $u_{\min} \leq u \leq u_{\max}$, $v_{\min} \leq v \leq v_{\max}$, і для будь-якої пари параметрів можна відразу сказати, чи лежить вона поза областю, усередині області або на границі області. Розглянуті вище поверхні дають широкі можливості для моделювання. Необмежені можливості для моделювання ми одержимо, якщо границі поверхонь зможуть мати довільний вигляд. Це можна зробити за допомогою двомірних кривих ліній $\bar{r}_i(t_i) = [u(t_i) \quad v(t_i)]^T$, $t_{i\min} \leq t_i \leq t_{i\max}$, $i = 1, 2, \dots, n$, на поверхні, які описуються в просторі параметрів u і v поверхні. Ці двомірні криві зібрані в контури й розділяють використовувану частину поверхні від інших частин.

Така поверхня з довільною границею являє собою найбільш загальний тип поверхні. Будемо називати її обмеженої контурами поверхнею. Обмежену контурами поверхню можна побудувати на основі будь-якої поверхні із прямокутною або трикутною областю шляхом зміни області визначення її параметрів. Саме ця поверхня лежить в основі моделювання тіл. Її параметрам дозволяється приймати значення тільки усередині області, обмеженої заданими на ній контурами. Границі поверхні являють собою відображення в тривимірний простір границь двомірної зв'язної області.

Вихідну поверхню із прямокутною або трикутною областю визначення параметрів будемо називати базовою поверхнею. Нехай $\bar{b}(u, v)$ є радіус-вектор її точок. Радіус-вектор обмеженої контурами поверхні описується тим же

вираженням, що й базова поверхня, але має іншу область визначення параметрів

$$\bar{r}(u, v) = \bar{b}(u, v), \quad u, v \in \Omega, \quad (2.58)$$

де Ω , – область визначення параметрів, що являє собою обмежену замкнутими кривими частину площини, з кожної точки якої можна потрапити в будь-яку іншу її частину, не перетинаючи границь. Область визначення параметрів визначається двомірними контурами (замкнутими складовими кривими). Двомірний контур $\bar{c}(t) = [u(t) \ v(t)]^T$, $t_{\min} \leq t \leq t_{\max}$, складається з декількох двомірних кривих, що стикуються між собою. Радіус-вектор контуру визначається деякою функцією $\bar{c}(t)$. Таких контурів може бути кілька. Сукупність контурів обмежує двомірну область параметрів і надає границям поверхні задану форму.

Область визначення параметрів обмеженої контурами поверхні являє собою зв'язну частину двомірного простору й описується набором непересічних двомірних контурів, один із яких є зовнішнім, а інші лежать усередині нього і є внутрішніми. Внутрішні контури не можуть бути вкладені друг у друга. Для того щоб легше було визначити, чи належить деяка двомірна точка $\bar{p} = [u \ v]^T$ області визначення параметрів поверхні, контури орієнтують певним чином. Нехай зовнішній контур орієнтований так, що обхід по ньому в позитивному напрямку здійснювався проти годинникової стрілки, якщо дивитися на поверхню назустріч її нормалі. Внутрішні контури повинні бути орієнтовані в протилежному напрямку.

2.11.1. Приналежність точки області визначення параметрів

Нехай потрібно відповісти на запитання, чи належить двомірна точка $\bar{p} = [u \ v]^T$ простору параметрів області визначення поверхні – класифікувати точку щодо області. Для областей прямокутної форми відповісти на нього просто – для цього потрібно перевірити, чи задовольняють параметри u й v нерівностям $u_{\min} \leq u \leq u_{\max}$, $v_{\min} \leq v \leq v_{\max}$. Для обмеженої контурами поверхні, це зробити трохи складніше.

Нехай потрібно класифікувати точку $\bar{p} = [u \ v]^T$ в параметричній області поверхні, обмеженої трьома контурами: одним зовнішнім і двома внутрішніми, як показано на рис. 2.24.

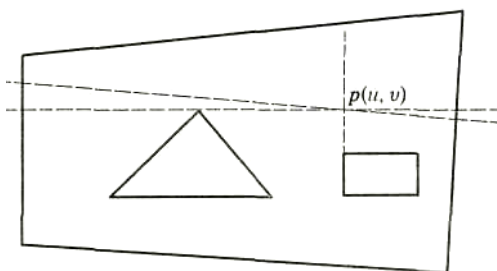


Рис. 2.24. Визначення положення точки щодо області

У загальному випадку для того щоб установити приналежність точки $\bar{p} = [u \ v]^T$ деякої двомірної області, можна знайти всі точки перетинання будь-якої двомірної прямої лінії (простіше горизонтальної або вертикальної), що проходить через цю точку, із границями області. Таких точок перетинання повинне бути парне число – кожній точці входу прямій в область повинна відповідати точка виходу з області. Але в деяких випадках, як показано на рис. 2.24, їх може бути непарне число через торкання або проходження прямої через кут граничного контуру. Якщо точок перетинання отримане непарне число, то можна перетнути границю області іншою двомірною прямою, що проходить через точку, що перевіряється, і складає невеликий кут із уже побудованою прямою. Далі легко визначити приналежність розглянутої точки заданої області. Якщо точка $\bar{p} = [u \ v]^T$ перебуває між точкою входу прямій в область і відповідної їй точкою виходу з області, то розглянута точка належить даній області. У протилежному випадку або у випадку відсутності точок перетинання прямій із границею області розглянута точка лежить поза даною областю. Якщо відомо, що зовнішній граничний контур області параметрів поверхні орієнтований проти годинникової стрілки, а внутрішні контури орієнтовані по годинникової стрілці, якщо дивитися на поверхню назустріч її нормалі, то класифікувати точку $\bar{p} = [u \ v]^T$ в параметричній області поверхні можна в такий спосіб.

Знайдемо найближчу до точки \bar{p} точку \bar{p}_0 на границі області. Побудуємо двомірний вектор \bar{w} зі знайденої точки \bar{p}_0 в точку \bar{p} .

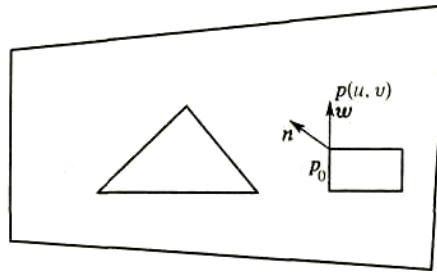


Рис. 2.25. Визначення положення точки щодо області визначення параметрів поверхні. Обчислимо нормаль \bar{n} до граничного контуру в точці \bar{p}_0 . Нормаль \bar{n} у правій системі координат завжди спрямована вліво, якщо дивитися уздовж напрямки контуру. Якщо точка \bar{p}_0 є точкою зламу граничного контуру, то як нормаль візьмемо середню нормаль до границі в цій точці. Якщо $\bar{n} \cdot \bar{w} \geq 0$, то точка \bar{p} належить області визначення поверхні. У протилежному випадку - немає. Це відображено на рис. 2.25.

2.11.2. Загальне правило

Обмежені контурами поверхні не повинні допускати багаторазового спадкування свого ж типу. Як базова поверхня для поверхні, обмеженої контурами, не повинна використатися інша обмежена контурами поверхня, а повинна бути взята базова поверхня останньої. Обмежені контурами поверхні повинні створюватися в останню чергу. Наприклад, якщо потрібно побудувати еквідистантну поверхню на базі обмеженої контурами поверхні, то спочатку повинна бути побудована еквідистантна поверхня на основі базової поверхні $\bar{b}(u, v)$, а потім на базі еквідистантної повинна бути створена обмежена контурами поверхня.

2.12. Способи побудови поверхонь

На рис. 2.26 наведені способи побудови поверхонь. Тут відображено лише найпоширеніші способи. В авіабудуванні широко використовуються різні сполучення поверхонь. Крім того, можна побудувати поверхні з характерними рисами – фічерсні поверхні (feature surface). Способи їхньої побудови можуть бути різними й залежать від специфіки області застосування.

Рис. 2.26 відображує процес побудови поверхонь при читанні його зліва на право. Початково поверхні можуть бути побудовані або по місцевій системі координат і скалярних параметрів, або по характерних точках, або на основі ліній. Далі ці поверхні можуть бути продовжені або усічені, на їхній основі можуть бути отримані еквідистантні поверхні. Всі згадані поверхні мають прямокутну або трикутну область визначення параметрів. На базі поверхонь із прямокутною або трикутною областю визначення параметрів можна побудувати обмежені контурами поверхні довільної форми.

Всі поверхні є рівноправними стосовно виконуваних ними функціям і виконуваним над ними операціям. Крім загальних функцій довільного геометричного об'єкта поверхні повинні виконувати функції обчислення радіуса-вектора поверхні, частинних похідних радіуса-вектора по параметрах, нормалі поверхні і її частинних похідних, коефіцієнтів перших і других квадратичних форм поверхні, функції надання інформації про області визначення параметрів поверхні (замкнутості, граничних контурах), функції визначення приналежності заданих параметрів області визначення поверхні, функції зміни положення й орієнтації в просторі, функції модифікації форми поверхні.

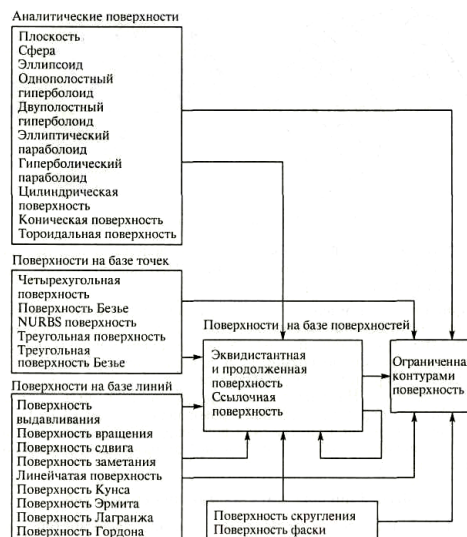


Рис. 2.26. Способи побудови поверхонь

Математична модель поверхні повинна бути доповнена функціями, що забезпечують взаємодію поверхні з іншими геометричними об'єктами й виконання над ними різних операцій.

Моделювання розглянутих поверхонь можливо, лише при наявності специфічної інформації про її точки, криві або деякі базові поверхні із яких їх можна одержати. Однак дуже часто трапляються випадки, коли ці дані одержати неможливо або непотрібно.

РОЗДІЛ 3

ГЕНЕРАЦІЯ ЛАНДШАФТІВ

Існує кілька основних принципів представлення даних для зберігання інформації про ландшафти:

- використання регулярної (рівномірної) сітки висот (або карти висот).
- використання іррегулярної (нерівномірної) сітки вершин і зв'язків, їх з'єднуючих (тобто зберігання простий триангулізованої карти).
- зберігання карти ландшафту, але не конкретних висот, а інформації про використаний блок. У цьому випадку створюється деяка кількість заздалегідь побудованих сегментів, а на карті вказуються тільки індекси цих сегментів.

3.1. Регулярна сітка висот

У цьому випадку дані представляються у вигляді двомірного масиву. Задаються дві координати x , y – по довжині й ширині масиву, а третя координата z задається значенням у конкретному осередку, це висота.

Звичайно карту висот зберігають у файлах картинок. Це дозволяє легко вносити зміни й більш-менш наочно переглядати дані. Тоді двома координатами буде положення конкретного пікселя на картинці, а третя координата буде представлена кольорами (чим вище значення, тим більше значення висоти для цієї точки). Звичайно такі картинки втримуються в монохромному варіанті, але можна використати й всі кольори веселки. Другий варіант дає більше градацій висоти, чим передбачувані 256 градацій у випадку монохромного представлення.

За допомогою цього способу можна представити досить великі простори. Але в нього є один істотний недолік – занадто багато описів для точок, а також, у деяких випадках, спостерігається надмірність даних (наприклад, коли задається проста площина, то, у цьому випадку, для її побудови буде використатися безліч точок, хоча можна б було обійтися трьома). Хоча іноді ця надмірність може бути корисна.

У цього методу існує й кілька позитивних сторін:

- наочність, у будь-якій програмі перегляду графічних файлів можна відразу побачити всю інформацію;
- простота зміни даних, тому що існує безліч програм для роботи з растровою графікою;
- у таких картах можна зберігати не тільки дані про висоту. Наприклад, припустимо, що для зберігання висоти використовується 16 біт, тобто два колірні компоненти. Виходить $256 \cdot 256 = 65536$ градацій висоти. Інші 8 біт можна використати для зберігання інформації про які-небудь особливості ландшафту, наприклад, розташування будинків, будов, мостів, рослинності й так далі.
- Можна зберігати, також числові, значення, але не висот, а блоків (назвемо їх ландшафтними). Дуже просто заздалегідь створити деяку кількість карт висот невеликого розміру (скажемо 8×8 або 16×16 пікселів), а в карті блоків висот зберігати ідентифікатор існуючого блоку. Це дає можливість зберігати й будувати значно більший розмір карти й, отже, ландшафту. Однак потрібно звернути особливу увагу на місця з'єднання блоків. Тобто перший спосіб подання даних для побудови ландшафту плавно перекочував у третій.
- Легкість знаходження координат (і висот) на карті.
- Тому що верхові точки розташовані регулярно й досить близько, можна більш правильно й досить акуратно робити динамічне висвітлення (найчастіше, освітленість вершини прямо залежить від відстані цієї вершини від джерела висвітлення). У цьому й полягає користь від надмірності даних.

3.2. Іррегулярна сітка

Наступний спосіб представлення даних для ландшафтів – іррегулярна сітка вершин і зв'язків їх з'єднуючих. Найчастіше такі рішення застосовуються в спеціалізованих пакетах для ігор (наприклад, редактор рівнів для 3DAction) або спеціальних пакетах для роботи із тривимірною графікою (типу 3Dmax, Maya і т.д.), і зберігаються у вигляді тривимірних моделей. Як правило, цим

способом описують реальні природні ландшафти. Це дає основний вигаш у порівнянні з картами висот:

- використовується значно менше інформації для побудови ландшафту. Необхідно зберігати тільки значення висот кожної вершини й зв'язки ці вершини з'єднуючі. Це дає вигаш у швидкості при передачі величезних масивів інформації в процесі візуалізації ландшафту.

Але крім безсумнівних достоїнств у цього способу є й безліч недоліків:

- алгоритми побудови ландшафтів в основному призначені для регулярних карт висот. Оптимізація таких алгоритмів під цей спосіб вимагає значних зусиль;
- складності при динамічному висвітленні – вершини розташовані досить далеко друг від друга й нерівномірно;
- зберігання, перегляд, модифікація такого ландшафту також представляє складності. При використанні карт висот звичайно користуються досить простими й «стандартними» засобами піксельної графіки. Хоча б тим же MS Paint'ом. При використанні іррегулярних сіток будуть потрібні більше складні пакети.

3.3. Посегментна карта висот

У даному способі також використовуються карти висот. Тільки замість висот у ній зберігаються індекси ландшафтних сегментів. Як ці сегменти представлені, у принципі, ролі не грає. Вони можуть бути й регулярними, і іррегулярними (причому можна використовувати й ті й інші одночасно). Це дає наступні переваги:

- можливість подання величезних відкритих просторів;
- крім самих ландшафтів у таких блоках можна зберігати й інформацію про будинки, будови, рослини, специфічні ландшафтні рішення (наприклад, печери або скелі, що нависають друг над іншому);
- можливість створення декількох варіантів того самого сегмента, але при різному ступені деталізації. Залежно від швидкості або завантаженості

комп'ютера можна вибирати більш-менш деталізовані варіанти (так звані LOD ландшафти – LOD – Level Of Detail).

Недоліків у такого способу теж вистачає:

- проблема стикування різних сегментів;
- неочевидність даних. Глянувши на картинку, дуже складно моментально представити, як це повинне буде виглядати після побудови.

Отже, виникає й проблема модифікації. Якщо при першому варіанті для створення карт висот можна використати Paint, у другому – 3DMax, то для третього, швидше за все, буде потрібно писати власний редактор.

3.4. Алгоритми генерації ландшафтів

У даному розділі будемо розглядати винятково ландшафти з регулярною сіткою. Варто відзначити, що для ландшафтів, описуваних іррегулярними сітками, різниця буде складатися лише в переході до рівномірних сіток до початку роботи алгоритму генерації. А способів перетворення іррегулярних сіток у регулярні на даним момент відомо досить багато.

Розглянемо один з найпростіших алгоритмів генерації деякого ландшафту:

1. Створити двомірний масив необхідного розміру.
2. Генерувати в цей масив (в усі точки) випадкові значення з деякого інтервалу висот.
3. Використати простий фільтр, що згладжує:
 - розглянути точку (усе один по одному);
 - навколо цієї точки беруться значення всіх восьми навколишніх її точок і значення обраної точки;
 - підсумувати всі дев'ять значень;
 - ділити отримане значення на 9 (просте усереднення);
 - отриманий результат записати у вихідну точку;
 - таким чином, обробити весь масив;
4. далі заповнити випадковими значеннями ще кілька точок у вихідному масиві (кількість точок усього – $m \times n$, де m , n – розмірність масиву по

відповідному вимірі, потрібно заповнити приблизно $10m$ точок), конкретна точка вибирається випадковим образом;

5. Повернутися до кроку 3.

Помітимо, що інтервали висот у кроках 2 і 4 необов'язково повинні збігатися. Все залежить від того, наскільки зрізаним повинен бути ландшафт, що генерується.

3.4.1. Генерація ландшафтів з використанням пагорбового алгоритму

Це ітераційний алгоритм, заснований на декількох вхідних параметрах. Він являє собою таку схему:

1. Створити двомірний масив і ініціалізувати його нульовим рівнем (заповнити всі точки нулями).
2. Взяти випадкову точку на ландшафті або біля його границь (за границями), а також випадковий радіус у заздалегідь заданих межах. Вибір цих меж впливає на вид ландшафту – або він буде пологим, або скелястим.
3. В обраній точці «підняти» пагорб заданого радіуса.
4. Повернутися до другого кроку й так далі до обраної кількості кроків. Від нього буде залежати зовнішній вигляд ландшафту, що генерується.
5. Провести нормалізацію ландшафту.
6. Провести «долинізацію» ландшафту, тобто зробити його схили більше пологими.

Перший, другий і четвертий кроки тривіальні, п'ятий і шостий розглянемо далі. Тепер же займемося третім кроком. Що ж означає «підняти» пагорб? Фактично пагорб – це в нашому випадку половина сфери (або будь-яка інша схожа на пагорб поверхня, розглянута в главі 2), чим більше радіус – тим більше пагорб (і вище).

Для сфери маємо

$$z^2 = r^2 - ((x_2 - x_1)^2 + (y_2 - y_1)^2),$$

де (x_1, y_1) – задана точка, r – обраний радіус, (x_2, y_2) – висота пагорба. Для інших розглянутих у главі 2 поверхонь всі набагато складніше, однак, одержуваний при їхньому використанні ландшафт буде більш реалістичним й гарнішим.

Щоб згенерувати ландшафт повністю необхідно побудувати безліч таких пагорбів. Але є ще дві деталі, на які необхідно звернути увагу. Перше – необхідно ігнорувати негативні значення висоти пагорба. Друге – при генерації наступних пагорбів краще додавати отримане значення для даного пагорба до вже існуючих значень. Це дозволяє побудувати більше правдоподібний ландшафт, ніж правильно обкреслені округлі пагорби.

Розглянемо тепер нормалізацію ландшафту (крок 5 алгоритму). При генерації значень для ландшафту ми не враховували виходи цих значень за деякі межі (наприклад – якщо потім ландшафт буде зберігатися в монохромній картинці, то нам необхідно, щоб всі значення перебували в межі від 0 до 255). Для цього необхідно зробити нормалізацію значень. Математично нормалізація – це процес одержання значень із однієї межі, і переклад його в інші межі. Як це виглядає графічно показано на рис. 3.1.

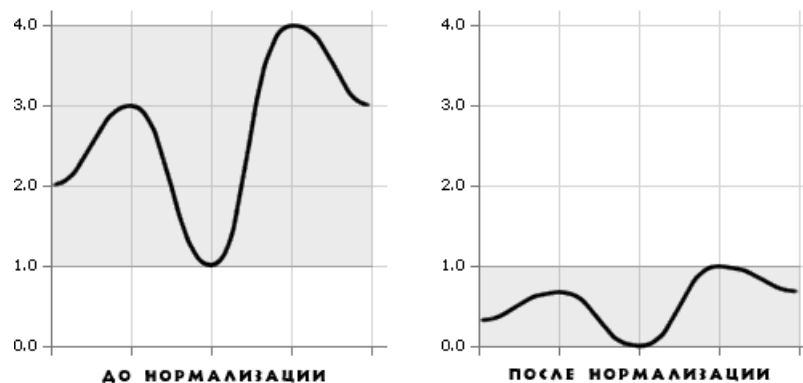


Рис. 3.1. Нормалізація

Щоб це зробити, потрібно виконати наступні дії:

- спочатку пройти по всьому масиву й визначити найбільше й найменше значення висот;
- після цього, зробити нормалізацію конкретних значень у межі від 0 до 1.

Математично це виглядає так:

$$z_{norm} = \frac{z - z_{min}}{z_{max} - z_{min}}$$

У результаті одержуємо нормалізований ландшафт, який можна використовувати надалі. Тепер перейдемо до питання про «долинізацію» ландшафту.

Всі значення зараз перебувають у межах від 0 до 1. Ідея «долинізація» полягає в наступному – взяти від кожного значення квадратний корінь. Це більшою мірою впливає на середні значення, практично не зачіпаючи мінімумів і максимумів. Графічно це показано на рис. 3.2.

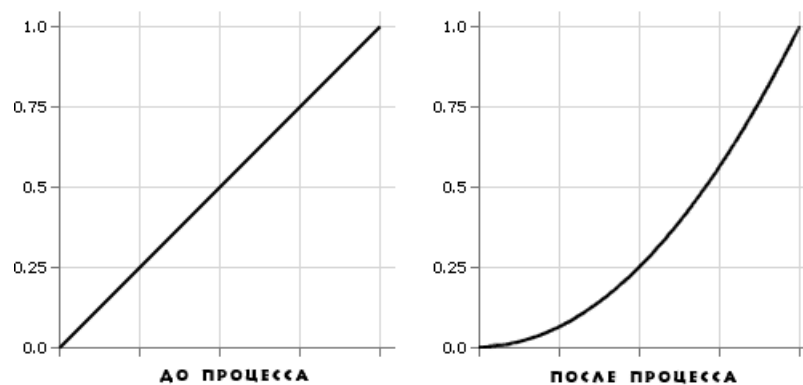


Рис. 3.2. Долинізація

В основному, розглянуті алгоритми призначені для створення простого горбкуватого або гористого ландшафту. Але існують і інші типи ландшафтів. Наприклад, острова (точніше групи островів), озерні ландшафти. Їх можна реалізувати досить просто:

- створити простий, досить горбкуватий ландшафт;
- потім перемістити рівень води нагору або вниз (при цьому варто обмовитися, що ми приймемо за рівень води, звичайно мається на увазі нульовий рівень там, де координати висоти нульові).

Практично просто проходиться весь масив висот, зміщаючи їх на якесь значення.

Розглянемо ще один тип ландшафтів – одиночні острови або гірські плато (як часто показують в американських фільмах), залежно від того, де потім «розмістити» воду.

3.4.2. Модифікація пагорбового алгоритму для островів

У багатьох випадках можна використати вже розглянутий алгоритм для генерації ландшафтів. Але іноді необхідно згенерувати острова, або острів. У цьому може допомогти той же алгоритм, правда злегка модифікований.

У вихідному алгоритмі центральна точка вибиралася випадковим образом, і вона могла розташовуватися в будь-якій частині ландшафту. Тепер же потрібно, щоб пагорби були розташовані ближче до центра. Щоб зробити це, уведемо дві змінних (які потім будемо випадковим образом змінювати), назвемо їх відстань і кут. Відстань буде означати, як далеко від центра перебуває центральна точка для одиночного пагорба. Воно може змінюватися від нуля (прямо по центру карти висот) до половини величини карти висот мінус радіус пагорба. Це дозволить уникнути ситуацій перетинання пагорбів із краєм карти висот. Кут буде показувати, у якому напрямку від центра потрібно буде згенерувати пагорб. Він змінюється в межах від 0 до 2π . Використовуючи ці два значення, можна одержати значення (x, y) для центральної точки конкретного пагорба й використати їх, як і в простому алгоритмі. Значення для x й y можна одержати в такий спосіб:

$$x = \frac{R}{2} + \cos(\theta)r,$$

$$y = \frac{R}{2} + \sin(\theta)r,$$

де R – розмір карти висот, r – відстань, θ – деякий кут, що змінюється в межах від 0 до 2π . Варто пам'ятати про те, що радіус повинен бути менше половини розміру карти висот.

У такий спосіб можна згенерувати острів.

3.4.3 Формування зображення гори за допомогою фракталів

У завданнях комп'ютерної графіки фрактальна технологія одержала найбільше поширення при формуванні деталей ландшафту, зокрема гір. Для цього використовується метод рекурсивної розбивки в середній точці, де як вихідний об'єкт береться тетраедр. Розглянемо одну грань тетраедра (рис. 3.3). Спочатку визначаються середні точки кожного із трьох ребер грані. Потім кожна середня

точка зміщується випадковим образом. У результаті формується чотири нові трикутні грані. Управляючи характеристиками генератора випадкових чисел, що задає зсув, можна домагатися потрібного ступеня м'якості або різкості рельєфу гори. Варто звернути увагу на те, що до вибору характеристик генератора потрібно віднестися дуже уважно, інакше результат не виправдає очікувань.

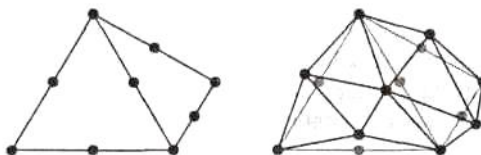


Рис. 3.3. Розбивка в середній точці граней тетраедра

Алгоритм розбивки в середній точці можна застосовувати стосовно будь-якої полігональної сітки. Можна починати із плоскої мережі із чотирикутних осередків у площині x, z , а далі розбивати кожний чотирикутник на чотири більше дрібних і зміщати всі вершини випадковим образом нагору (у напрямку осі висот).

Існує безліч інших алгоритмів для генерації ландшафтів, однак наведені найбільш прості в реалізації, легко модифікуються й дозволяють одержувати ефектні результати з невеликими витратами.

За допомогою алгоритмів генерації ландшафтів можна одержувати як існуючі природні, рельєфні поверхні, так і неіснуючі, які потрібно одержати, наприклад, при розробці комп'ютерних ігор. Для генерації існуючих ландшафтів потрібно лише знати топографічні особливості місцевості (карту висот і деяку іншу інформацію).

РОЗДІЛ 4

КОМП'ЮТЕРНА МОДЕЛЬ СКЛАДНИХ ЛАНДШАФТНИХ ФОРМ

4.1. Комп'ютерне моделювання рельєфних поверхонь, що описуються функціональними залежностями

Побудова поверхонь різних форм робиться за допомогою створення полігональних сіток, розмір яких вибирається з міркувань доцільності. Якщо деяка ландшафтна форма представлена у вигляді функціональної залежності, то її комп'ютерна модель створюється досить швидко. Варто відзначити, що всі поверхні, розглянуті в розділі 2, дозволяють одержати ландшафтні форми, описані функціонально. Як правило, такі поверхні будуть складатися з різних «шматків», але за допомогою методів, розглянутих у тім же розділі, їх можна зробити гладкими. Дуже часто можливе представлення деяких існуючих або генерованих ландшафтів за допомогою аналітичних залежностей. Так поверхня, що задається формулою

$$z = \sqrt{x^2 + y^2} \sin(x) \cos(y),$$

має вигляд складної гористої місцевості (рис. 4.1).

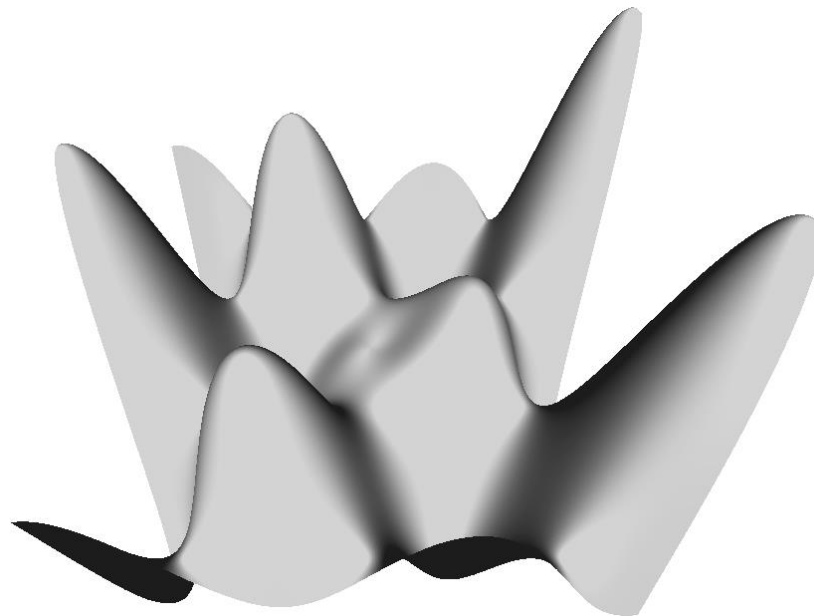


Рис. 4.1. Гористий ландшафт, що описується функціональною залежністю

Функція

$$z = \sqrt{x^2 + y^2} \sin^3(x) \cos(x) \cos(y)$$

описує поверхню, що нагадує горбкуватий ландшафт. Її зображення показано на рис. 4.2.

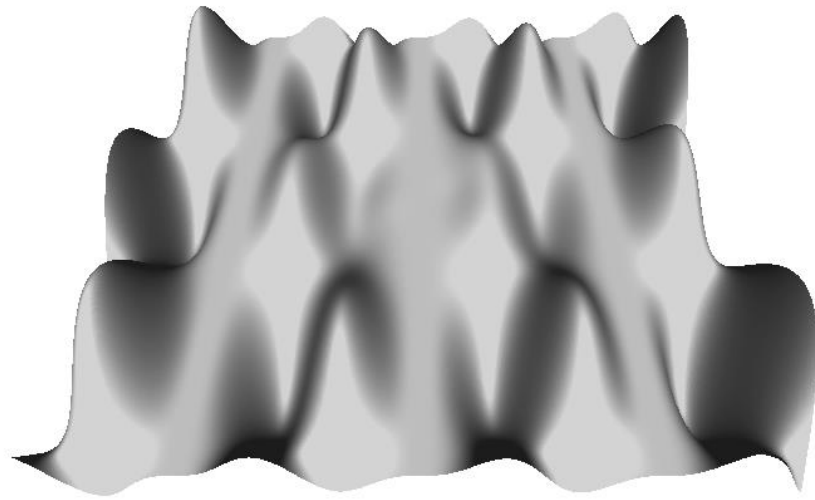


Рис. 4.2. Горбкуватий ландшафт

Іноді, щоправда, можна одержати дуже цікаві, але досить сюрреалістичні зображення, прямо з ландшафтами не пов'язані, однак все-таки представляючи деякий інтерес. Приклади таких зображень представлені на рис. 4.3. і 4.4.

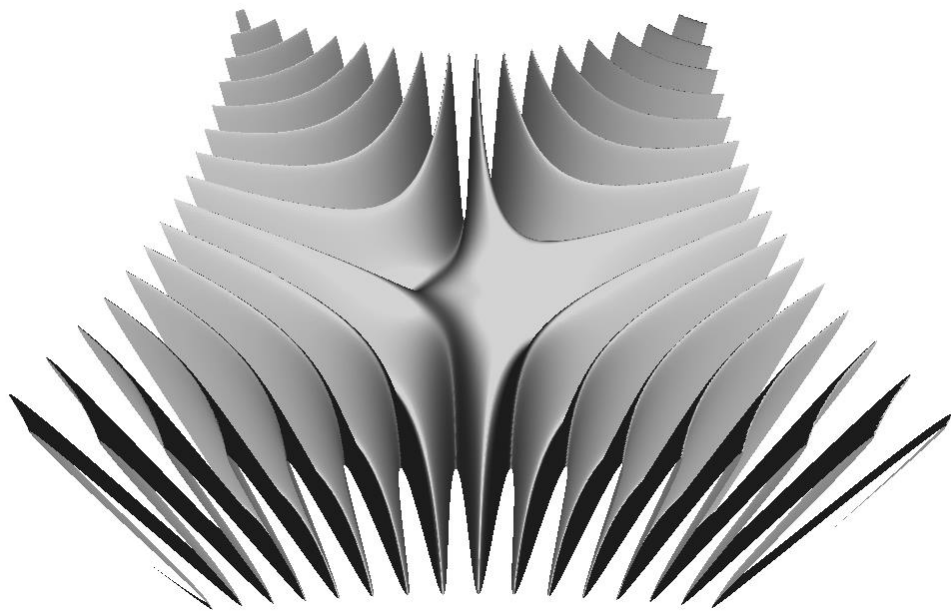


Рис. 4.3. Поверхня, що описується функцією $z = \sqrt{x^2 + y^2} \sin(xy) \cos(xy)$

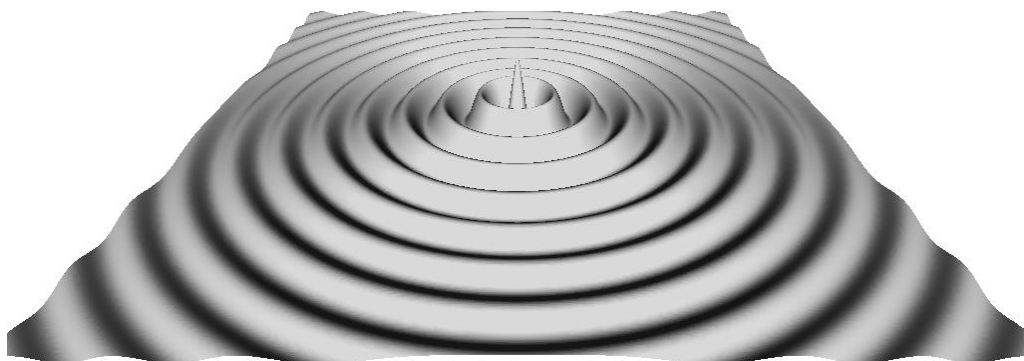


Рис. 4.4. Поверхня, що описується функцією $z = \frac{\cos(10\sqrt{x^2 + y^2})}{1 + 3\sqrt{x^2 + y^2}}$

Як видно, поверхні досить цікаві, однак не завжди можливо їхнє подання аналітичними залежностями. Якщо ділянка ландшафту дуже складна або дані про неї неточні, то побудувати її можна лише скориставшись алгоритмами генерації складних ландшафтних форм. Варто відзначити також, що їхнє застосування обґрунтоване й при необхідності одержати моделі будь-яких інших (у тому числі описаних аналітично) поверхонь, тому що витрати при їхньому використанні досить малі.

4.2. Генерація ландшафту за допомогою стандартного алгоритму

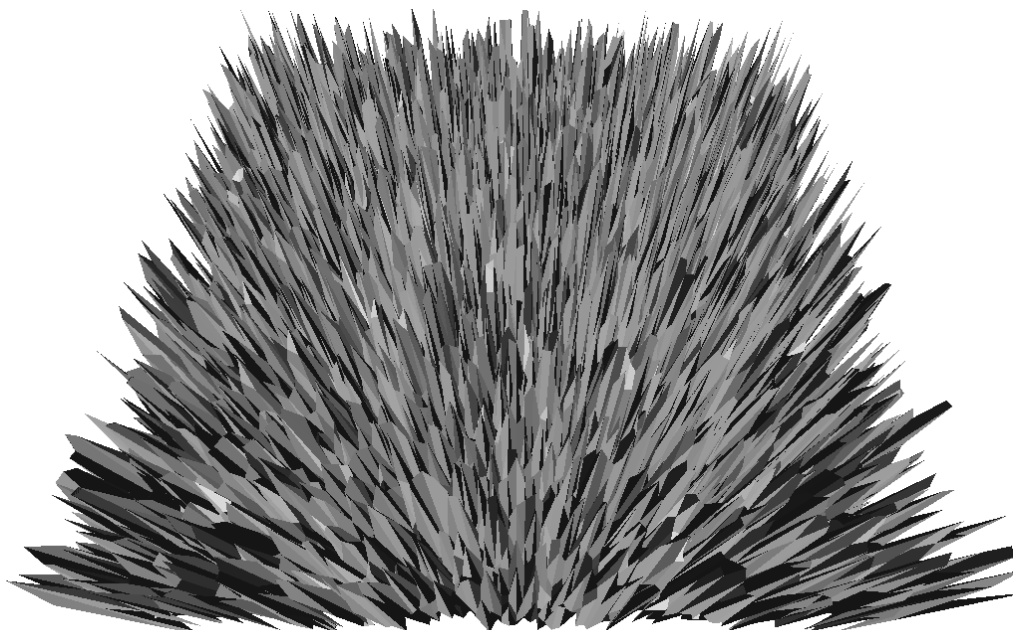


Рис. 4.5. Результат роботи простого алгоритму на 2 кроці

Розглянутий у попередній главі простий алгоритм генерації ландшафтів дозволяє будувати різні ландшафти: від гористих до рівнинних. Після другого кроку алгоритму генерована поверхня має вигляд, показаний на рис. 4.5.

Далі після застосування фільтра, що згладжує, виходить наступне зображення (рис. 4.6).

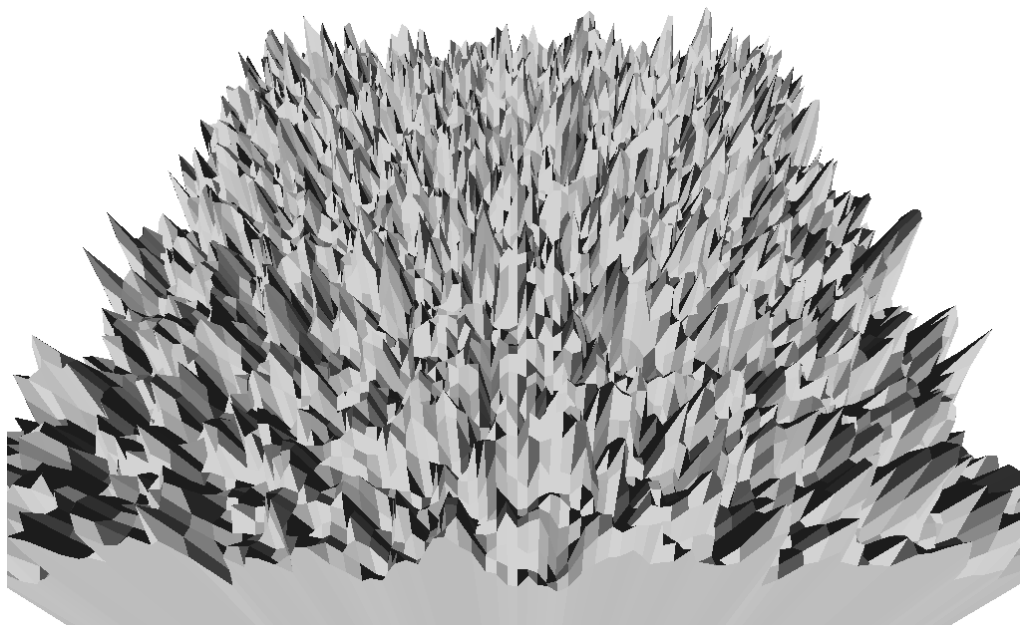


Рис. 4.6. Результат роботи простого алгоритму після згладжування: гористий ландшафт із високим рівнем зрізаності

Далі, отриманий ландшафт можна зробити ще більш зрізаним (рис. 4.7).

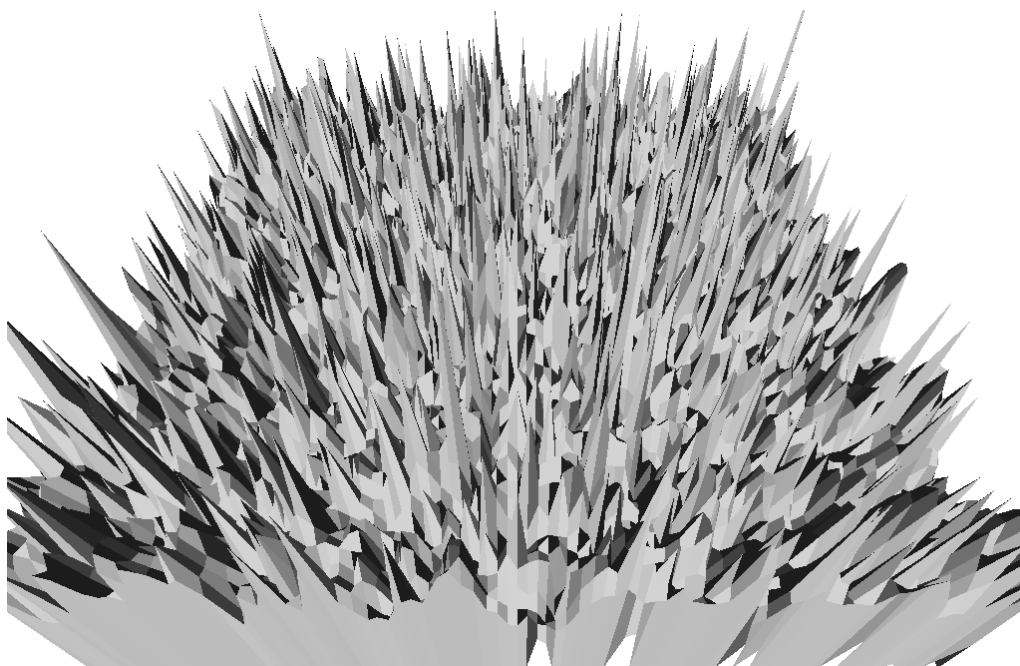


Рис. 4.7. Збільшення рівня зрізаності генерованого ландшафту

Як видно, результат не дуже нагадує реальну гористу місцевість, тому звичайно наприкінці застосовують згладжування, домагаючись необхідної відповідності. Результати коригувальних впливів показані на рис. 4.8.

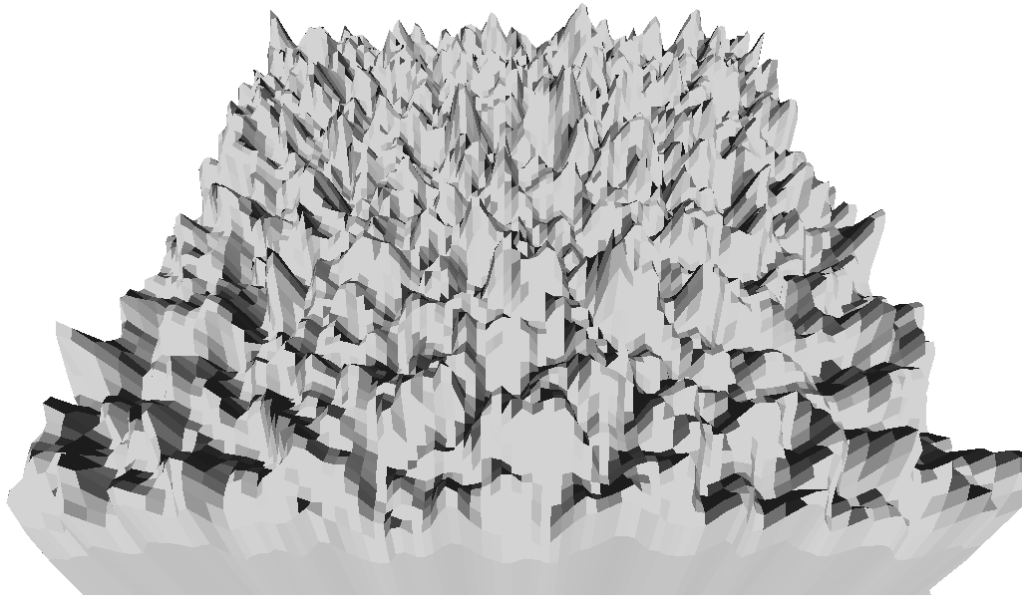


Рис. 4.8. Корегування зрізаності

Підсилюючи ефект, що згладжує, можна домогтися того, що ландшафт стане горбкуватим (рис. 4.9.)

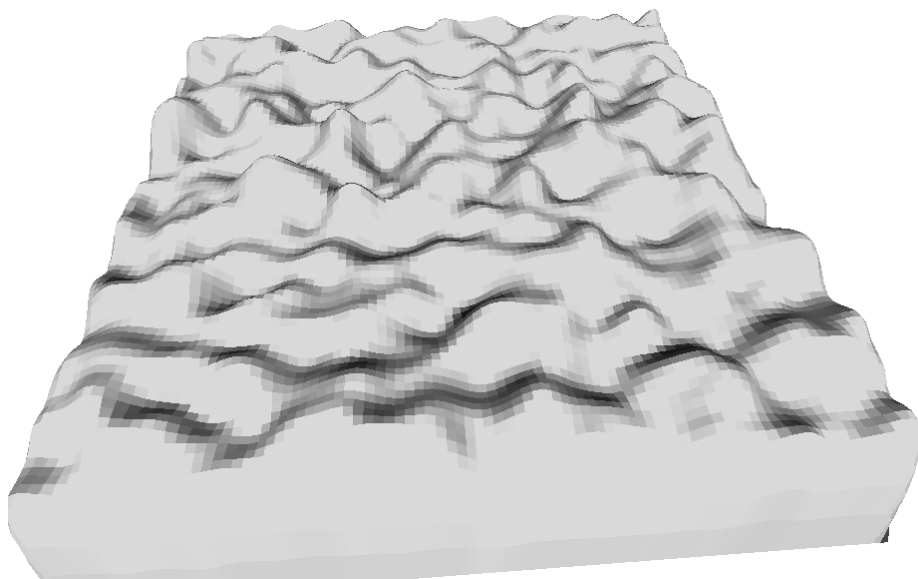


Рис. 4.9. Горбкуватий ландшафт

або й зовсім рівнинним (рис. 4.10)

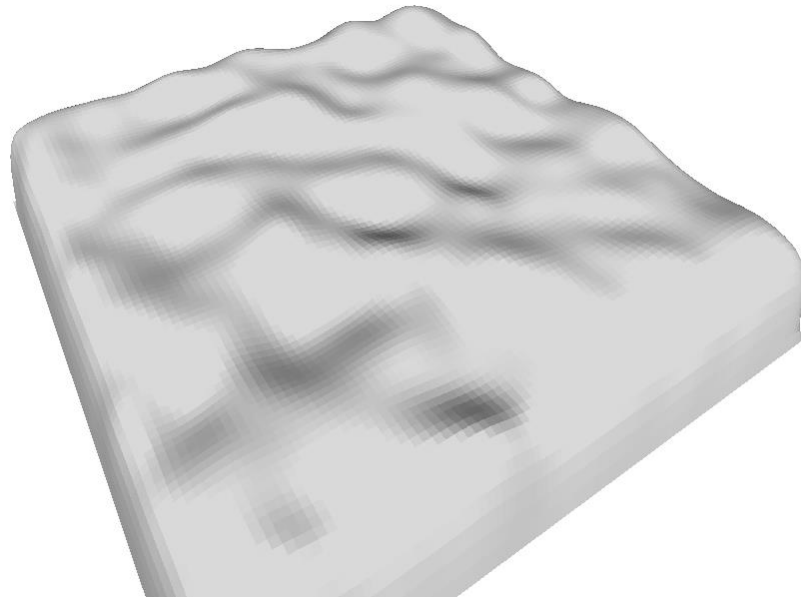


Рис. 4.10. Ландшафт близький до рівнинного

Використовуючи даний алгоритм можна одержати практично будь-яку ландшафтну форму, однак іноді доводиться дуже довго варіювати ступінь зрізаності й ефект згладжування для одержання потрібного результату.

4.3. Генерація ландшафту за допомогою пагорбового алгоритму

Простий алгоритм генерації ландшафту дає можливість одержувати широкий спектр складних природних поверхонь, однак іноді на цей процес витрачається багато часу, оскільки більшість одержуваних ландшафтів у неопрацьованому виді виглядають занадто нереально. Цих недоліків позбавлений пагорбовий алгоритм.

На першому кроці генерується початкова площина, на якій буде будуватися ландшафт (рис. 4.11).



Рис. 4.11. Початкова площина

Далі в деякій точці з координатами (x, y) генерується пагорб деякої висоти h (рис. 4.12).

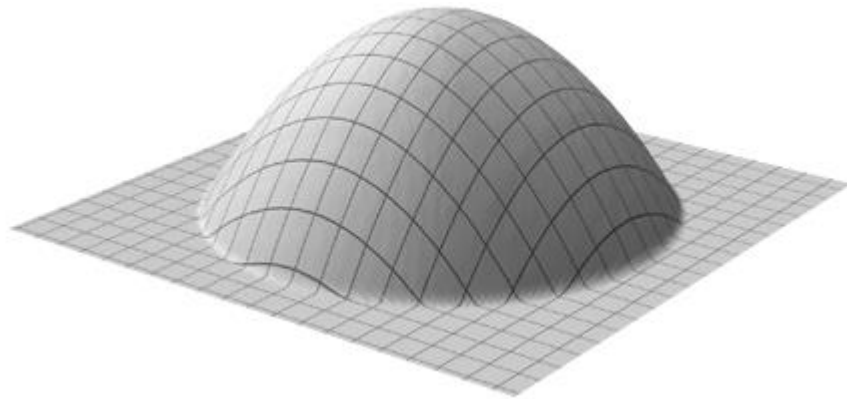


Рис. 4.12. Одиночний пагорб

Щоб згенерувати ландшафт повністю необхідно побудувати безліч пагорбів. Результати роботи алгоритму представлені на рис. 4.13.-4.16.

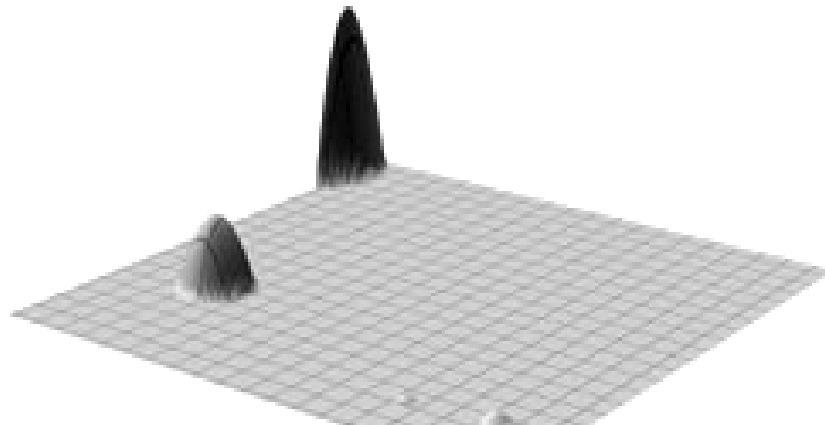


Рис. 4.13. Результат роботи пагорбового алгоритму на 5 кроці

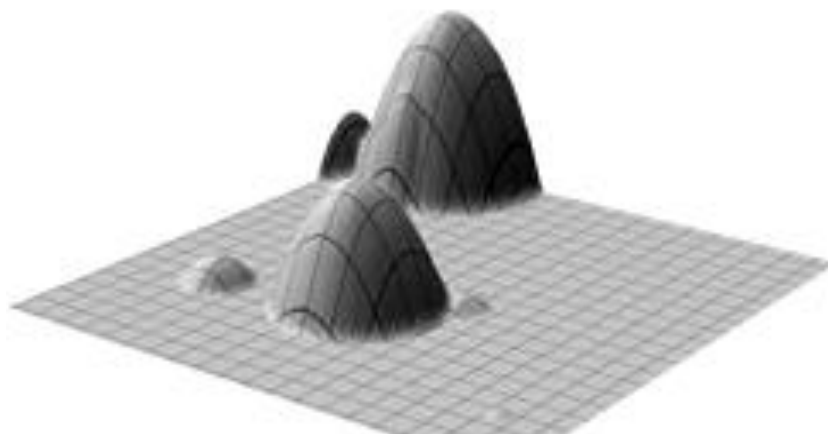


Рис. 4.14. Результат роботи пагорбового алгоритму на 10 кроці

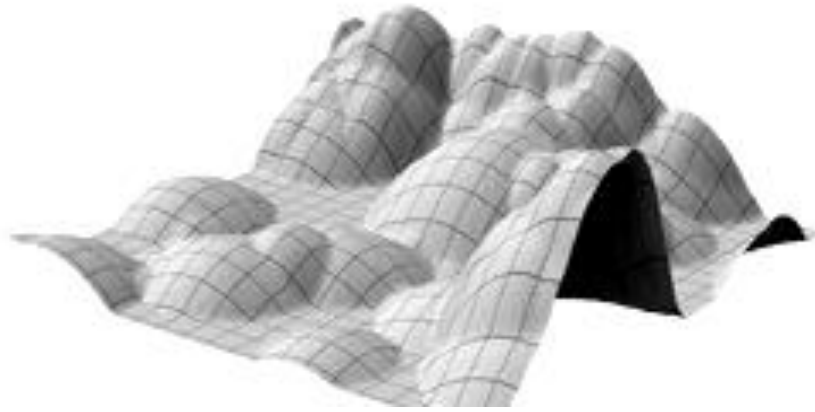


Рис. 4.15. Результат роботи пагорбового алгоритму на 50 кроці



Рис. 4.15. Результат роботи пагорбового алгоритму на 200 кроці

Отриманий ландшафт нагадує реальну гористу місцевість. Однак для одержання остаточного результату потрібно провести його нормалізацію й «долинізацію» (рис. 4.17).



Рис. 4.17. Остаточний результат роботи пагорбового алгоритму

Варіюючи кількість «долинізацій» можна збільшити або зменшити ступінь зрізаності одержуваного ландшафту.

Розглянутий алгоритм є простим у реалізації й дає досить ефектні результати. Як уже згадувалося в розділі 3 його можна модифікувати для моделювання островів або гірських плато. Результати роботи алгоритму для цих випадків показані на рис. 4.18 - 4.19.

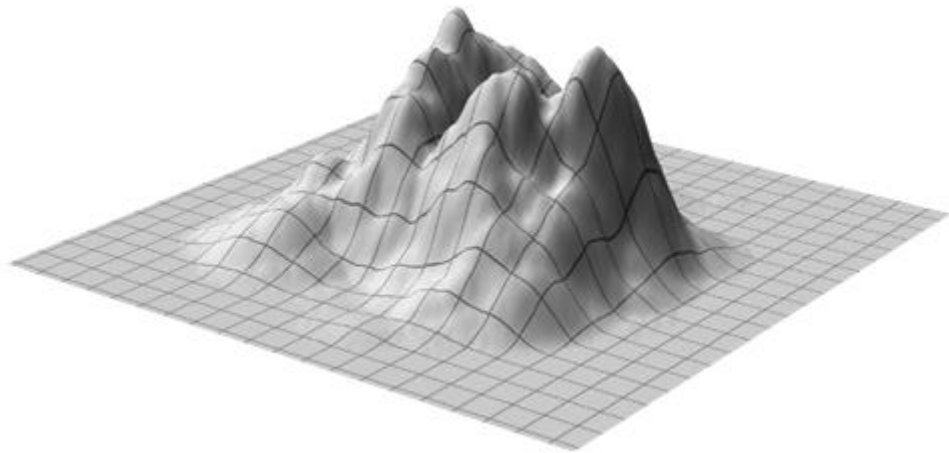


Рис. 4.18. Гірське плато

Острів буде відрізнятися тільки наявністю води.

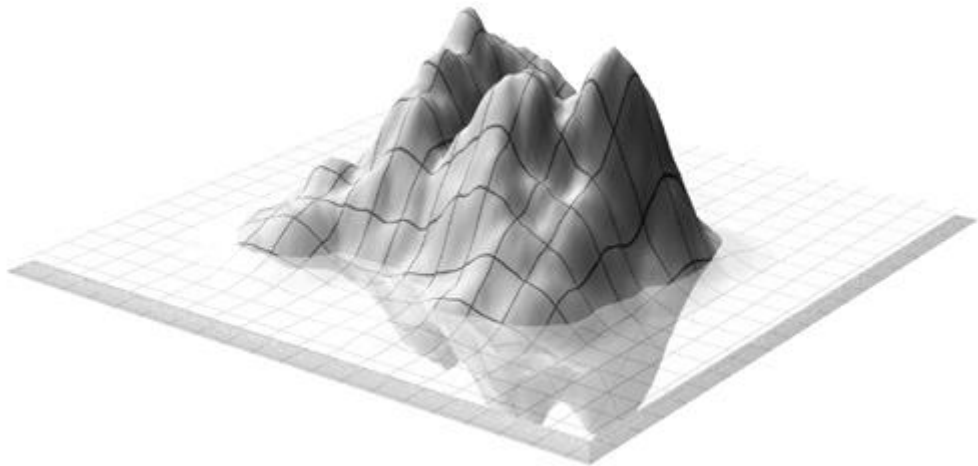


Рис. 4.19. Острів

4.4. Генерація ландшафту за допомогою фрактального алгоритму

Реалізація фрактального алгоритму дуже мало відрізняється від простого алгоритму генерації ландшафтів. Різниця складається лише в тім, що в одному випадку використовується рекурсія, а в іншому – ітераційний процес. Результати роботи фрактального алгоритму показані на рис. 4.20.

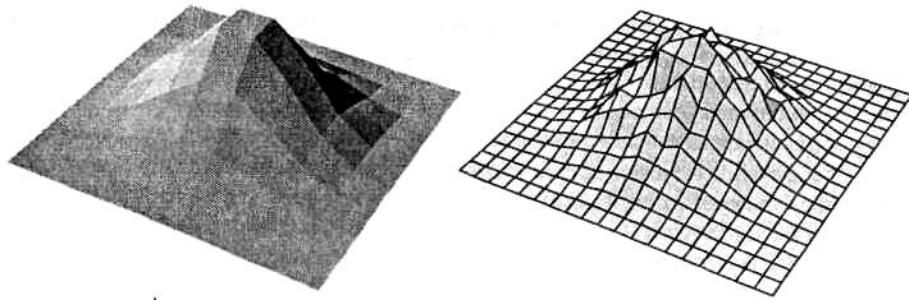


Рис. 4.20. Гора, що генерована за допомогою фрактального алгоритму

Розглянуті алгоритми дозволяють одержувати різні складні ландшафтні форми й працювати з ними. Їхня реалізація є досить простою і економічною.

ВИСНОВОК

Для того щоб полегшати роботу у різних сферах діяльності дизайнерів, архітекторів, розроблювачів комп'ютерних ігор тощо треба використовувати новітні розробки в галузі комп'ютерної техніки, геометричного моделювання та ін. У даній роботі проведено систематизацію відомостей про побудову різних типів поверхонь, розглянуто декілька алгоритмів генерації складних ландшафтних форм, створено кілька їх комп'ютерних моделей. Але в обсязі дослідження магістра неможливо розкрити весь масштаб проблеми моделювання складних географічних ландшафтів, тому потрібно продовжувати дослідження у цьому напрямку й можливо до створення якоїсь новітньої геоінформаційної системи.

ЛИТЕРАТУРА

1. Ньюмен У., Спрулл Р. Основы интерактивной машинной графики. – М.: Мир, 1976. – 200с.
2. Павлидис Т. Алгоритмы машинной графики и обработки изображений. – М.: Радио и связь, 1986. – 250с.
3. Фоли Дж., ван Дэм А. Основы интерактивной машинной графики. В 2-х книгах. – М.: Мир, 1985. – 400с.
4. Е.В. Шикин, А.В. Боресков Компьютерная графика полигональные модели. – М.: Диалог-МИФИ, 2000г. – 461с.
5. Н.Н. Голованов Геометрическое моделирование. – М.: Физматлит, 2002, – 472с.
6. Роджерс Д., Адамс Дж. Математические основы машинной графики: пер. с англ. – М.: Мир, 2001. – 604с.
7. Алберг Дж., Нильсон Э., Уолш Дж. Теория сплайнов и ее приложения. – М.: Мир, 1972. – 156с.
8. Березин И.С, Жидков Н.П. Методы вычислений. Т. 1,2. – М.: Наука, 1966. – 220с.
9. Де Бор К. Практическое руководство по сплайнам. – М.: Радио и связь, 1985. – 450с.
10. Дубровин Б.А., Новиков С.П., Фоменко А.Т. Современная геометрия. – М.: Наука, 1986. – 326с.
11. Ильин В.А., Позняк Э.Г. Аналитическая геометрия. – М.: Наука, 1981. – 240с.
12. Калиткин Н.Н. Численные методы. – М.: Наука, 1978. – 356с.
13. Препарата Ф., Шеймос М. Вычислительная геометрия. – М.: Мир, 1989. – 228с.
14. Самарский А.А., Гулин А.В. Численные методы. – М.: Наука, 1989. – 310с.
15. Мищенко А.С., Фоменко А.Т. Курс дифференциальной геометрии и топологии. – М.: Изд-во «Факториал Пресс», 2000. – 448с.

16. Тихомиров Ю. Программирование трехмерной графики. – СПб.: ВHV-Санкт-Петербург, 1998. – 200с.
17. Иванов В.П., Трехмерная компьютерная графика. – М.: Радио и связь, 1988. – 450с.
18. Майкл Л. Вычислительная геометрия и компьютерная графика на C++. – М: Бином, 1997. – 356с.
19. Шикин Е.В., Плис А.И. Кривые и поверхности на экране компьютера. – М.: Диалог-МИФИ, 1996. – 286с.
20. Краснов М.В. OpenGL графика в проектах Delphi. – С.-Пб.: ВHV, 2000. – 270с.

ДОДАТОК

Код програми генерації ландшафту

```
unit main;

interface

uses

  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  OpenGL, Addition, Menus, paramInput, about, FractDateInput;

type

  TMnFrmGL = class(TForm)
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N11: TMenuItem;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure FormKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormResize(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N11Click(Sender: TObject);
    procedure N7Click(Sender: TObject);
    procedure N8Click(Sender: TObject);
```

```

private
  { Private declarations }
  DC : HDC;
  hrc: HGLRC;
  xAngle, yAngle, zAngle: Real;
  eyex, eyey, eyez,
  centerx, centery, centerz,
  upx, upy, upz: Real;
  CountFnc: Integer;
  StartFlag: Boolean;
public
  { Public declarations }
  k: Integer;
  FracFkag: Boolean;
protected
  procedure WMPaint(var Msg: TWMPaint); message WM_PAINT;

end;

var
  MnFrmGL: TMnFrmGL;

implementation

{$R *.DFM}

procedure TMnFrmGL.WMPaint(var Msg: TWMPaint);
var
  ps : TPaintStruct;
begin
  glClear( GL_COLOR_BUFFER_BIT or GL_DEPTH_BUFFER_BIT );

  glPushMatrix;
  gluLookAt(eyex, eyey, eyez, centerx, centery, centerz, upx, upy,
upz);

```

```

    glRotatef(xAngle, 1.0, 0.0, 0.0);
    glRotatef(yAngle, 0.0, 1.0, 0.0);
    glRotatef(zAngle, 0.0, 0.0, 1.0);
    if not StartFlag then
    AnalyticSurface(CountFnc) else
    SimpleGenerate;
    if FracFkag then
    FractalGenerate;
    glPopMatrix;
    SwapBuffers(DC);
    EndPaint(Handle, ps);
end;

{=====
=====
Формат пикселя}
procedure SetDCPixelFormat (hdc : HDC); // hdc - ссылка на
контекст воспроизведения
var
    pfd : TPixelFormatDescriptor; // описание формата пикселя
    nPixelFormat : Integer;      // формат пикселя
begin
    FillChar (pfd, SizeOf (pfd), 0); // создание описания формата
пикселя
    pfd.dwFlags := PFD_DRAW_TO_WINDOW or PFD_SUPPORT_OPENGL or
PFD_DOUBLEBUFFER; // флаги дескриптора
    nPixelFormat := ChoosePixelFormat (hdc, @pfd); // получение
индекса формата пикселя
    SetPixelFormat (hdc, nPixelFormat, @pfd); // установка формата
пикселя
end;

procedure TMnFrmGL.FormCreate(Sender: TObject);
const lposition: Array[0..3] of GLfloat = (0.0, 0.0, -5.0, 1.0);
begin
    Randomize;

```

```

StartFlag := True;
FracFkag  := False;
CountFnc  := 0;

xAngle := 0.0;
yAngle := 0.0;
zAngle := 0.0;
k      := 10;

eyex   := 0.0;
eyey   := 0.0;
eyez   := 10.0;
centerx := 0.0;
centery := 0.0;
centerz := 0.0;
upx     := 0.0;
upy     := 1.0;
upz     := 0.0;

DC := GetDC (Handle); // получение ссылки на контекст устройства
SetDCPixelFormat(DC); // установка формата пикселя
hrc := wglCreateContext(DC); // создание контекста
воспроизведения
wglMakeCurrent(DC, hrc); // установка контекста
glClearColor (1.0, 1.0, 1.0, 1.0); // цвет фона
glColor3f (0.3, 0.2, 0.6);          // текущий цвет примитивов
// glPolygonMode (GL_FRONT_AND_BACK, GL_LINE); // режим вывода
glEnable(GL_DEPTH_TEST);
glEnable(GL_LINE_SMOOTH);
glEnable(GL_NORMALIZE);
glEnable(GL_LIGHT0);
glEnable(GL_LIGHTING);
glEnable(GL_COLOR_MATERIAL);
glEnable(GL_AUTO_NORMAL);
end;

```



```

procedure TMnFrmGL.FormDestroy(Sender: TObject);
begin
  wglMakeCurrent(0, 0); // освобождение контекста воспроизведения
  wglDeleteContext(hrc); // удаление контекста воспроизведения
  ReleaseDC(Handle, DC); // освобождение контекста устройства
  DeleteDC(DC); // удаление контекста устройства
end;

procedure TMnFrmGL.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = VK_ESCAPE then Close;
  if Key = VK_UP      then centery := centery - 0.5;
  if Key = VK_DOWN   then centery := centery + 0.5;
  if Key = VK_LEFT   then centerx := centerx + 0.5;
  if Key = VK_RIGHT  then centerx := centerx - 0.5;
  if Key = 90        then eyez     := eyez + 0.5;
  if Key = VK_SHIFT  then eyez     := eyez - 0.5;
  if Key = 49        then xAngle   := Rot(xAngle, True);
  if Key = 50        then xAngle   := Rot(xAngle, False);
  if Key = 51        then yAngle   := Rot(yAngle, True);
  if Key = 52        then yAngle   := Rot(yAngle, False);
  if Key = 53        then zAngle   := Rot(zAngle, True);
  if Key = 54        then zAngle   := Rot(zAngle, False);
  if Key = 55        then glPolygonMode(GL_FRONT_AND_BACK,
GL_LINE); // режим вывода
  if Key = 56        then glPolygonMode(GL_FRONT_AND_BACK,
GL_FILL); // режим вывода
  if Key = VK_SPACE  then begin
                        inc(CountFnc);
                        if CountFnc > 5 then CountFnc := 0;
                      end;

  if Key = 13 then glScalef(0.5,0.5,0.5);
end;

procedure TMnFrmGL.FormResize(Sender: TObject);

```

```

begin
  glViewport(0, 0, ClientWidth, ClientHeight); // область вывода
  glMatrixMode (GL_PROJECTION); // текущая матрица проекций
  glLoadIdentity; // возврат к исходной позиции
// glFrustum(-1.0,1.0,-1.0,1.0,1.0,15.0);
  gluPerspective (60.0, ClientWidth/ClientHeight, 1.0, 1500.0); //
параметры вида
// gluLookAt(eyex, eyez, eyez, centerx, centery, centerz, upx,
upy, upz);
  glMatrixMode (GL_MODELVIEW); // текущая матрица модели
  glLoadIdentity; // возврат к исходной позиции

  // этот фрагмент нужен для придания трёхмерности
// glTranslatef(0.0, 0.0, -10.0); // перенос объекта - ось Z
// glRotatef(90.0, 1.0, 0.0, 0.0); // поворот объекта - ось X
// glRotatef(70.0, 0.0, 1.0, 0.0); // поворот объекта - ось Y

  InvalidateRect(Handle, nil, False); // перерисовка
end;

procedure TMnFrmGL.N4Click(Sender: TObject);
begin
  InputParamForm.Show;
  InputParamForm.AdditionalButton.Enabled := False;
  InputParamForm.EndGenButton.Enabled := False;
end;

procedure TMnFrmGL.N2Click(Sender: TObject);
begin
// InputParamForm.SimpleGenPanel.Visible := False;
  StartFlag := False;
  N4.Enabled := True;
end;

procedure TMnFrmGL.N6Click(Sender: TObject);

```

```
begin
    aboutForm.ShowModal;
end;

procedure TMnFrmGL.N11Click(Sender: TObject);
begin
    Application.Terminate;
end;

procedure TMnFrmGL.N7Click(Sender: TObject);
begin
    InputParamForm.SimpleGenPanel.Show;
    N4.Enabled := True;
end;

procedure TMnFrmGL.N8Click(Sender: TObject);
begin
    FractDateForm.ShowModal;
end;

end.
```