

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Харківський національний економічний університет
імені Семена Кузнеця**

3D-графіка

Навчальний посібник

Укладач

В. П. Гаврилов

Відповідальний за випуск

О. І. Пушкар

Харків ХНЕУ ім. С. Кузнеця 2018

УДК 004.92(075)

T-76

Рецензенти:

Доцент кафедри бойового застосування та експлуатації АСУ факультету автоматизованих систем управління та наземного забезпечення польотів авіації Харківського університету Повітряних Сил імені Івана Кожедуба кандидат технічних наук, доцент Голубничий Дмитро Юрійович;

доцент кафедри медіасистем і технологій Харківського національного університету радіоелектроніки т технічних наук, доцент Дейнеко Жанна Валентинівна.

Рекомендовано до видання розв'язанням вченої ради Харківського національного економічного університету імені Семена Кузнеця.

Протокол № 10 від 03.04 2018 р.

3D-графіка [Електронній ресурс]: навчальний посібник / В. П. Гаврилов. – Харків: ХНЕУ ім. С. Кузнеця, 2018. 135 с.

ISBN _____

Наведено матеріал, який допомагає студентам засвоїти базові теми навчальної дисципліни. Подано чотири логічно взаємопов'язаних теми, де розглянуто загальні питання, зв'язаних зі створенням аналітичних, сплайнових і полігональних 3D-об'єктів. Рекомендовано для студентів спеціальності 186 "Видавництво та поліграфія" усіх форм навчання.

УДК 004.92(075)

© Харківський національний
економічний університет
імені Семена Кузнеця, 2018
© Гаврилов В. П., 2018

Вступ

3D-графіка – це процес створення об'ємної моделі за допомогою спеціальних комп'ютерних програм. На основі креслень, малюнків, детальних описів або будь-якої іншої графічної чи текстової інформації 3D дизайнер створює об'ємне зображення.

У спеціальній програмі модель можна подивитися з усіх боків, вбудувати на будь-яку площину і у будь-яке оточення. Тривимірною комп'ютерна графіка, як і векторна, є об'єктно-орієнтованою, що дозволяє змінювати як всі елементи тривимірної сцени, так і кожен об'єкт окремо. За допомогою графічних редакторів 3D-графіки можна виконувати наочні зображення.

Тривимірною графікою може бути будь-якої складності. Можна створити просту тривимірну модель, з низькою деталізацією і спрощеної форми. Це також може бути складна модель, у якій наявне опрацювання найдрібніших деталей.

3D-графіка застосовується у дуже багатьох сферах. У першу чергу, це створення архітектурних, ландшафтних та інтер'єрних дизайн-проектів.

3D-моделей дуже популярні у будівництві сайтів. Для створення особливого ефекту деякі творці сайтів додають у дизайн не просто графічні елементи, а тривимірні моделі, іноді навіть і анімовані.

Звичайно, у основному тривимірні моделі використовуються з демонстраційною метою. Вони незамінні для презентацій, виставок, а також використовуються у роботі з клієнтами, коли необхідно наочно показати, яким буде підсумковий результат. Крім того, методи тривимірного моделювання потрібні там, де треба показати у об'ємі вже готові об'єкти або ті об'єкти, які існували колись давно.

Перевага у тривимірного моделювання перед іншими способами візуалізації досить багато. Тривимірне моделювання дає дуже точну модель, максимально наближену до реальності. Сучасні програми допомагають досягти високої деталізації. Разом із тим значно підвищується наочність проекту. Зобразити тривимірний об'єкт у двомірній площині не просто, тоді як 3D-візуалізація дає можливість ретельно проробити і що найголовніше, проглянути всі деталі. Це природніший спосіб візуалізації.

Існує досить велика кількість найрізноманітніших програм для 3D-графіки. Так, однією з популярних програм, які спеціально розроблені

для створення тривимірної графіки і дизайну інтер'єрів, є програма 3ds max. Вона дозволяє реалістично візуалізувати об'єкти різноманітної складності. Крім того, ця програма дає можливість компонувати їх, задавати траєкторії переміщень і зрештою навіть створювати повноцінне відео за участю тривимірних моделей.

У створенні 3D-об'єктів (наприклад, мультфільму) беруть участь:

художник – визначає загальний стиль мультфільму, малює плани, нариси персонажів і т. д.;

модельєр – моделює 3D-моделі персонажів і оточення;

художник із текстур – малює текстури;

сетепер – створює для CG персонажів скелет і вчить їх рухатися;

аніматор – анімує рухи героїв мультфільму і предметів оточення;

освітлювач – виставляє світло;

оператор – виставляє камеру;

VFX – художник – створює різні спецефекти (вибухи, симуляції води, одягу, волосся і т. д.);

візуалізатор – налаштовує рендер (програму для візуалізації картинки, приміром, V-Ray), який з усіх цих 3D-даних створить реалістичну картинку;

композер – додає останні штрихи до анімації.

Розділ 1. Аналітичні 3D-об'єкти

1. Аналітична 3D-графіка

1.1. Способи побудови поверхонь

Способи побудови поверхонь можуть бути різними і залежать від специфіки сфери застосування. Спочатку поверхні можуть бути побудовані або за місцевою системою координат і скалярними параметрами, або за характерними точками, або на основі ліній. Далі ці поверхні можуть бути продовжені або зрізані, на їх основі можуть бути отримані еквідистантні поверхні. Всі згадані поверхні мають прямокутну або трикутну область визначення параметрів. На базі поверхонь із прямокутною чи трикутною областю визначення параметрів можна побудувати обмежені контурами поверхні довільної форми.

Всі поверхні є рівноправними відносно виконуваних ними функцій і

виконуваних над ними операцій. Окрім загальних функцій довільного геометричного об'єкта, поверхні мають виконувати функції:

- обчислення радіус-вектора поверхні;
- частинних похідних радіус-вектора по параметрах;
- нормалі поверхні і її частинних похідних;
- коефіцієнтів першої і другої квадратичних форм поверхні;
- надання інформації про область визначення параметрів поверхні (замкненості, граничних контурів);
- визначення приналежності заданих параметрів області визначення поверхні;
- зміни положення і орієнтації у просторі;
- модифікації форми поверхні.

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

1.2. 3D-графіка

3D-графіка – розділ комп'ютерної графіки, сукупність прийомів і інструментів, призначених для зображення об'ємних об'єктів.

Тривимірне зображення на площині відрізняється від двовимірного тим, що включає побудову геометричної проекції тривимірної моделі сцени на площину за допомогою спеціалізованих програм. Разом із тим модель може як відповідати об'єктам з реального світу, так і бути повністю абстрактною.

У тривимірній комп'ютерній графіці всі об'єкти зазвичай подаються як набір поверхонь або часток. Мінімальну поверхню називають полігоном. У якості полігону зазвичай вибирають трикутники.

У комп'ютерній графіці використовуються три види матриць:

- матриця повороту;
- матриця зрушення;
- матриця масштабування.

Будь-який полігон можна подати у вигляді набору з координат його вершин. Координати кожної вершини є вектором (x, y, z) . Помноживши вектор на відповідну матрицю, отримаємо новий вектор. Зробивши таке перетворення з всіма вершинами полігону, отримаємо новий полігон, а перетворивши всі полігони, отримаємо новий об'єкт, повернений, зрушений або масштабований відносно початкового.

Будь-яке зображення на моніторі через його площину стає растровим, оскільки монітор це – матриця. 3D-графіка існує лише у уяві, оскільки те, що зображено на моніторі – це проекція тривимірної фігури. Таким чином, візуалізація графіки буває тільки растровою і векторною, а спосіб візуалізації це – тільки растр, де від кількості пікселів залежить спосіб задавання зображення.

Спочатку об'єкт подається у вигляді набору точок або координат у тривимірному просторі. Тривимірна система координат визначається трьома осями X , Y і Z . З'єднавши вершини об'єкта лініями, отримуємо каркасну модель, названу так через те, що видимими є тільки краї поверхонь тривимірного тіла, які бувають заповнені кольором текстурами і можуть освітлюватися променями світла.

Існують такі різновиди 3D-графіки:

аналітична;

фрактальна;

полігональна;

сплайнова.

Полігональна графіка є найбільш поширеною. Це пояснюється, передовсім, високою швидкістю її оброблення. Будь-який об'єкт полігональної графіки задається набором полігонів. Полігон – це плоский багатокутник. Простим варіантом є трикутні полігони, оскільки, як відомо, через будь-які три точки у просторі можна провести площину. Кожен полігон задається набором точок. Точка задається трьома координатами – X , Y , Z . Таким чином, можна задати 3-мірний об'єкт як масив або структуру.

Аналітична графіка полягає, по суті, у тому, що об'єкти задаються аналітично, тобто формулами. Інший спосіб створення аналітичних об'єктів – это створення тіл обертання.

Фрактальна графіка заснована на понятті фрактала – самоподібності. Об'єкт називають самоподібним, коли збільшені частини об'єкта схожі на сам об'єкт і один на одного. До самоподібного класу належить місцевість. Так, зазубрений край зламаного каменю схожий на гірський хребет на горизонті. Фрактальна графіка, як і векторна, заснована на математичних обчисленнях. Базовим елементом фрактальної графіки є математична формула, у зв'язку з цим ніякі об'єкти у пам'яті комп'ютера не зберігаються і зображення будується виключно за рівняннями.

Графіка сплайна заснована на понятті сплайна. Термін “сплайн” походить від англійського *spline*. Так прийнято називати гнучка смужка стали, за допомогою якої креслярі проводять через задані точки плавні криві. У минулі часи подібний спосіб плавного обведення різних тіл (корпуса корабля, кузова автомобіля) був широко поширений у практиці машинобудування. У результаті форма тіла задавалася за допомогою набору дуже точно виготовлених сечений-плазов. Поява комп'ютерів дозволила перейти від цього, плазово-шаблонного, методу до ефективнішого способу задавання поверхні обтічного тіла. В основі цього підходу до опису поверхонь лежить використання порівняно нескладних формул, що дозволяють відтворювати вигляд виробу з великою точністю.

У процесі моделювання сплайнами найчастіше застосовується метод бікубічних раціональних B-сплайнів на нерівномірній сітці (NURBS). Вигляд поверхні при цьому визначається розташованою у просторі сіткою опорних точок. Кожній точці присвоюється коефіцієнт, величина якого визначає міру її впливу на частину поверхні, що проходить поблизу точки. Від взаємного розташування точок і величини коефіцієнтів залежать форма і гладкість поверхні.

Метод тривимірного геометричного моделювання реалізований у програмах Mathcad, AutoCAD і ArchiCAD.

1.3. Математична модель поверхонь

Поверхні, як і лінії, є математичними абстракціями, що дають уявлення про окремі властивості предметів. Так само, як і криві лінії, поверхні служать будівельним матеріалом для створення тіл. У свою чергу, поверхні можна описувати за допомогою скалярних величин, векторів, ліній та інших поверхонь. Розглянемо способи побудови різних поверхонь.

Нехай у просторі задана деяка глобальна декартова прямокутна система координат з базисними векторами e_1 , e_2 , e_3 . Подамо поверхню векторної функцією (1.1)

$$r(u, v) = \sum_{i=1}^3 r_i(u, v) \cdot e_i, \quad u, v \in \Omega \quad (1.1)$$

два скалярні аргументи u і v . Компоненти, $r_1(u, v)$, $r_2(u, v)$, $r_3(u, v)$ радіус-вектора точки поверхні $r(u, v)$ є однозначними безперервними функціями параметрів u і v . Такий опис поверхонь називається параметричним.

Область зміни параметрів поверхні Ω у загальному випадку є довільною зв'язковою двовимірною областю на площині uv (рис. 1.1), обмежену двовимірними 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 двовимірної точки $p = [uv]^T$ на параметричній площині, що належить області Ω . Область визначення параметрів поверхні має зовнішню межу і може мати вирізи всередині, але вона має бути зв'язковою, так, щоб із кожної точки області можна було потрапити у будь-яку іншу точку, не перетинаючи межі області.

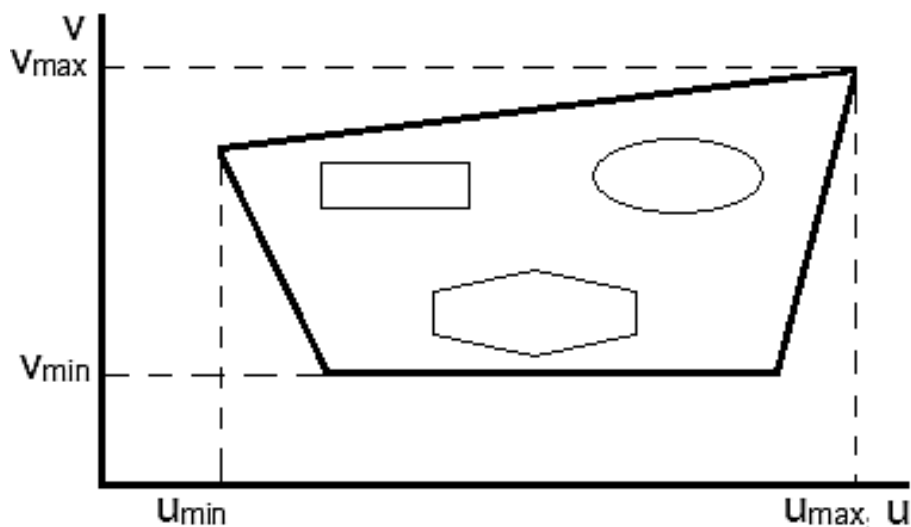


Рис. 1.1. Область визначення поверхні Ω на площині параметрів

У окремому випадку область визначення параметрів поверхні може бути прямокутником, коли параметри набувають значень у межах $u_{\min} \leq u \leq u_{\max}$, $v_{\min} \leq v \leq v_{\max}$. Прямокутні області визначення параметрів зручні для практичного застосування і часто використовуються, оскільки їх математична модель досить проста. Прямокутні області визначення параметрів мають початкові або базові поверхні, які слугують заготовками для складніших поверхонь. Приклад області параметрів поверхні наведений на рис. 1.1.

Поверхні можуть бути замкненими або розімкненими. Замкнені поверхні матимуть прямокутні області визначення параметрів. Для замкненої по u -линии поверхні $r(u \pm (u_{\max} - u_{\min}), v) = r(u, v)$; для

замкненої по v -лінії поверхні $r(u, v \pm (v_{\max} - v_{\min})) = r(u, v)$ поверхня може бути замкнена по обох координатних лініях.

Деякі поверхні можна описати за допомогою рівняння $F(r_1, r_2, r_3) = 0$, якому задовольняють компоненти радіус-вектора її точок. Такий опис поверхні називається неявним. У звичайній точці рівняння поверхні можна переписати у вигляді, дозволеному відносно 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, у процесі переходу в іншу систему координат рівняння, що описує поверхню, змінюється. Для моделювання поверхонь використовуватимемо їх параметричне подання.

Математична модель поверхні включає функціональну залежність радіус-вектора $r(u, v)$ від параметрів і двовимірну область Ω зміни параметрів u і v . Таким чином, математична модель поверхні у загальному випадку повинна містити і математичну модель області визначення своїх параметрів.

Спілкування з математичною моделлю поверхні відбувається аналогічно до спілкування з кривою. Звертаємося до функцій поверхні з деякими значеннями двох її параметрів і у якості відповіді отримуємо геометричну інформацію про поверхню у точці, що відповідає цим значенням параметрів. Якщо деяка функція поверхні допускає вихід значень параметрів за область визначення, то ця функція повинна видавати геометричну інформацію для продовження поверхні.

1.4. Аналітичні поверхні

Координати радіус-вектора деяких поверхонь можуть бути подані аналітичними функціями двох параметрів. Такі поверхні будуть називатися аналітичними. До них належать площину, поверхню тора, поверхні другого порядку та інші поверхні. Розглянемо їх параметричне представлення, а також наведемо неявне опис за допомогою рівнянь для координат їх радіус-вектора.

Опис аналітичних поверхонь зручно представляти у вигляді:

$$r(u, v) = p + x(u, v)i_x + y(u, v)i_y + z(u, v)i_z, \quad (1.2)$$

де

p – деяка точка прив'язки характерної точки поверхні;

i_x, i_y, i_z – взаємно ортогональні вектори одиничної довжини, $x(u, v)$, $y(u, v)$;

$z(u, v)$ – координатні функції.

Точка p і вектори i_x, i_y, i_z представляють місцеву декартову систему координат, а функції $x(u, v)$, $y(u, v)$, $z(u, v)$ можна розглядати як координати радіус-вектора поверхні у місцевій системі координат. Орти можуть бути подані у вигляді розкладання за базисом e_1, e_2, e_3 глобальної декартової системи координат (1.3).

$$\begin{aligned}i_x &= a_{11}e_1 + a_{12}e_2 + a_{13}e_3, \\i_y &= a_{21}e_1 + a_{22}e_2 + a_{23}e_3, \\i_z &= a_{31}e_1 + a_{32}e_2 + a_{33}e_3.\end{aligned}\tag{1.3}$$

Точка p також може бути подана у вигляді розкладання за базисом e_1, e_2, e_3 глобальної декартової системи координат (1.4):

$$p = p_1e_1 + p_2e_2 + p_3e_3.\tag{1.4}$$

Положення і орієнтація місцевої декартової системи координат становлять частину структури цих аналітичних поверхонь. Перетворення з місцевої системи координат у глобальну систему координат і назад. Для опису аналітичних поверхонь використовуються параметричні залежності, що дають однозначні функції радіус-вектора. Всі вони описуються за допомогою місцевої декартової системи координат, в якій запис має найбільш простий вигляд, називається канонічним.

1.4.1. Площина

Простою поверхнею є площина. Вона може бути описана точкою p і двома не колінеарними векторами i_1, i_2 . Радіус-вектор площини описується залежністю (1.5):

$$r(u, v) = p + ui_1 + vi_2, \quad -\infty < u < \infty, \quad -\infty < v < \infty.\tag{1.5}$$

Довжина векторів i_1, i_2 і їхня взаємна орієнтація у загальному випадку може бути довільною. Параметри u і v є координатами місцевої декартової системи координат, пов'язаної з площиною. Площина може

бути подана як одна з координатних площин певної місцевої декартової системи координат. Вона використовується у якості конструктивної площини для побудови двовимірних (плоских) геометричних об'єктів. У якості конструктивного елемента площина використовується у сукупності з двовимірними контурами, що описують область Ω визначення її параметрів u, v . Така плоска поверхня називається обмеженою контурами площини. Радіус-вектор обмеженої контурами площини описується залежністю (1.6):

$$r(u,v) = p + ui_1 + vi_2, \quad u, v \in \Omega, \quad (1.6)$$

де область визначення параметрів складається з сукупності двовимірних контурів.

Три взаємно пересічні площини подані на рис. 1.2.

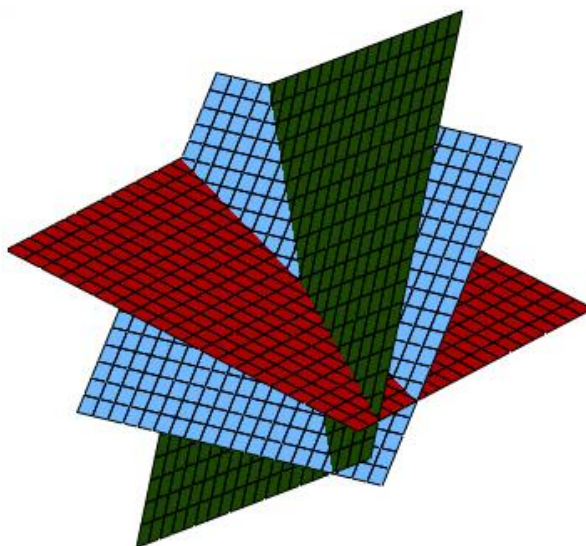


Рис. 1.2. Площини

1.4.2. Сфера

Сферичну поверхню або сферу можна описати, задавши її радіус r , положення центра p і три взаємно ортогональних вектори одиничної довжини i_x, i_y, i_z , визначальне положення і орієнтацію місцевої декартової системи координат. Радіус-вектор сфери визначається рівністю (1.7):

$$r(u, v) = p + r \cdot \cos(v) \cdot \cos(u) i_x + r \cdot \cos(v) \cdot \sin(u) \cdot i_y + r \cdot \sin(v) \cdot i_z, \quad (1.7)$$

$$0 \leq u \leq 2 \cdot \pi, \quad \frac{-\pi}{2} \leq v \leq \frac{\pi}{2}.$$

Сфера замкнена за параметром u , але не замкнена за параметром v , хоча має видимість замкненої з усіх боків поверхні. Замкненість за параметром u характеризується тим, що для параметра v це не справедливо. Скалярні функції сфери (1.8):

$$\begin{aligned} x(u, v) &= r \cdot \cos(v) \cdot \cos(u), \\ y(u, v) &= r \cdot \cos(v) \cdot \sin(u), \\ z(u, v) &= r \cdot \sin(v) \end{aligned} \quad (1.8)$$

пов'язані рівнянням (1.9):

$$x^2 + y^2 + z^2 = r^2. \quad (1.9)$$

Виведення скалярних функцій сфери виходить з рис. 1.3.

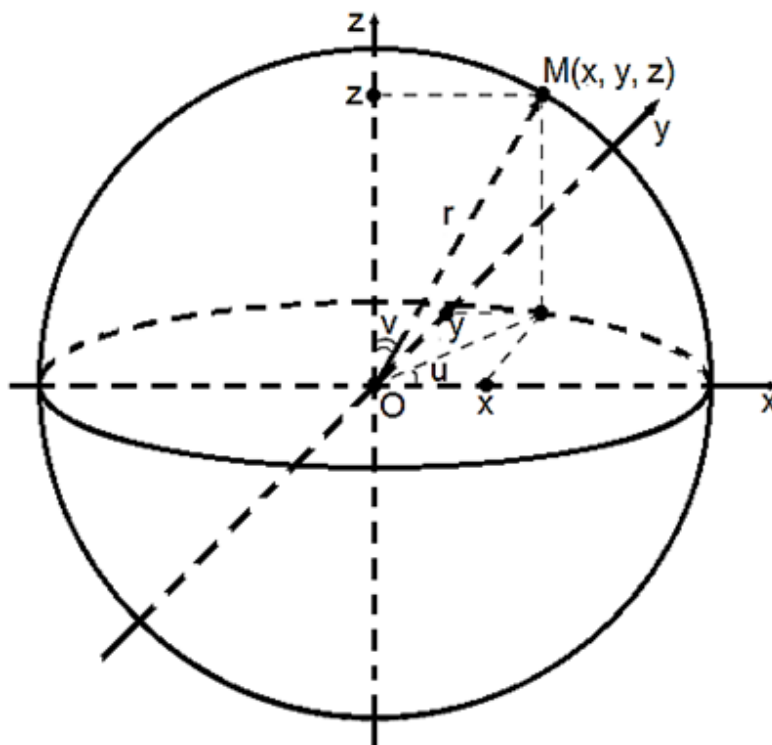


Рис. 1.3. Виведення скалярних функцій сфери

Точка r є центром сфери (рис. 1.4).

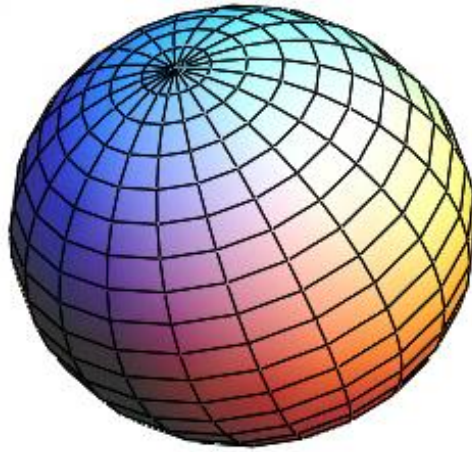


Рис. 1.4. Сфера

Всі точки сфери віддалені від центра на однакову відстань, рівне r .

1.4.3. Еліпсоїд

Еліпсоїд може бути описаний формулою (1.10):

$$\begin{aligned} r(u, v) &= p + a \cdot \cos(v) \cdot \cos(u) i_x + b \cdot \cos(v) \cdot \sin(u) \cdot i_y + c \cdot \sin(v) \cdot i_z, \\ 0 \leq u &\leq 2 \cdot \pi, \quad \frac{-\pi}{2} \leq v \leq \frac{\pi}{2}. \end{aligned} \quad (1.10)$$

Еліпсоїд, як і сфера, є замкненою по u і не замкненою по v поверхні. Перерізи еліпсоїда площинами, ортогональними векторами i_x , i_y , i_z , є еліпсами.

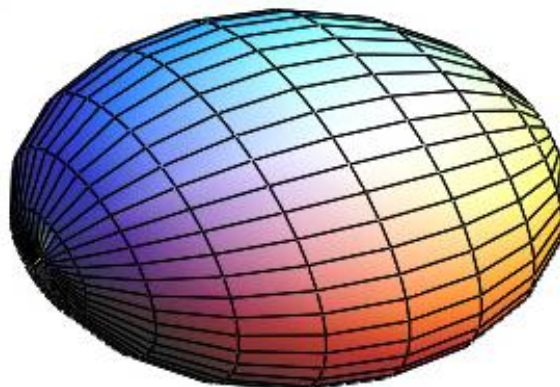


Рис. 1.5. Еліпсоїд

Скалярні функції еліпсоїда (1.11):

$$\begin{aligned}
 x(u, v) &= a \cdot \cos(v) \cdot \cos(u), \\
 y(u, v) &= b \cdot \cos(v) \cdot \sin(u), \\
 z(u, v) &= c \cdot \sin(v)
 \end{aligned}
 \tag{1.11}$$

пов'язані рівнянням (1.12):

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1.
 \tag{1.12}$$

1.4.4. Однопорожнинний гіперболоїд

Радіус-вектор однопорожнинного гіперболоїда (рис. 1.6) може бути описаний формулою (1.13):

$$\begin{aligned}
 \mathbf{r}(u, v) &= \mathbf{p} + a \cdot \operatorname{ch}(v) \cdot \cos(u) \mathbf{i}_x + b \cdot \operatorname{ch}(v) \cdot \sin(u) \mathbf{i}_y + c \cdot \operatorname{sh}(v) \mathbf{i}_z, \\
 0 \leq u &\leq 2 \cdot \pi, \quad v_{\min} \leq v \leq v_{\max},
 \end{aligned}
 \tag{1.13}$$

де – півосі гіперболоїда.

Однопорожнинний гіперболоїд є замкнутим по u і не замкнутим по v поверхнею.

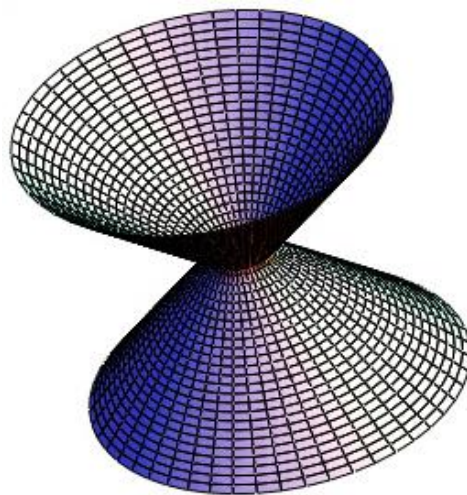


Рис. 1.6. Однопорожнинний гіперболоїд

Скалярні функції однопорожнинного гіперболоїда (1.14):

$$\begin{aligned}
 x(u, v) &= a \cdot \operatorname{ch}(v) \cdot \cos(u), \\
 y(u, v) &= b \cdot \operatorname{ch}(v) \cdot \sin(u), \\
 z(u, v) &= c \cdot \operatorname{sh}(v)
 \end{aligned}
 \tag{1.14}$$

пов'язані рівнянням (1.15):

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - \left(\frac{z}{c}\right)^2 = 1.
 \tag{1.15}$$

Перерізи однопорожнинного гіперболоїда площинами, ортогональними векторам i_x , i_y , є гіперболами, а перерізи однопорожнинного гіперболоїда площинами, ортогональними вектору i_z , є еліпсами.

1.4.5. Двопорожнинний гіперболоїд

Радіус-вектор однієї з двох частин двопорожнинного гіперболоїда (рис. 1.7) може бути описаний формулою (1.16):

$$\begin{aligned}
 r(u, v) &= p + a \cdot \operatorname{sh}(v) \cdot \cos(u) i_x + b \cdot \operatorname{sh}(v) \cdot \sin(u) \cdot i_y + c \cdot \operatorname{ch}(v) \cdot i_z, \\
 0 \leq u &\leq 2 \cdot \pi, \quad v_{\min} \leq v \leq v_{\max},
 \end{aligned}
 \tag{1.16}$$

де a , b , c – півосі гіперболоїда.

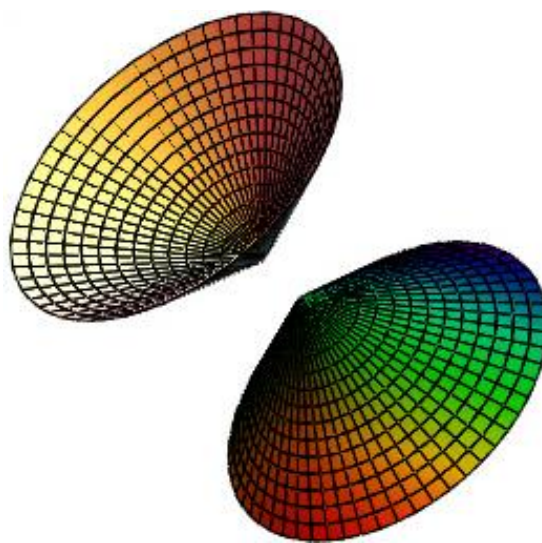


Рис. 1.7. Двопорожнинний гіперболоїд

Друга частина гіперболоїда описується цим же виразом, але орт i_z повинен мати протилежний напрямок або повинен бути змінений знак на протилежний. Двопорожнинний гіперболоїд є замкненим по u і не замкненим по v поверхнею.

Скалярні функції двопорожнинного гіперболоїда (1.17):

$$\begin{aligned}x(u, v) &= a \cdot \text{sh}(v) \cdot \cos(u), \\y(u, v) &= b \cdot \text{sh}(v) \cdot \sin(u), \\z(u, v) &= c \cdot \text{ch}(v)\end{aligned}\tag{1.17}$$

пов'язані рівнянням (1.18):

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - \left(\frac{z}{c}\right)^2 = -1.\tag{1.18}$$

Перерізи двопорожнинного гіперболоїда площинами, ортогональними векторам i_x, i_y , є гіперболами, а перерізу гіперболоїда площинами, ортогональними вектору i_z , є еліпсами.

1.4.6. Еліптичний параболоїд

Еліптичний параболоїд може бути описаний векторною функцією (1.19)

$$\begin{aligned}r(u, v) &:= p + a \cdot v \cdot \cos(u) i_x + b \cdot v \cdot \sin(u) \cdot i_y + v^2 \cdot i_z, \\0 \leq u &\leq 2 \cdot \pi, \quad v_{\min} \leq v \leq v_{\max},\end{aligned}\tag{1.19}$$

де a, b – півосі еліпса, що виходить як координатні лінії $v = \text{const}$ поверхні. Еліптичний параболоїд є замкнутим по u і не замкнутим по v поверхнею. Скалярні функції еліптичного параболоїда (1.20)

$$\begin{aligned}x(u, v) &= a \cdot v \cdot \cos(u), \\y(u, v) &= b \cdot v \cdot \sin(u), \\z(u, v) &= v^2,\end{aligned}\tag{1.20}$$

пов'язані рівнянням (1.21)

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = z. \quad (1.21)$$

Перерізу параболоїда площинами, що проходять через вісь i_z , є параболами, а перерізу параболоїда площинами, ортогональними вектору i_z , є еліпсами. Еліптичний параболоїд показаний на рис. 1.8.

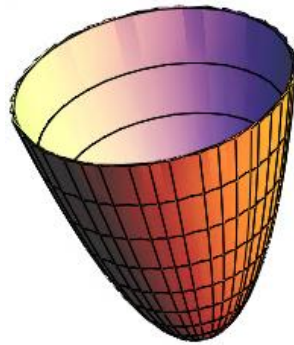


Рис. 1.8. Еліптичний параболоїд

Одну і ту ж поверхню можна параметризувати різними способами. Еліптичний параболоїд може бути описаний іншою векторною функцією (1.22):

$$\begin{aligned} \mathbf{r}(u, v) &= \mathbf{p} + a \cdot u \mathbf{i}_x + b \cdot v \mathbf{i}_y + (u^2 + v^2) \cdot \mathbf{i}_z, \\ u_{\min} &\leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}. \end{aligned} \quad (1.22)$$

Еліптичний параболоїд є не замкненої за обох параметрах поверхнею.

1.4.7. Гіперболічний параболоїд

Гіперболічний параболоїд (рис. 1.9) може бути описаний векторною функцією (1.23):

$$\begin{aligned} \mathbf{r}(u, v) &= \mathbf{p} + a \cdot u \mathbf{i}_x + b \cdot v \mathbf{i}_y + (u^2 - v^2) \cdot \mathbf{i}_z, \\ u_{\min} &\leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \end{aligned} \quad (1.23)$$

де a, b – півосі гіперболи.

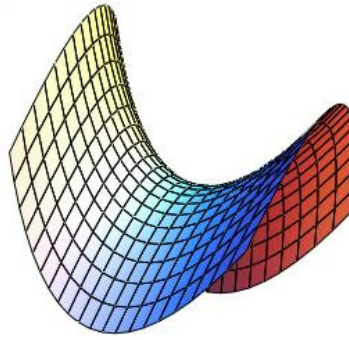


Рис. 1.9. Гіперболічний параболоїд

Гіперболічний параболоїд є не замкнутою по обох параметрах поверхнею. Якщо гіперболічний параболоїд по аналогії з еліптичним параболоїдом подати векторною функцією (1.24)

$$\begin{aligned} \mathbf{r}(u, v) &= \mathbf{p} + \mathbf{a} \cdot v \cdot \operatorname{ch}(u) \mathbf{i}_x + \mathbf{b} \cdot v \cdot \operatorname{sh}(u) \cdot \mathbf{i}_y + v^2 \cdot \mathbf{i}_z, \\ u_{\min} &\leq u \leq u_{\max}, \quad v_{\min} \leq v \leq v_{\max}, \end{aligned} \quad (1.24)$$

то вона буде описувати тільки частину поверхні.

Скалярні функції $x(u, v)$, $y(u, v)$, $z(u, v)$ обох подань гіперболічного параболоїда пов'язані рівнянням (1.25):

$$\left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = z. \quad (1.25)$$

Формули описують обмежені гіперболоїди і параболоїди, усічені параметрами v_{\min} і v_{\max} або u_{\min} і u_{\max} . Якщо в формулах (1.21, 1.25) $a = b$, то будуть отримані відповідні поверхні обертання навколо місцевої осі iz . Окремими випадками поверхні другого порядку є циліндрична і конічна поверхні.

1.5.8. Циліндр

Еліптичний циліндр (рис. 1.10) може бути описаний формулою (1.26):

$$\begin{aligned} \mathbf{r}(u, v) &= \mathbf{p} + \mathbf{a} \cdot \cos(u) \mathbf{i}_x + \mathbf{b} \cdot \sin(u) \cdot \mathbf{i}_y + \mathbf{h} \cdot v \cdot \mathbf{i}_z, \\ 0 &\leq u \leq 2 \cdot \pi, \quad 0 \leq v \leq 1, \end{aligned} \quad (1.26)$$

де
 a, b – півосі еліпса, що є поперечним перерізом циліндричної поверхні;
 h – довжина циліндра.

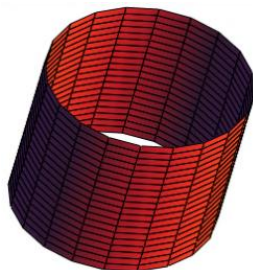


Рис. 1.10. Циліндрична поверхня

Якщо $a = b$ (1.26), то отримують круговий циліндр. Циліндрична поверхня є замкненою по параметру u і усіченою по параметру v .

Конус. Кругова (рис. 1.11) конічна поверхня може бути описана формулою (1.27):

$$r(u, v) = p + (r + h \cdot v \cdot \operatorname{tg}(\gamma)) \cdot (\cos(u) i_x + \sin(u) \cdot i_y) + h \cdot v \cdot i_z, \quad (1.27)$$

$$0 \leq u \leq 2 \cdot \pi, \quad 0 \leq v \leq 1,$$

де

r – радіус однієї з підстав конуса;

h – довжина конуса;

γ – кут між твірною і віссю конуса.

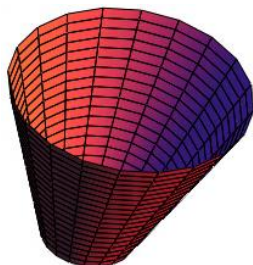


Рис. 1.11. Конічна поверхня

Конічна поверхня, як і циліндрична, є замкненою по параметру u і всіченою по параметру v .

1.4.9. Тороїдальна поверхня

Поверхня тора (рис. 1.12) може бути описана положенням центра p , трьома взаємно ортогональними векторами одиничної довжини i_x, i_y, i_z , що визначають положення і орієнтацію місцевої декартової системи координат, головним радіусом R і малим радіусом r (1.28):

$$r(u, v) = p + (R + r \cdot \cos(v)) \cdot \cos(u) i_x + (R + r \cdot \cos(v)) \cdot \sin(u) \cdot i_y + r \cdot \sin(v) \cdot i_z, \quad (1.28)$$
$$0 \leq u \leq 2 \cdot \pi, \quad -\pi \leq v \leq \pi.$$

Скалярні функції тора (1.29):

$$\begin{aligned} x(u, v) &= (R + r \cdot \cos(v)) \cdot \cos(u), \\ y(u, v) &= (R + r \cdot \cos(v)) \cdot \sin(u), \\ z(u, v) &= r \cdot \sin(v) \end{aligned} \quad (1.29)$$

пов'язані рівнянням (1.30):

$$\left(\sqrt{x^2 + y^2} - R\right)^2 + z^2 = r^2. \quad (1.30)$$

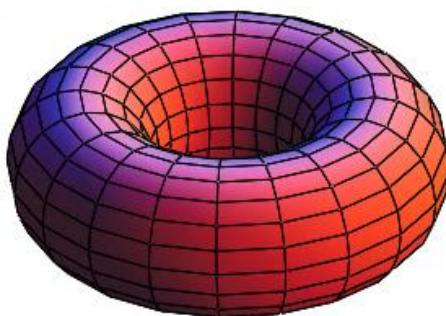


Рис. 1.12. Відкритий тор

Область зміни параметра v тороїдальної поверхні записана у припущенні, що $r < R$. Якщо $r > R$, то для того, щоб тор не перетинав сам себе, треба зменшити область зміни параметра v (1.31)

$$-\pi + v_0 \leq v \leq \pi - v_0, \quad (1.31)$$

де

$$v_0 = \arccos(R/r).$$

Коли $R = 0$ $v_0 = \pi/2$ і тор перетвориться на сферу радіусу r . Тороїдальна поверхня називається відкритою, якщо $r < R$, і називається закритою, якщо $r > R$. Відкритий і закритий тори показані на рис. 1.12 і 1.13.

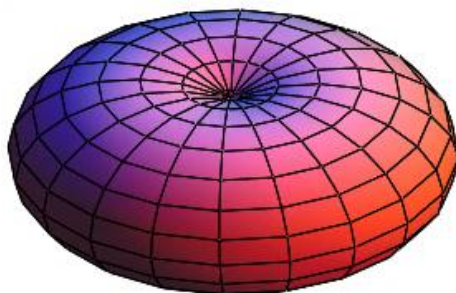


Рис. 1.12. Закритий тор

Існують ще ряд поверхонь, які можна описати аналітичними параметричними залежностями. Похідні радіус-вектора аналітичних поверхонь можна знайти диференціюванням компонент радіус-вектора за параметрами.

1.5. Приклади 3D-графіків у Mathcad

1.5.1. Побудова поверхонь за матрицею аплікату

Класичним способом побудови графіку поверхні, заданої функцією від двох змінних $z = f(x, y)$, є заповнення матриці значеннями цієї функції. Разом із тим рядки і стовпці матриці інтерпретуються як абсциси і ординати.

Приклад. Необхідно визначити функцію двох змінних $f(u, v) = \cos(u \cdot v)$. Для визначення функції задамо кількість ліній для побудови графіку і масштабу $N = 15$ і $M = 4$ відповідно, а також діапазон зміни індексів $i = 0 \dots N$, $j = 0 \dots M$. Далі обчислюємо масиви абсцис, ординат і аплікату відповідно (1.32).

$$u_i = \frac{i - \frac{N}{2}}{M}, \quad v_j = \frac{j - \frac{N}{2}}{M}, \quad S_{i,j} = f(u_i, v_j). \quad (1.32)$$

У шаблон тривимірного графіка вводимо назву масиву аплікату (рис. 1.14).

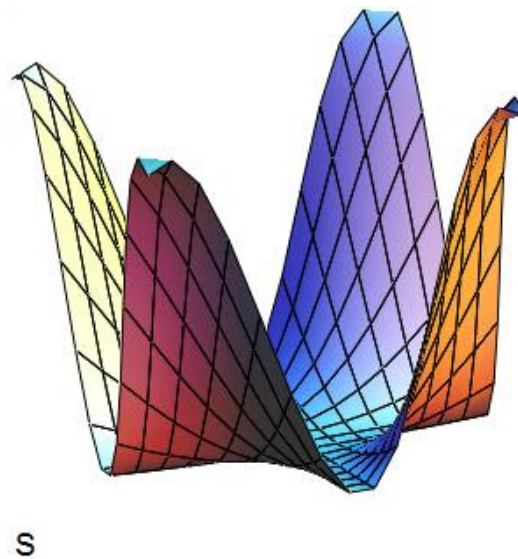


Рис. 1.14. Графік функції $f(u, v) = \cos(u \cdot v)$

1.5.2. Побудова поверхонь за допомогою функції *CreateMesh*

Функція *CreateMesh* належить до категорії *Vector and matrix* (вектори і матриці), оскільки результатом роботи функції буде матриця координат. Формат виклику функції:

`CreateMesh(S, u1, u2, v1, v2, ugrid, vgrid, mesh).`

Параметри функції *CreateMesh*:

S – вигляд функції (може бути або формула, або тривимірний вектор, який задає кожену координату в параметричній формі, або три окремі функції, які задають координати в параметричному вигляді);;

u1 – нижня межа змінної u;

u2 – верхня межа змінної u;

v1 – нижня межа змінної v;

v2 – верхня межа змінної v;

ugrid – кількість точок змінної u;

vgrid – кількість точок змінної v;

Mesh – кількість ліній у сітці функції.

Приклад. Необхідно Побудуйте 3D-графік функції $f(u, v) = u^2 + v^2$ у середовищі Mathcad (рис. 1.15).

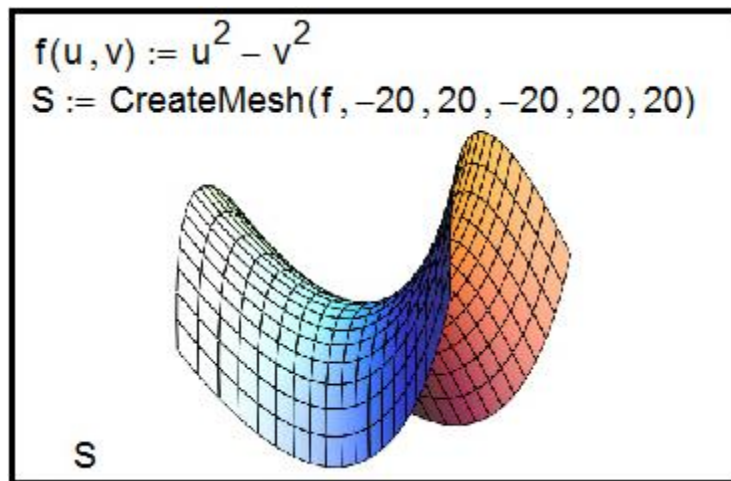


Рис. 1.15. Лістинг Mathcad

1.5.3. Побудова поверхонь обертання

Для побудови поверхонь обертання у Mathcad зручно використовувати функцію CreateMesh.

Параметри функції CreateMesh:

X, Y, Z – матриці значень для кожної координати;

-5 – нижня межа змінної u;

5 – верхня межа змінної u;

0 – нижня межа змінної v;

2π – верхня межа змінної v;

30 – кількість ліній у сітці графіку.

Приклад: розглянемо гіперболу $v^2 - u^2 = 1$. У результаті обертання цієї гіперболи навколо осі Ox виходить однопорожнинний гіперboloїд. Побудуємо цю поверхню обертання (рис. 1.16).

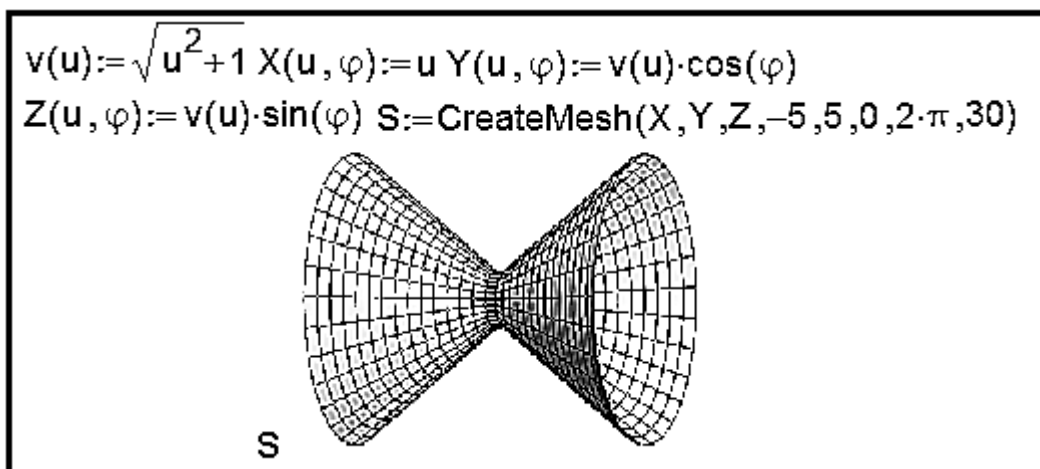


Рис. 1.16. Лістинг Mathcad

Розглянемо ще один приклад побудови тривимірної поверхні, утвореної обертанням одновимірної функції відносно осі x . У результаті виходять так звані фігури обертання. Існують загальні принципи параметричного задавання таких поверхонь. Якщо задана функція $f(x)$, то з її допомогою задаються три параметричні функції $X(u, v) = u$, $Y(u, v) = f(u) \cdot \cos(v)$, $Z(u, v) = f(u) \cdot \sin(v)$ і потім у шаблон графіка поверхні заносяться три матриці (X, Y, Z) . На рис. 1.17. показана процедура побудови фігури, утвореної обертанням функції $f(x) = x^2 \sin x$ відносно осі x .

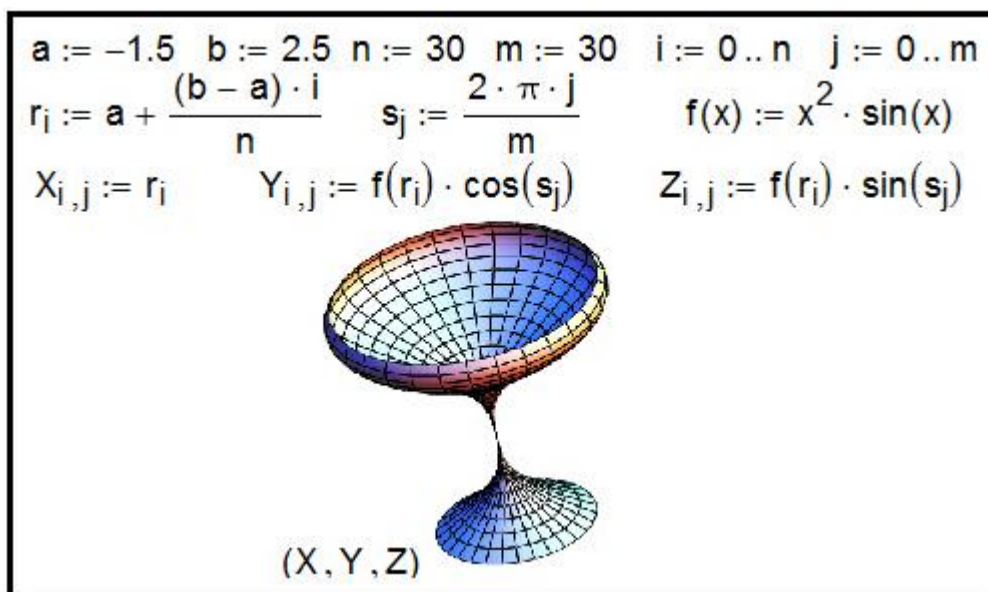


Рис. 1.17. Лістинг Mathcad

1.5.4. Форматування тривимірних графіків

На усіх приведених рисунках цього розділу графіки мають досить примітивний вигляд. Розглянемо, які засоби має Mathcad для модифікації вже створеної поверхні.

Вікно форматування тривимірного графіка 3D Plot Format викликається подвійним клацанням по області побудови графіка або командою основного меню Format > Graph > 3D Plot ... за виділеному графіку. Параметри, що впливають на зовнішній вигляд графіка, згруповані за принципом дії на декількох вкладках:

General – загальні параметри зображення;

Axes – параметри координатних осей (тип, товщина, колір, кількість відміток, їх нумерація, масштаб і под.);

Appearance – параметри відображення графіку (колір лінії, тип точок, використовуваних під час побудови);

Backplanes – параметри граней;

Lighting – умови освітлення і вибір схеми освітлення;

Special – спеціальні параметри (контурні лінії і т.д.);

Title – створення заголовків і їх параметри;

Advanced – додаткові параметри (перспектива, світлові ефекти, якість друку і под.);

QuickPlot Data – параметри швидкої побудови графіку за даними.

Розглянемо вкладку встановлення загальних параметрів General.

У групі параметрів View (вид) можна задати такі встановлення:

Rotation – кут повороту (від 0 до 360 градусів);

Tilt – кут нахилу (від 0 до 180);

Twist – кут обертання (від 0 до 360 градусів);

Zoom – відносний розмір (одиниця за замовчуванням).

Група Axes Style (стиль осей) дозволяє встановити такі параметри:

Perimeter – осі по периметру;

Corner – осі у кутку;

None – немає осей;

Equal Scales – задає рівний масштаб по усіх осях.

Група прапорців Framers задає параметри обрамлення:

Show Border – показує рамку навколо рисунка;

Show Box – не виводить осі.

Остання група параметрів вкладки View, Plot 1 Display As: дозволяє змінити тип поверхні, що дає можливість не виконувати побудову графіків різного типу для однієї й тієї ж функції кілька разів.

На вкладці Axes (осі) можна встановити параметри координатних осей (XAxes, YAxes, ZAxes) тривимірного графіка.

Параметри для форматування координатної сітки розташовані в групі Grids:

Draw Lines – виведення ліній сітки;

Draw Tics – виведення поділок на осях;

Auto Grid – автоматичний вибір кількості ліній;

Line Color – вибір кольору ліній;

Number – кількість поділок;

Line Weight – ширина лінії сітки.

Формат координатних осей встановлює група параметрів Axis Format:

Show Number – оцифрування осей;

Axis Color – колір осей;

Axis Weight – ширина лінії осей.

Межі зміни координат задаються у групі Axis Scale:

Auto Scale – масштаб встановлюється автоматично;

Minimum Value – мінімальне значення по осі;

Maximum Value – максимальне значення по осі;

Останні два параметри доступні тільки за вимкненого Auto Scale.

Зовнішній вигляд самої поверхні змінюється за допомогою вкладки Appearance. Розглянемо основні групи параметрів цього діалогового вікна:

Fill Options – забарвлення поверхонь і контурних ліній;

Line Options – відображення ліній і їх забарвлення;

Point Options – відображення різними символами і їх фарбування.

Перемикачі, наявні у кожній групі, призначені для вибору схеми забарвлення (Colormap – колірна карта, Solid Color – основний колір).

Вкладка Title, призначена для введення заголовка у графічну область (поле Graph Title) і встановлення її місця (вгорі, внизу, приховати) розташування.

Вкладка Lighting дозволяє встановити ефект освітлення поверхні. Параметри цієї вкладки активізуються після встановлення прапорця Enable Lighting. Параметри модифікації граней (XY – Backplanes, YZ – Backplanes, XZ – Backplanes) можна встановити на вкладках Backplanes, які містять два основні прапорці:

Fill Backplanes – заливка кольором відповідної грані;

Backplanes Border – встановлення обрамлення грані.

Групи параметрів Grid і Subgrid призначені для задавання сітки на гранях.

За допомогою встановлення параметрів на вкладці Special можна створити спеціальні ефекти. На вкладці розташовано чотири групи параметрів:

Contour Options – параметри контурних зображень;

Bar Plot Layout – розмітка стовпчикowego графіку (гістограми);

Interpolated Mesh – інтерполяція поверхні за заданою кількістю ліній у рядках і стовпцях, за умовчуванням кількість ліній – 21;

Connectivity – зв'язок функціонального забарвлення з параметрами, що змінюються.

Встановлення спецефектів можливе для певних типів графіків. Додаткові параметри знаходяться на вкладці Advanced. Група Advanced View Options містить встановлення:

Enable Fog – ефект серпанку;

Perspective – відображення у перспективі;

Vertical Scale – масштаб по вертикалі;

Viewing Distance – відстань, з якої розглядається фігура. Група Printing містить два прапорці, які задають друк зі звичайною і підвищеною якістю, а група Colormap дозволяє встановити колірну карту за збільшенням значень координат точок уздовж осей.

Встановити основні параметри для швидкої побудови 3D-графіків можна на вкладці Quick Plot Data, яка містить три групи параметрів:

Range 1 – межі за першим параметром (start – початкове значення, end – кінцеве значення, # of Grids – кількість ліній, з яких складається поверхня);

Range 2 – межі за другому параметру;

Coordinates System – вибір однієї з трьох систем координат (прямокутної, сферичної, циліндричної).

1.5.5. Приклади для самотійного вирозв'язання

1. Побудуйте тривимірну поверхню, що відповідає функції $F(x) = x \cdot \exp(-x^2/2 - y^2/4)$, у разі зміни незалежних змінних x і y у межах від -4 до +4. Кількість точок розбиття області $N = 40$.

2. Побудуйте тривимірну поверхню, що відповідає функції $F(\sin(x) + \cos(y))$, x і y змінюються від 0 до 2π з кроком $\pi/10$. Кількість точок розбиття області $N = 20$. Поверхню подати у вигляді сітки (Wire frame).

3. Побудуйте тривимірну поверхню, задану параметричними функціями $X(u, v) = a \cdot \sin(u) \cdot \cos(v)$, $Y(u, v) = b \cdot \sin(u) \cdot \sin(v)$, $Z(u, v) = c \cdot \cos(u)$, $a = 1$, $b = 2$, $c = 3$, u і v міняються від 0 до 2π . Кількість точок розбиття області $N = 20$. Поверхню подати у вигляді сітки з видаленням невидимих ліній Hide Lines.

4. Побудуйте поверхню другого порядку, задану параметрично функціями $X(u, v) = a \cdot u \cdot \cos(v)$, $Y(u, v) = b \cdot u \cdot \sin(v)$, $Z(u, v) = 1/2 \cdot u$, $a = 2$, $b = 4$, u і v міняються від 0 до 2π . Кількість точок розбиття області $N = 20$. Поверхню подати у вигляді сітки з функціональним зафарбовуванням (Fill Surface).

5. Побудуйте фігуру обертання, що задається функцією $f(x) = \sin(x)/x$, величина x змінюється у межах $[1, 10]$. Кількість точок розбиття $N = 20$. Обертання відбувається відносно осі x .

6. Побудуйте фігуру обертання, що задається функцією $f(x) = \sin(x)/x$, величина x змінюється у межах $[0, 10]$. Кількість точок розбиття $N = 20$. Обертання відбувається відносно осі y . Поверхню подати з функціональним зафарбовуванням без сітки.

7. Побудуйте тривимірну поверхню, задану параметрично функціями $X(u, v) = \cos(u) \cdot \cos(v)$, $Y(\cos(u) + 5) \cdot \sin(v)$, $Z(u, v) = \cos(u)$ з використанням графічної функції CreateMesh. Кількість точок розбиття по осях дорівнює 40.

Контрольні запитання

1. Аналітична графіка.
2. Класифікація способів побудови 3D-графіків.
3. Математична модель поверхні.
4. Поверхні другого порядку.
5. 3D-графіка у Mathcad.
6. Алгоритм побудови поверхні у Mathcad.
7. Матриця вузлових точок.
8. Алгоритм побудови поверхонь обертання.
9. Засоби побудови 3D-графіків у Mathcad.

2. Фрактальна 3D-графіка

2.1. Фрактали

Слово “фрактал” з'явилося завдяки геніальному вченому Бенуа Мандельброту. Він увів цей термін у сімдесятих роках минулого століття, запозичивши слово *fractus* з латині, де воно буквально означає ламаний або роздрібнюваний. Сьогодні під словом “фрактал” найчастіше прийнято мати на увазі графічне зображення структури, якого у більшому масштабі подібна сама собі.

Французький математик Лейбніц побудував фрактал на основі простої формули, проітерированной циклом зворотного зв'язку. Це означає, що конкретне значення числа знаходиться на основі попереднього.

Мандельброт був першим, хто використовував комп'ютер для прорахунку класичного фрактала. Обробивши послідовність, що складається з великої кількості значень, Бенуа переніс результати на графік (рис. 2.1).

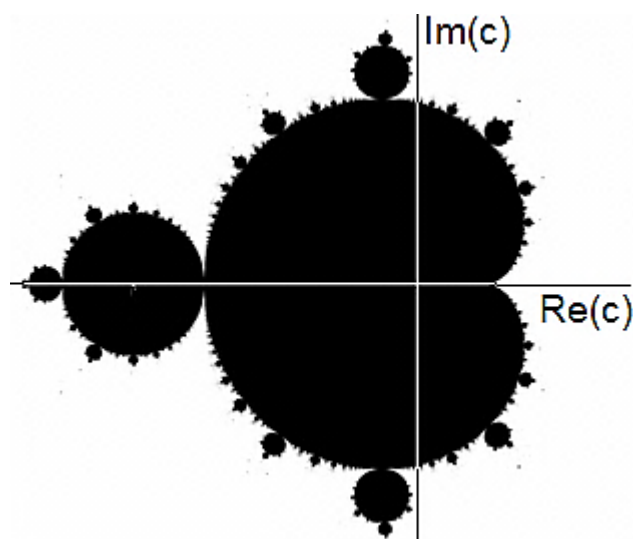


Рис. 2.1. Фрактал Мандельброта

Згодом це зображення було розфарбоване і стало одним із найпопулярніших зображень, які тільки були створені людиною (рис. 2.2).

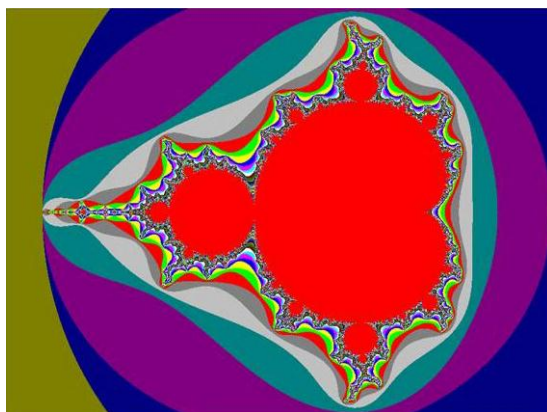


Рис. 2.2. **Зображення фрактала Мандельброта**

2.2. Сфера застосування фракталів

Теорія фракталів знайшла практичне застосування в художників.

Засновник студії "Pixar" аніматор Лорен Карпентер вивчив принципи фрактальної геометрії і знайшов спосіб реалізувати її у комп'ютерній графіці. Принцип, який використовував Лорен для досягнення мети, полягав у тому, щоб поділяти більшу геометричну фігуру на дрібні елементи, а ті, у свою чергу, ділити на аналогічні фігури меншого розміру (рис. 2.3).

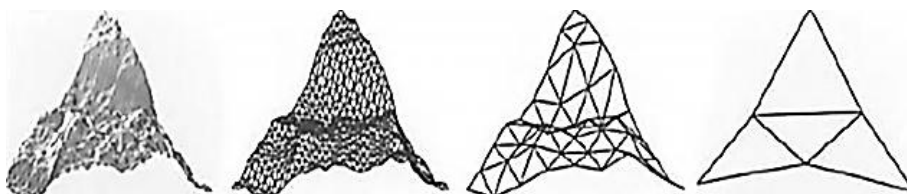


Рис. 2.3. **Принцип поділу великої геометричної фігури**

Використовуючи більші трикутники, Карпентер дробив їх на чотири дрібних і потім повторював цю процедуру знову, поки у нього не виходив реалістичний гірський ландшафт. Таким чином, йому вдалося стати першим художником, який застосував у комп'ютерній графіці фрактальний алгоритм для побудови зображень.

Нині усі популярні застосування для створення 3D-зображень використовують аналогічний принцип генерування природних об'єктів.

Фрактальні зображення застосовуються для дизайну інтер'єру, прикрашання одягу, аерографії та в інших областях.

Для дизайну інтер'єру використовуються різні картини і постери,

які кардинально перетворюють приміщення (рис. 2.4).



Рис. 2.4. Дизайн інтер'єру

Картини, створені на основі фрактальної графіки, дозволяють візуально збільшити простір кімнати, інші надають відтінки затишку, треті містять певну інформацію.

Друк фрактальної графіки на одязі дозволяє якісно й ефектно виразити свою індивідуальність, реалізувати і привнести до світу свій смак і своє бачення світу. Фрактальну графіку друкують на футболках, на толстовках, на бейсболках, сумках, навіть на шкарпетках і парасольках (рис. 2.5).

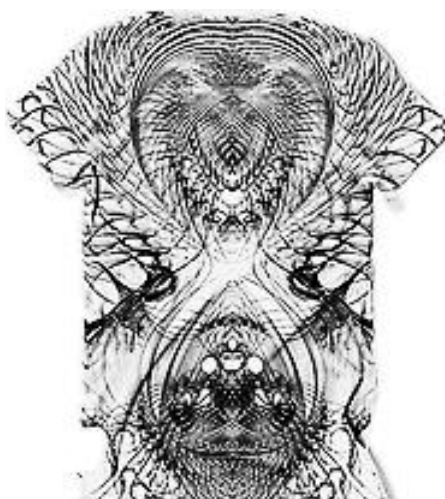


Рис. 2.5. Фрактальна графіка на футболці

Фрактальні зображення, нанесені за допомогою аерографії, прикрашають автомобілі (рис. 2.6).



Рис. 2.6. Фрактальне зображення на кузові автомобіля

2.3. Математичний опис фракталів

Фрактали дозволяють описувати цілі класи зображень, для детального опису яких потрібно відносно мало пам'яті. Програмні засоби для роботи з фрактальною графікою призначені для автоматичної генерації зображень шляхом математичних розрахунків. Створення фрактальної художньої композиції полягає не у малюванні або оформленні, а у програмуванні. Таким чином, для створення конкурентоспроможних і комерційних фракталів потрібно знання чисельних методів.

Математичною основою фрактальної графіки є фрактальна геометрія. У сучасній науково-технічній літературі розглядається велика кількість різних методів синтезу фрактальних зображень. Для того щоб розібратися у них треба їх класифікувати. Існують класифікації, наприклад, рукотворні і природні. До рукотворних належать ті фрактали, які були придумані вченими, вони за будь-якого масштабу мають фрактальні властивості. На природні фрактали накладається обмеження на область існування, тобто максимальний і мінімальний розміри, за яких в об'єкта спостерігаються фрактальні властивості. Усе рукотворне множина фракталів поділяється на геометричні, алгебраїчні і стохастичні.

Геометричні фрактали – найнаочніші, тому що у них відразу видно самоподібність. Прикладом таких кривих слугує трикутник Серпінського. Побудова трикутника Серпінського заснована на методі IFS [1,3]. Система ітераційних функцій (IFS) – це сукупність стискальних афінних перетворень. Як відомо, афінне перетворення включає масштабування, поворот і паралельне перенесення. Афінне перетворення вважається таким, що стискає, якщо коефіцієнт масштабування менший від одиниці.

Сутність методу IFS можна пояснити за допомогою перетворень. Помістимо прямокутний трикутник з довгим катетом, що дорівнює одиниці, на комплексну площину Z (рис. 2.7).

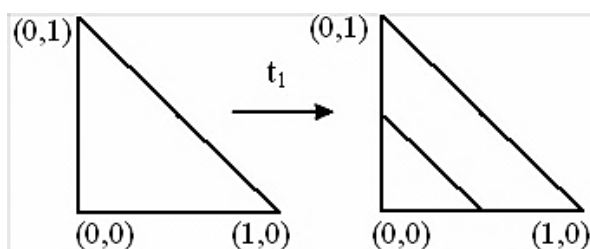


Рис. 2.7. Стискує перетворення по дійсній осі

Задамо лінійне перетворення t_1 , яке переводить на комплексній площині рівносторонній трикутник у два рази меншого розміру (2.1):

$$t_1 : f_1(Z) = \frac{Z}{2} \quad (2.1)$$

Якщо тепер змістити цей маленький трикутник по горизонталі вправо на величину рівну $1/2$, то отримаємо перетворення t_2 , що переводить початковий трикутник, у трикутник, поданий на рис. 2.8. справа.

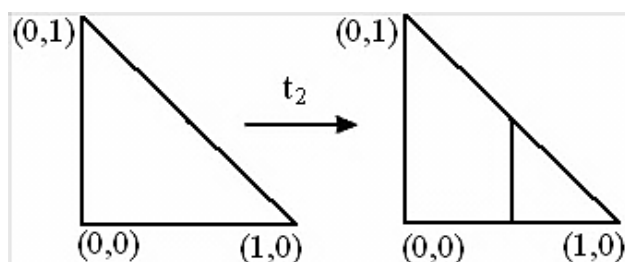


Рис. 2.8. Здви́г по дійсній осі

Відповідна цьому перетворенню функція дорівнює (2.2):

$$t_2 : f_2(Z) = \frac{Z+1}{2} \quad (2.2)$$

Останній, третій маленький трикутник виходить за допомогою

перетворення t_3 , поданого на рис. 2.9.

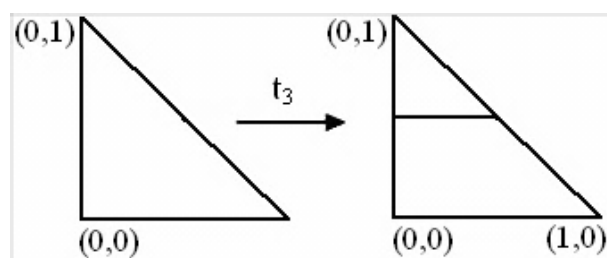


Рис. 2.9. Стискальне перетворення по мнимою осі

Відповідна йому функція $f_3(Z)$ виходить з $f_1(Z)$ із трансляцією на комплексний вектор (2.3):

$$t_3 : f_3(Z) = \frac{Z + i}{2} \quad (2.3)$$

У результаті три наведені лінійні функції виконують шукане стискальне перетворення початкового трикутника у три трикутники у два рази меншого розміру. Якщо тепер кожен з цих трьох маленьких трикутників піддати цим трьом стискальним перетворенням, то тоді виникне вже 9 трикутників, в 4 рази менше за початковий. Таким чином, виконуючи цю процедуру певну кількість разів, отримаємо трикутник Серпінського.

Декілька різних варіантів реалізації трикутника Серпінського наведені на рис. 2.10. Перший є класичним трикутником Серпінського, інший імітує "комаху" і третій – архітектурну споруду.

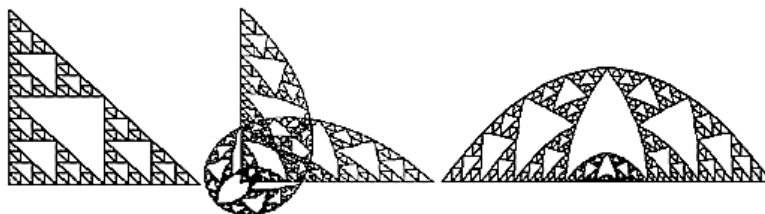


Рис. 2.10. Варіанти трикутника Серпінського

Для побудови фракталів алгебри використовуються ітерації нелінійних відображень, що задаються простими формулами алгебри. Прикладами фракталів алгебри є множини Мандельброта і Жюліа.

Алгоритм побудови фракталів алгебри досить простий і заснований на створенні фрактальних об'єктів на площині. Сутність алгоритму полягає у тому, що одне комплексне число $Z_n = x_n + iy_n$ перетворюється у інше комплексне кількість $Z_{n+1} = x_{n+1} + iy_{n+1}$ за ітераційним правилом, яке представлено формулою $Z_{n+1} = f(Z_n)$, де $f(Z_n)$ – певна нелінійна функція Z_n , n – номер ітерації.

Множина Жюліа поряд з безліччю Мандельброта посідає центральне місце в теорії фракталів. Безліч Жюліа є кордон області тяжіння нерухомих точок комплексних відображень.

Найбільш вивченим з них є відображення, подане формулою (2.4):

$$f(Z) = Z^2 + c, \quad (2.4)$$

де

$$c = c_1 + ic_2 - \text{const}, \quad Z, c \in \mathbb{C};$$

\mathbb{C} – множина всіх комплексних чисел.

Розглянутий алгоритм можна подати, як прямий і заснований на такій теоремі. Припустимо, що $|c| < 2$. Нехай $z \in \mathbb{C}$ і нехай $Z_n = f(n)(Z)$ для $n = 1, 2, 3, \dots$. Якщо існує таке n , що $|Z_n| \Rightarrow 2$, то має місце (2.5):

$$\lim_{n \rightarrow \infty} Z_n = \infty, \quad (2.5)$$

тобто орбіта (2.6)

$$\left[f^{(n)}(Z) \right]_{n-1}^{\infty} \quad (2.6)$$

прагне до нескінченності і Z не належить множині Жюліа. На цій теоремі заснований перший алгоритм – побудова на основі початкового відображення.

Другий алгоритм побудови, заснований на оберненому відображенні, спирається на властивість множині Жюліа, що полягає у тому, що для зворотного відображення, тобто (2.7)

$$f^{-1}(Z) = \sqrt{Z - c} \quad (2.7)$$

множина Жюліа – атрактор.

Результати реалізації наведених алгоритмів подані на рис. 2.11.

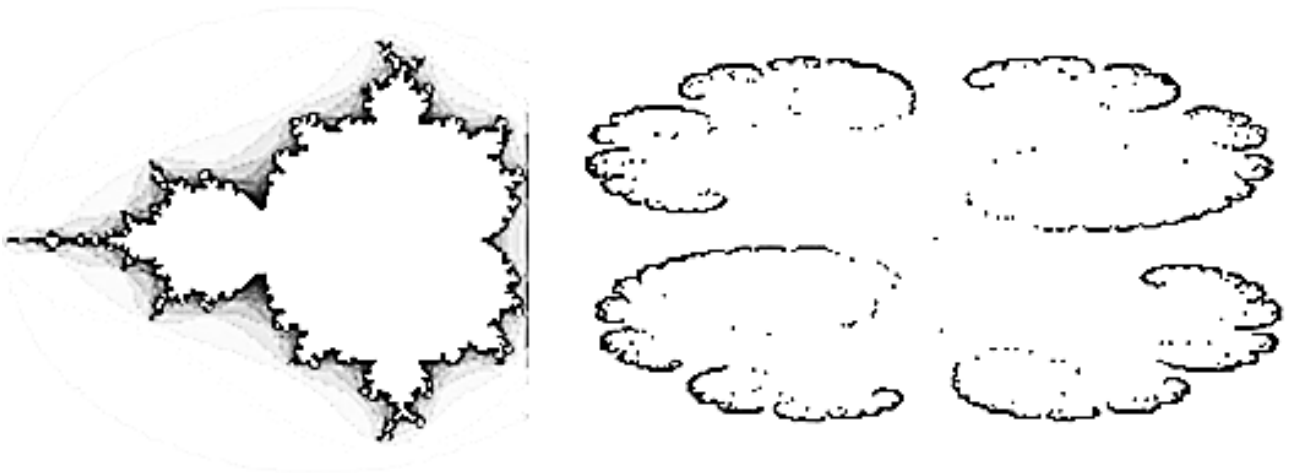


Рис. 2.11. Множини Мандельброта і Жюліа

Ще одним відомим класом фракталів є стохастичні фрактали, які отримують у тому випадку, коли у ітераційному процесі випадковим чином змінюють які-небудь його параметри. Разом із тим виходять об'єкти, дуже схожі на природні, – несиметричні дерева, порізані берегові лінії і т. д. Двовимірні стохастичні фрактали використовуються під час моделювання рельєфу місцевості і поверхні моря.

Алгоритм побудови стохастичних фракталів ще називають грою в "хаос", у якості початкової множини вибирають одну точку:

x_0 – початкова точка (з довільними координатами)

$$x_1 = T_1(x_0) \text{ або } T_2(x_0) \text{ або } T_3(x_0)$$

...

$$x_n = T_1(x_{n-1}) \text{ або } T_2(x_{n-1}) \text{ або } T_3(x_{n-1}).$$

На кожному кроці, замість того щоб застосовувати відразу три перетворення $T_1(S)$, $T_2(S)$, $T_3(S)$, застосовується тільки одне, вибране випадковим чином. Таким чином, на кожному кроці виходить рівно одна точка. Виявляється, що після певного перехідного процесу точки, згенеровані алгоритмом, заповнюють точно фрактальне зображення.

Чудовою властивістю алгоритмів, заснованих на теорії IFS, є те, що їхній результат (атрактор) абсолютно не залежить від вибору початкової множини E_0 або початкової точки x_0 . У разі детермінованого алгоритму це означає, що у якості E_0 можна взяти будь-яку компактну множину на площині: гранична множина, як і раніше, співпадатиме із

заданим фрактальним зображенням. У разі застосування алгоритму побудови стохастичних фракталів вибір початкової точки x_0 не впливає на формування заданого фрактального зображення.

Для рівномірного розподілу точок по екрану у алгоритмі побудови стохастичних фракталів афінне перетворення слід вибирати з вірогідністю (2.8):

$$P_i = \frac{\det(A_i)}{\sum_{j=1}^m \det(A_j)}, \quad (2.8)$$

де

A_i – матриця відповідного афінного перетворення ($i = 1, 2, \dots, m$).

Очевидно, що $p_1 + p_2 + \dots + p_m = 1$.

Результат побудови фрактального дерева за допомогою алгоритму побудови стохастичних фракталів наведений на рис. 2.12 (фрактальне дерево).



Рис. 2.12. Фрактальне дерево

Для побудови використовувалися стискальні афінні перетворення (2.9):

$$\begin{aligned}x_{n-1} &= a \cdot x_n + b \cdot y_n + e, \\y_{n-1} &= c \cdot x_n + d \cdot y_n + f.\end{aligned}\tag{2.9}$$

Рівняння можна подати в матричному вигляді (2.10).

$$\begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}.\tag{2.10}$$

У загальному випадку афінне перетворення на площині задається шістьма незалежними дійсними числами. Два числа e і f описують переміщення фігури на площині, а чотири числа a , b , c , d задають обертання фігури за незмінного початкового положення.

За допомогою алгоритму побудови стохастичних фракталів створюються складні фрактали (рис. 2.13).

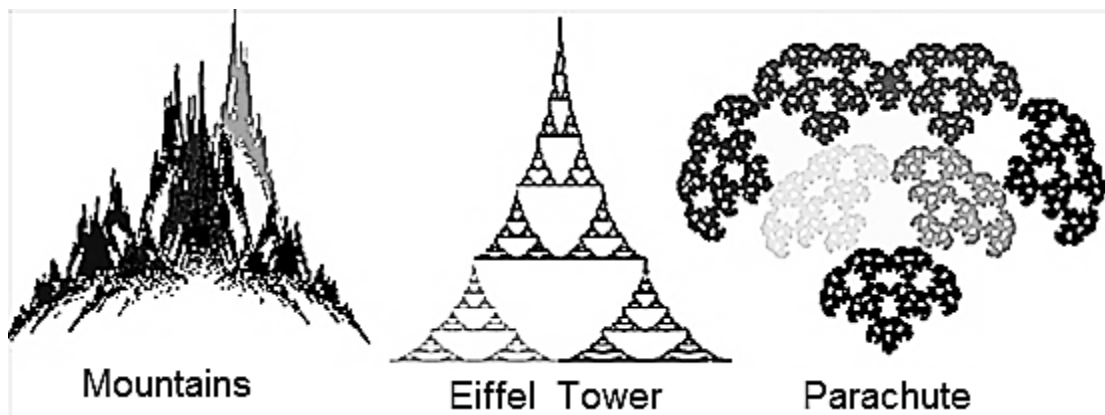


Рис. 2.13. Стохастичні фрактали

Існують фантастичні зображення рослин, отримані за допомогою програм. Це так звані L-системи, які становлять віртуальний програмований пристрій, що складається з пера, керованого ним механізму і аркуша паперу. Механізм, що управляє пером, дозволяє:

накреслити відрізок прямої заданої довжини в напрямку поточної орієнтації пера (команда f);

змінити орієнтацію пера по відносно до поточної на заданий відносний кут (команди $+$ і $-$);

запам'ятати поточний стан (команда $[$);

згадувати стан (команда $]$), що раніше запам'ятав.

Під станом у цьому випадку розуміють трійку чисел (x, y, a) , де x і y – це координати пера і a – це кут, що визначає напрям орієнтації пера. Таким чином, задавши певний початковий напрям a_0 , визначивши відносний кут повороту γ і задавши довжину відрізка, за допомогою послідовності команд $f+f+f+f$ можна намалювати квадрат. Визначивши відносний кут повороту γ у 60° , за допомогою послідовності команд $f++f++f$ можна намалювати рівносторонній трикутник.

На рис. 2.14 наведено два приклади фрактальних структур, побудованих за допомогою L-системи.

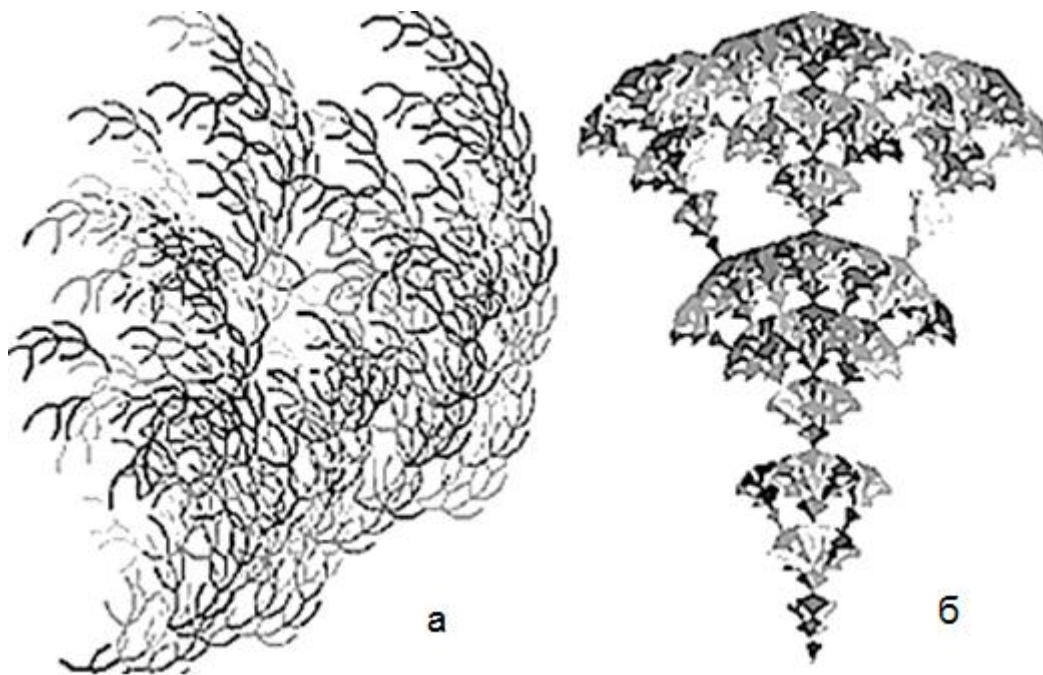


Рис. 2.14. Фрактали, побудовані за допомогою L-системи

Для створення фрактальних структур рис. 2.14 а і б використовують відповідні рядки команд:

$a = \{ " ", "- ", "f", "+", "f", "+", "[", "+", "f", "- ", "f", "- ", "]", "- ", "[", "- ", "f", "+", "f", "+", "f", "]" \};$

$b = \{ " ", "f", "[", "+", "f", "f", "]", "[", "- ", "f", "f", "]", "f", "[", "- ", "f", "]", "[", "+", "f", "]", "f" \}.$

Метод L-систем дозволяє зменшувати або збільшувати глибину рекурсії гілки. А змінюючи закон руху, можна отримувати різні зображення (рис. 2.15).



Рис. 2.15. Фрактали, побудовані за допомогою L-систем

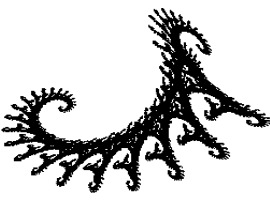

Потрібно відзначити, що фрактали – це область математичного мистецтва і інтуїції, коли за допомогою найпростіших формул і алгоритмів виходять картини надзвичайної краси і складності. У контурах побудованих зображень нерідко вгадуються листя, дерева і квіти. Одне з найбільш потужних додатків фракталів лежать в комп'ютерній графіці – це побудова ландшафтів, дерев, рослин і генерування фрактальних текстур.

2.4. Задавання на самостійну роботу

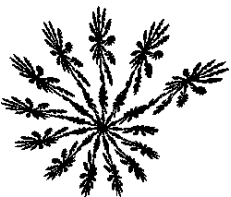





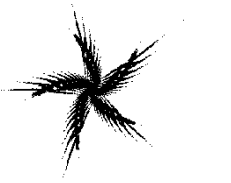

Використовуючи стискальне афінне перетворення (2.9) і (2.10) згенеруйте у середовищі Mathcad фрактали, наведені у табл. 2.1.

Таблица 2.1


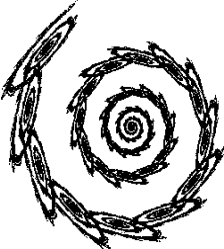

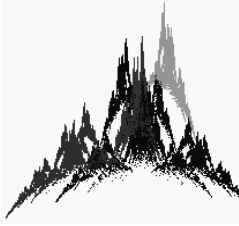

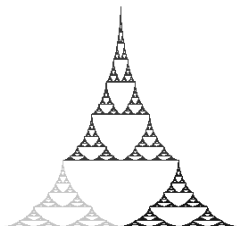
Фрактали

| Вид фрактала | Значення коефіцієнтів (a, b, c, d, e, f) |
|---|---|
| 1 | 2 |
|  | <pre>Dragon { 0.824074 0.281482 -0.212346 0.864198 -1.882290 -0.110607 0.787473 0.088272 0.520988 -0.463889 -0.377778 0.785360 8.095795 0.212527 }</pre> |
|  | <pre>Spiral{ 0.787879 -0.424242 0.242424 0.859848 1.758647 1.408065 0.895652 -0.121212 0.257576 0.151515 0.053030 -6.721654 1.377236 0.052174 0.181818 -0.136364 0.090909 0.181818 6.086107 1.568035 0.052174 }</pre> |

Продовження табл. 2.1

| 1 | 2 |
|---|---|
|  | <pre>Swirl { 0.745455 -0.459091 0.406061 0.887121 1.460279 0.691072 0.912675 -0.424242 -0.065152 -0.175758 -0.218182 3.809567 6.741476 0.087325 }</pre> |
|  | <pre>Dendrit{ 0.0 -0.7 0.7 0.0 0.0 0.0 0.5 0.7 0.0 0.0 0.7 0.3 0.0 0.5 }</pre> |
|  | <pre>Flower{ -0.87 0.32 -0.32 -0.87 0.00 0.00 0.89 -0.32 -0.11 0.11 -0.32 1.32 -0.11 0.11 }</pre> |
|  | <pre>Windy{ 0.358591 0.329160 -0.334007 0.580720 -5.889050 1.924278 0.361396 0.796218 0.136942 0.075187 0.719074 1.086998 0.408452 0.638604 }</pre> |
|  | <pre>Fern{ 0 0 0 0.165 0 0 0.01 0.84 -0.045 0.045 0.86 0 1.6 0.85 0.25 -0.26 0.23 0.25 0 1.6 0.07 -0.135 0.28 0.26 0.245 0 0.44 0.07 }</pre> |
|  | <pre>Maple{ 0.49 -0.01 0 0.62 0.25 0.02 0.316 0.27 -0.52 0.4 0.36 0 -0.56 0.316 0.18 0.73 -0.5 0.26 0.88 -0.08 0.316 0.04 0.01 -0.5 0 0.52 -0.32 0.052 }</pre> |
|  | <pre>Star{ 0.62 0 0 0.09 0 0 0.03 -0.79 0.55 -0.55 -0.79 1.79 0.55 0.97 }</pre> |
|  | <pre>Leaf{ 0.000000 0.243900 0.000000 0.305300 0.000000 0.000000 0.010000 0.724800 0.033700 -0.025300 0.742600 0.206000 0.253800 0.710544 0.158300 -0.129700 0.355000 0.367600 0.138300 0.175000 0.137386 0.338600 0.369400 0.222700 -0.075600 0.067900 0.082600 0.142169 }</pre> |

Закінчення табл. 2.1

| 1 | 2 |
|---|---|
|  | <pre>Tree{ 0.05 0 0 0.6 0 0 0.028000 0.05 0 0 -0.5 0 1 0.023256 0.459627 -0.321394 0.385673 0.383022 0 0.6 0.279070 0.469846 -0.153909 0.171010 0.422862 0 1.1 0.209302 0.433013 0.275 -0.25 0.476314 0 1 0.255814 0.421325 0.257115 -0.353533 0.306418 0 0.7 0.204651 }</pre> |
|  | <pre>Spiral{ 0.143268 0.372630 0 0.411999 -0.424035 0.228385 0.05 0.817528 -0.472 0.472 0.817528 -0.06 0.051538 0.95 }</pre> |
|  | <pre>Tree{ 0.458805 -0.225654 0.072682 0.601870 -0.001772 0.319478 0.31 0.342683 0.375610 -0.203368 0.546420 -0.022095 0.329764 0.20 0.135817 0.502731 -0.313469 0.138456 -0.020039 0.217063 0.16 0.253255 -0.489749 0.307978 0.349892 -0.007397 0.198101 0.23 0.066406 0.000000 0.000000 0.479424 -0.014923 -0.024215 0.10 }</pre> |
|  | <pre>Mountains{ 0.5 0 0.2 0.4 -0.6 0 0.25 -0.2 0 0.2 0.5 0.3 0.8 0.25 -0.8 0 0 0.8 0.1 0 0.25 0.2 0 0 0.6 0 0.35 0.25 }</pre> |
|  | <pre>Branch{ 0.409445 0.635563 0.699769 -0.364293 -1.362228 6.947697 0.6 0.440026 -0.218392 0.413335 0.376756 -3.087240 3.169590 0.25 0.107881 0.438639 0.084792 0.257114 -5.570585 1.155089 0.15 }</pre> |
|  | <pre>Eiffel_Tower{ 0.5 0 0 0.3 0 0 0.246154 0.5 0 0 0.3 0.5 0 0.246154 0.5 0 0 0.7 0.25 0.3 0.507692 }</pre> |

2.5. Інструментальні засоби для створення 3D-фракталів

2.5.1. XenoDream

XenoDream – призначена для створення тривимірних фракталів за допомогою спеціальних будівельних блоків (Holon).

Програма має стандартний інтерфейс (рис. 2.16), який включає:

заголовок;

робоче вікно;

панель інструментів вгорі і знизу робочого вікна;

панель управління, що складається з декількох вкладок.

На вкладці File панелі управління знаходяться дві круглі кнопки швидкого старту:

зелена кнопка – для випадкового вибору фрактала з наявних заготовок;

синя кнопка – для випадкового вибору параметрів фрактала.

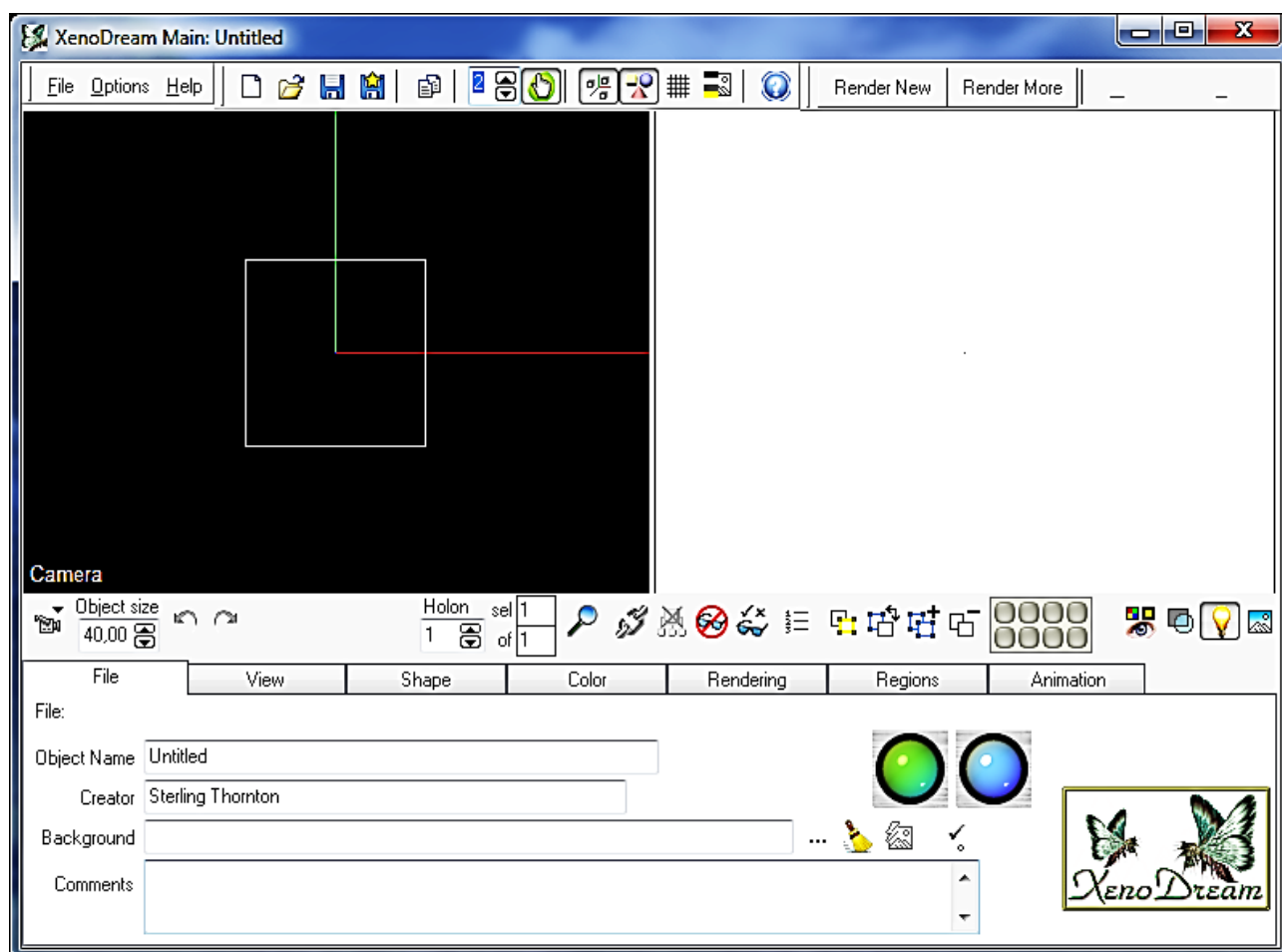


Рис. 2.16. Інтерфейс програми XenoDream

Потрібно натиснути на зелену кнопку кілька разів, щоб з'явився фрактал у вигляді листка папороті (рис. 2.17).

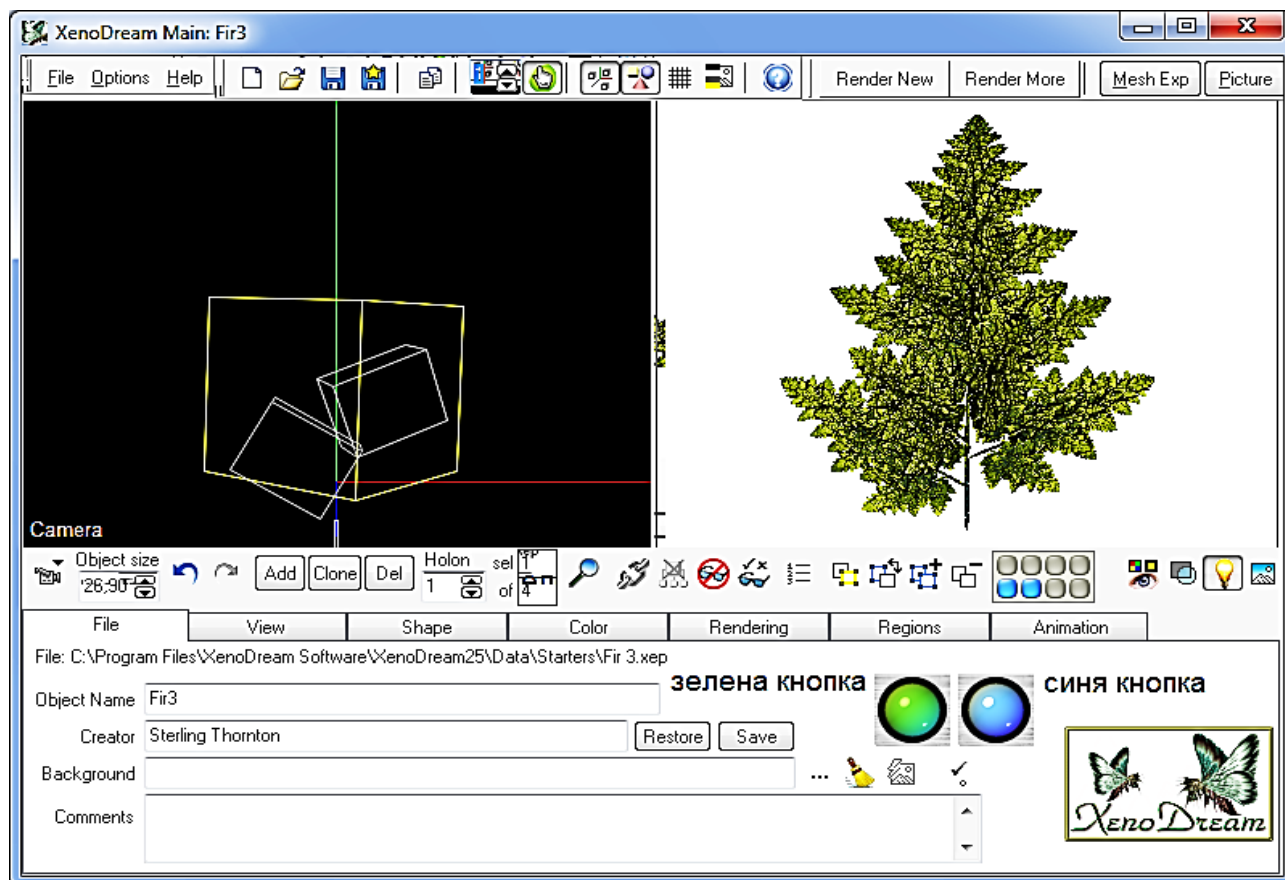


Рис. 2.17. Фрактал у робочому вікні програми

Жовтим підсвічується активний будівельний блок (Holon). Загалом фрактал складається з двох будівельних блоків.

Після завантаження фрактала у робоче вікно програми активізувався нижній ряд кнопок панелі інструментів. За допомогою цих кнопок виконується управління будівельними блоками.

Вкладка View панелі управління призначена для зміни позиціонування фрактала у робочому вікні (рис. 2.18).

Для управління позиціонуванням фрактала використовуються чотири елементи управління. Перші два переміщують фрактал уздовж осей X, Y і Z, а другі два елементи управління обертають фрактал навколо цих осей.

Сітка дозволяє фіксувати проміжні варіанти позиціонування фрактала. Вона може запам'ятовувати до 20 положень фрактала у просторі.

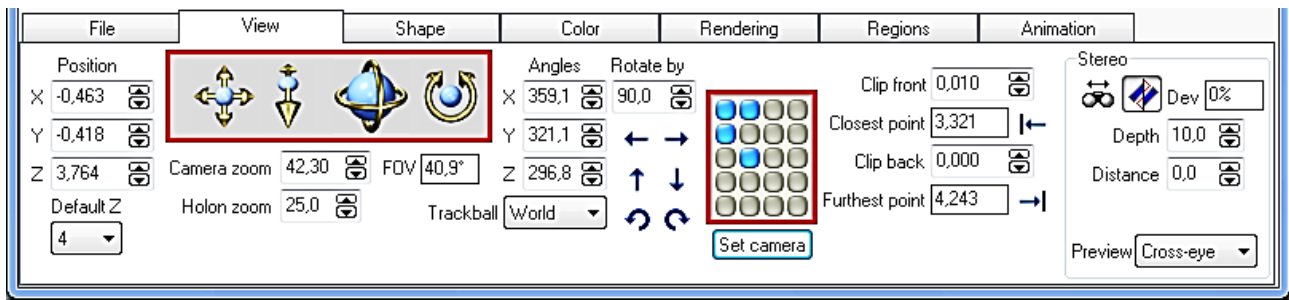


Рис. 2.18. Вкладка View панелі управління

Вкладка Shape панелі управління дозволяє (рис. 2.19):
 управляти режимами позиціонування фрактала;
 центрувати і вирівнювати будівельні блоки;
 встановлювати стилі будівельних блоків;
 перетворювати (Metamorph) будівельні блоки.

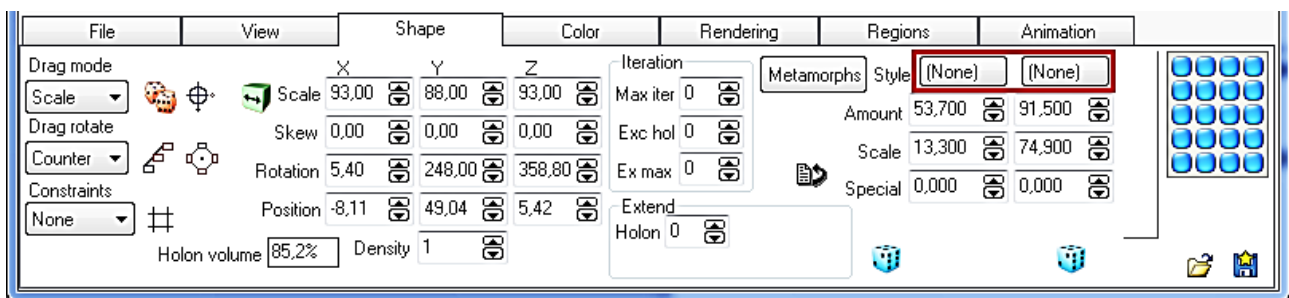


Рис. 2.19. Вкладка Shape панелі управління

Перетворення може бути застосоване до всіх будівельних блоків відразу або окремо. Зміна стилю будівельних блоків проводиться за допомогою кнопок. Кожен будівельний блок може мати до двох стилів, якщо натиснути на кнопку "none" з'являється вікно вибору стилів (рис. 2.20).

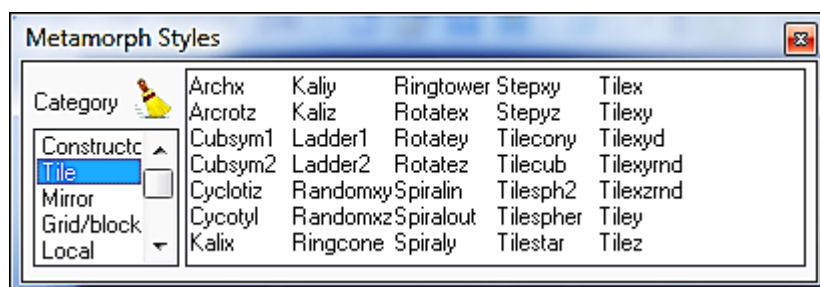


Рис. 2.20. Вікно стилів будівельних блоків

Вкладка Color панелі управління (рис. 2.21) призначена для: вибору методу роботи з кольором Holon Sequence і Metamorph; налаштування параметрів градієнта або вибору із списку заздалегідь налагоджених.

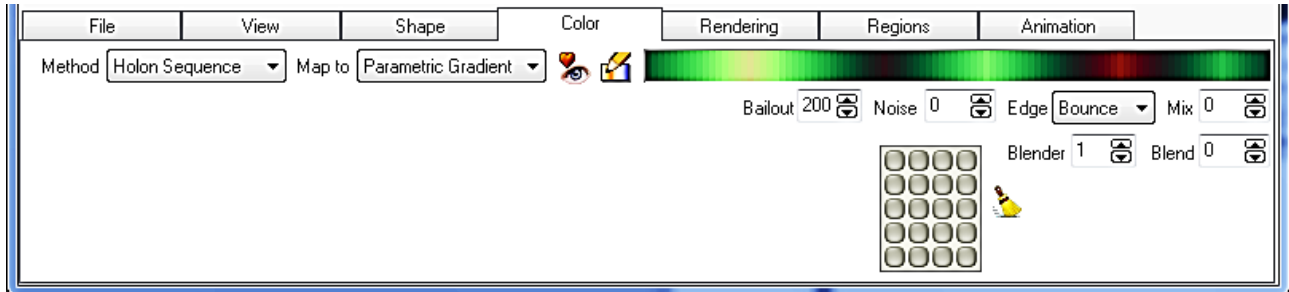


Рис. 2.21. Вкладка Color панелі управління

Вкладка Rendering (рис. 2.22) призначена для: управління освітленням; налаштування прозорості будівельних блоків; встановлення розміру зображень.

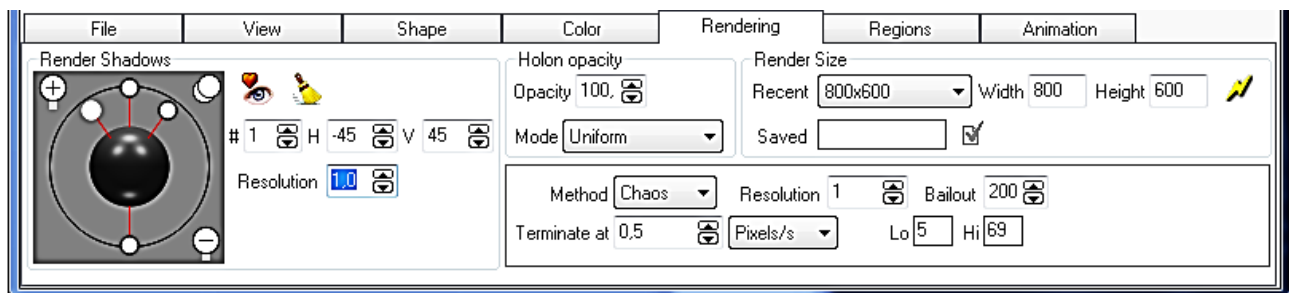


Рис. 2.22. Вкладка Rendering панелі управління.

Після встановлення і налаштування параметрів вкладки Rendering натискаємо кнопку “render new”, відкривається вікно рис. 2.23.

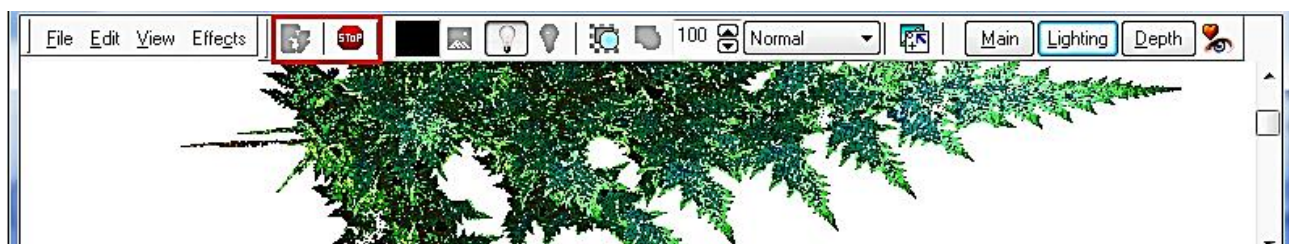


Рис. 2.23. Вікно промальовування фрактала

Елементи управління вікном:

File – зберігає і відкриває зображення фрактала і карту глибини;

Edit – копіює у буфер зображення фрактала;

View – встановлює розмір зображення у відсотках;

Effects – управляє ефектами, такими, як розташування фрактала, шум, інверсія, контрастність та ін.;

Stop – зупиняє поточний рендер зображення, потім його можна відновити, натиснувши кнопку “start new render”;

Найбільш крайня права кнопка дозволяє вибрати заздалегідь встановлені параметри рендера.

2.5.2. Incendia

Головне вікно програми Incendia можна умовно розподілити на дві частини вікно роботи з фракталом і вікно попереднього перегляду.

Вікно роботи з фракталом (рис. 2.24) містить сім вкладок Fractals, Render, Camera, Editors, About, Volume і Animat.

Перша група елементів управління на вкладці Fractals дозволяє вибрати сам фрактал, натиснувши на фрактальне ім'я. Після вибору фрактала, у вікні перегляду з'являється зображення з випадковим набором різних параметрів. В останніх версіях програми міститься понад 40 типів фракталів.

Далі йде коротка інформація про вибраний фрактал:

ім'я фрактала;

кількість перетворень;

кількість базових форм.

Наступна група елементів управління кнопки завантаження і збереження параметрів:

Save Parametres і Load Parametres – кнопки для збереження в текстовий файл, у якому визначені всі параметри програми і виконується завантаження вже раніше збережених параметрів (розширення файла “*.par”);

Save Buffers і Load Buffers – кнопки для збереження всіх параметрів програми у буфер на конкретний момент і вивантаження їх із буфера (розширення файла “*.Zb3”);

Save Script і Load script – кнопки збереження і завантаження скриптів. Кнопку Save Script видно тільки у деяких фрактальних типах, що дозволяє користувачеві експортувати вибрані для фрактала сценарії

(розширення файла "*.PRG").

Сценарій може бути змінений у будь-якому текстовому редакторі. Кнопку Load Script видно тільки в процесі вибору фрактала Incendia Script. За допомогою цієї кнопки можна завантажити будь-який Incendia Script.

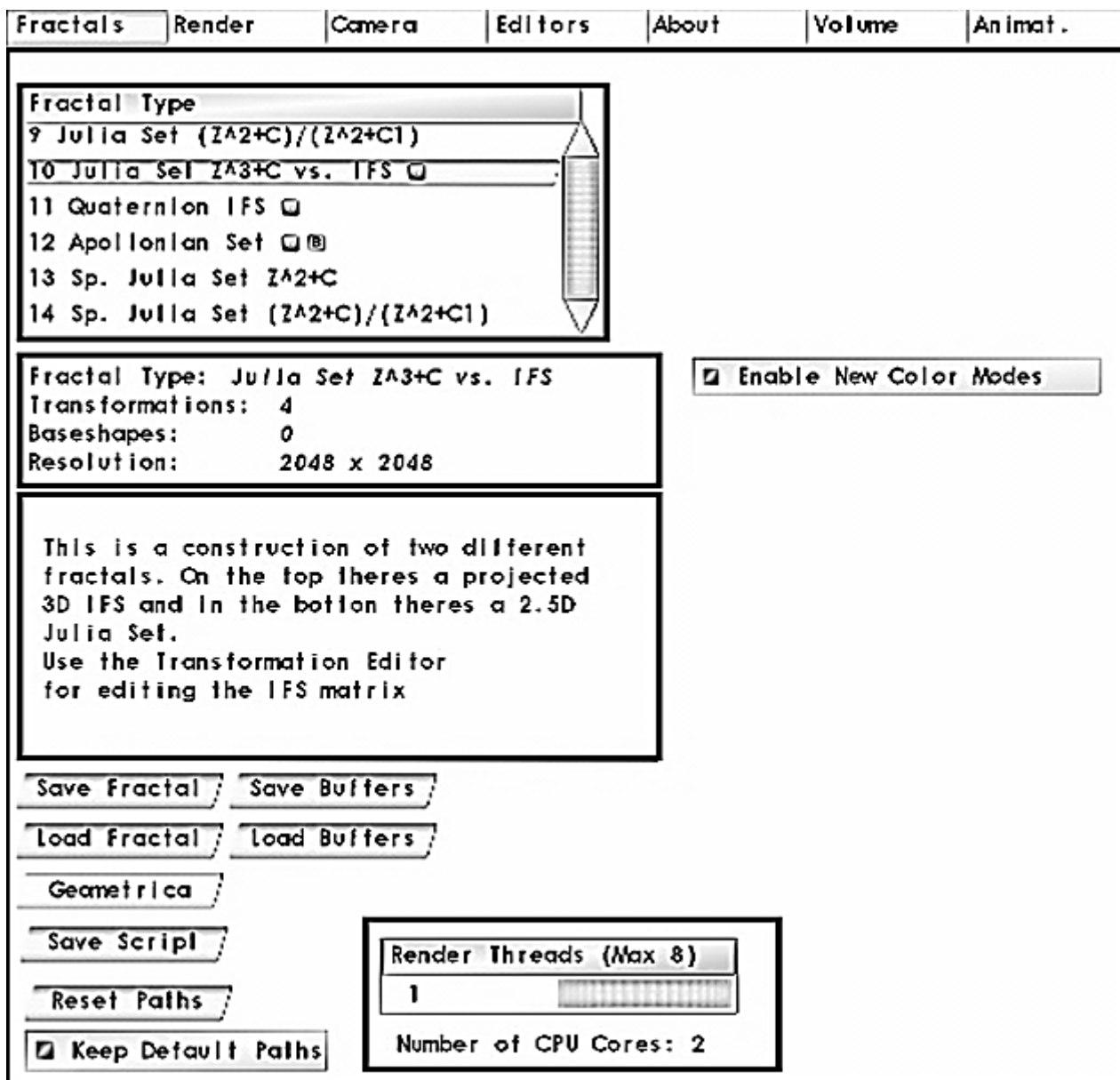


Рис. 2.24. Вікно роботи з фракталом

Робоче вікно вкладки Render (рис. 2.25) має чотири вкладки: Materials налаштування текстур об'єктів; Environment налаштування доквілля; Shaders налаштування шейдерів;

Advanced просунуті елементи управління і налаштування.

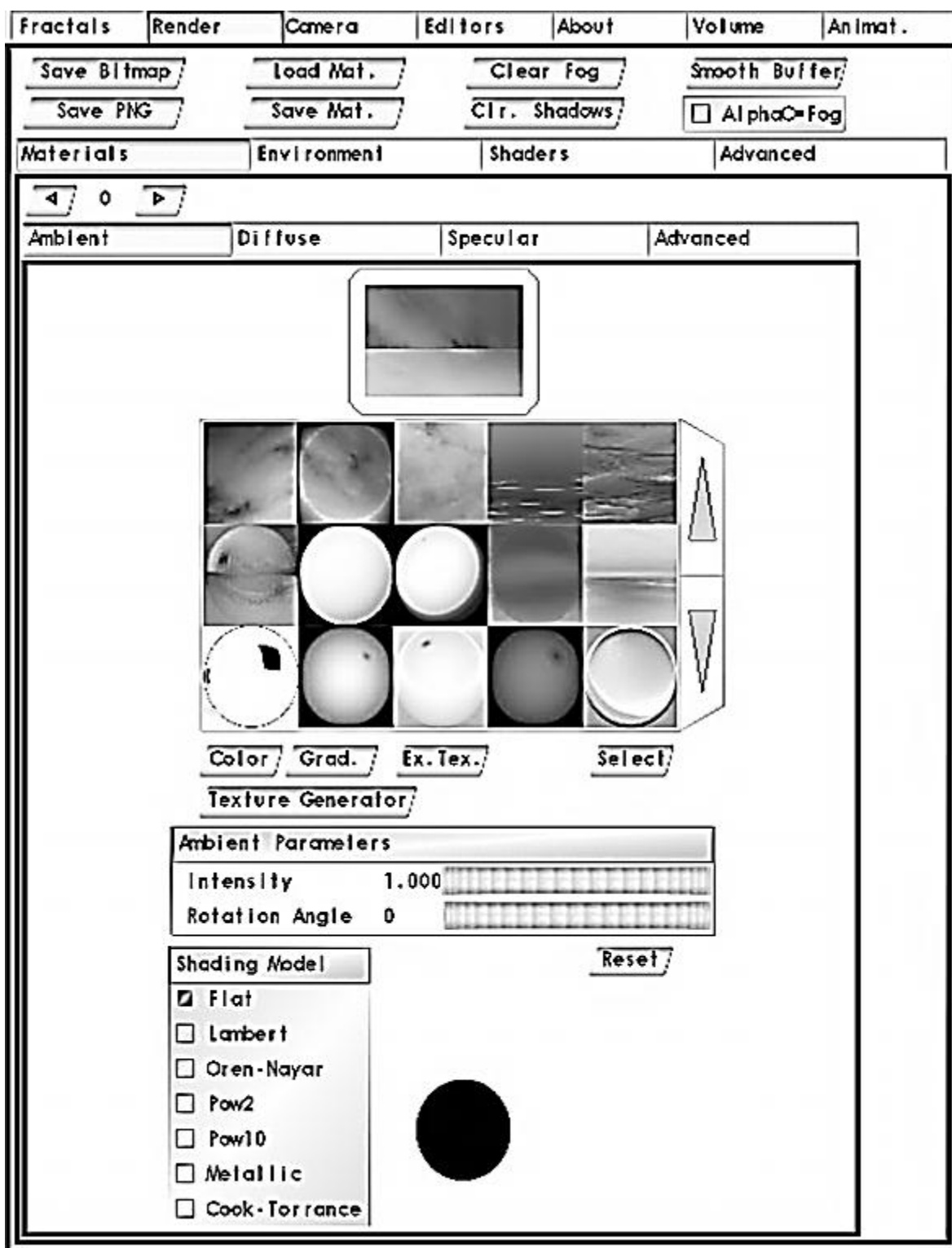


Рис. 2.25. Вікно вкладки Render

У верхній частині вкладки знаходиться група кнопок:
Save Bitmap – кнопка збереження фрактала у BMP-файл;

Save Mat. і Load Mat. – кнопки збереження і завантаження конфігурації фактичних матеріалів, у тому числі текстури;

Clear Fog – кнопка очищає буфер туману;

Clear Shadows – кнопка очищає тінювий буфер;

Save PNG – збереження фрактала у png-файл.

На вкладці Camera знаходяться елементи управління камерою. Програма Incendia дозволяє створювати анімовані 3D-фрактали.

Вкладка Editors призначена для створення фрактала на основі базових форм (рис. 2.26).

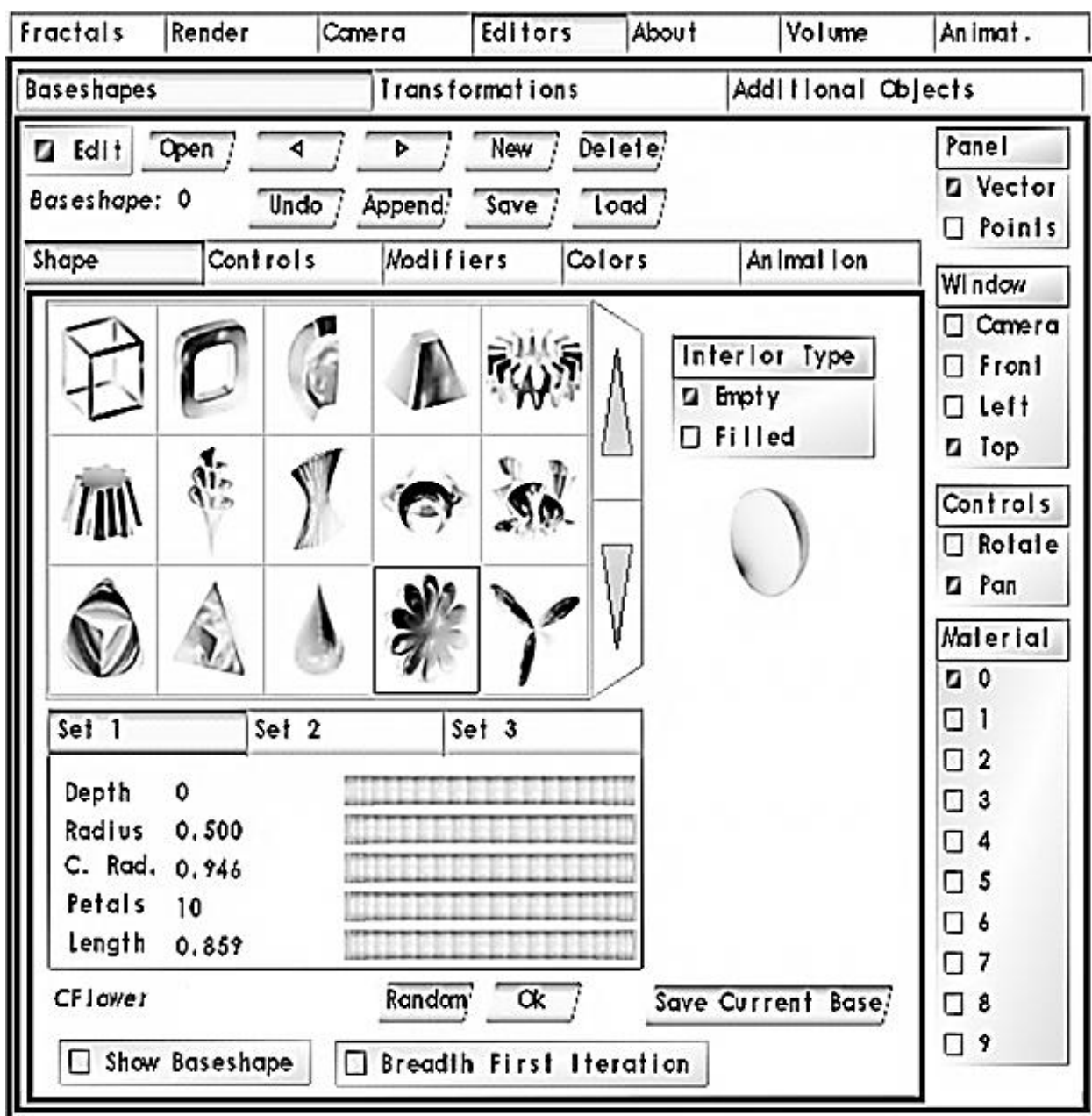


Рис. 2.26. Вкладка Editors

Вкладка складається з двох основних вкладок Baseshape і Transformations.

Для активації вкладки Baseshape (рис. 2.26) треба натиснути кнопку Edit. У верхній частині цієї вкладки знаходиться група кнопок:

Edit відкриває і закриває редактор базових форм;

Open відкриває всі елементи редактора;

New додає нову форму під фрактал, відкриваючи доступ до поточних параметрів;

Delete видаляє поточну форму із фрактала;

дві кнопки зі стрілками дозволяють переключатися між створеними формами (якщо їх більше від однієї);

Undo відмінює останню дію;

Append, Save, Load дозволяють додавати у програму, зберігати і завантажувати у фрактал інші базові форми у форматі bsf.

Далі йде блок роботи з базовими формами, який включає чотири вкладки – Shape, Controls, Modifiers colors, Anim.

Вкладка Shape включає вікно вибору базових форм.

Вкладка включає елементи управління параметрами тієї чи іншої форми:

Depth параметр, що визначає кількість фрактальних ітерацій;

Radius параметр, що визначає фактичний розмір базової форми;

Random задає випадкові параметри поточної форми.

Під вікном базових форм розміщені кнопки, які дозволяють побачити конкретно вибрану форму у вікні попереднього перегляду:

Show Baseshape дозволяє показати або приховати основну форму у центрі фрактала;

Breadth First Iteration встановлює випадкову кількість ітерацій базової форми.

Controls (рис. 2.27) дозволяє переміщувати і обертати фрактал у просторі.

Елементи управління цією вкладкою обмежують переміщення фрактала по окремим осям:

F - повне вільне переміщення по осям;

X - обмеження переміщення по осі X;

Y - обмеження переміщення по осі Y;

Z - обмеження переміщення по осі Z;

0 - переміщення заборонені, базова форма закріплена на початку

КООРДИНАТ.

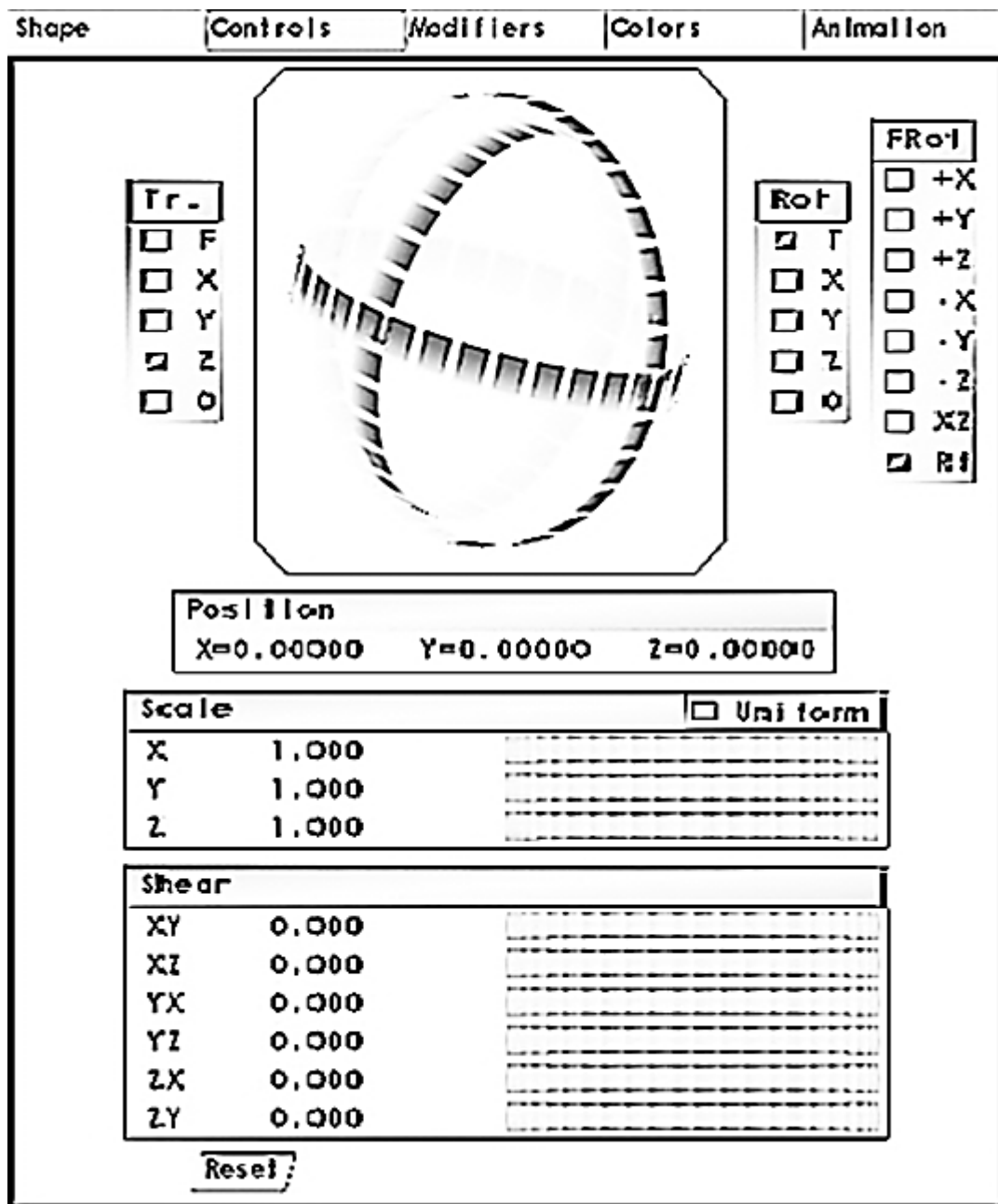


Рис. 2.27. Вікно вкладки Controls.

Пункт Rotation має таке ж призначення як і у Translation, тільки заважає обертанню базових форм:

- ++X повертає базову форму на 90 градусів по осі X;
- ++Y повертає базову форму на 90 градусів по осі Y;
- ++Z повертає базову форму на 90 градусів по осі Z;
- X повертає базову форму на -90 градусів по осі X;
- Y повертає базову форму на -90 градусів по осі Y;

-- Z повертає базову форму на -90 градусів по осі Z.

Rf відбиває фактичну орієнтацію базової форми, внаслідок чого напрями осей змінюються на протилежні.

Вкладка Colors відповідає за колір базових форм.

Функції вкладки Transformations ідентичні до тих, що знаходяться у вкладці Controls Baseshapes (рис. 2.28).

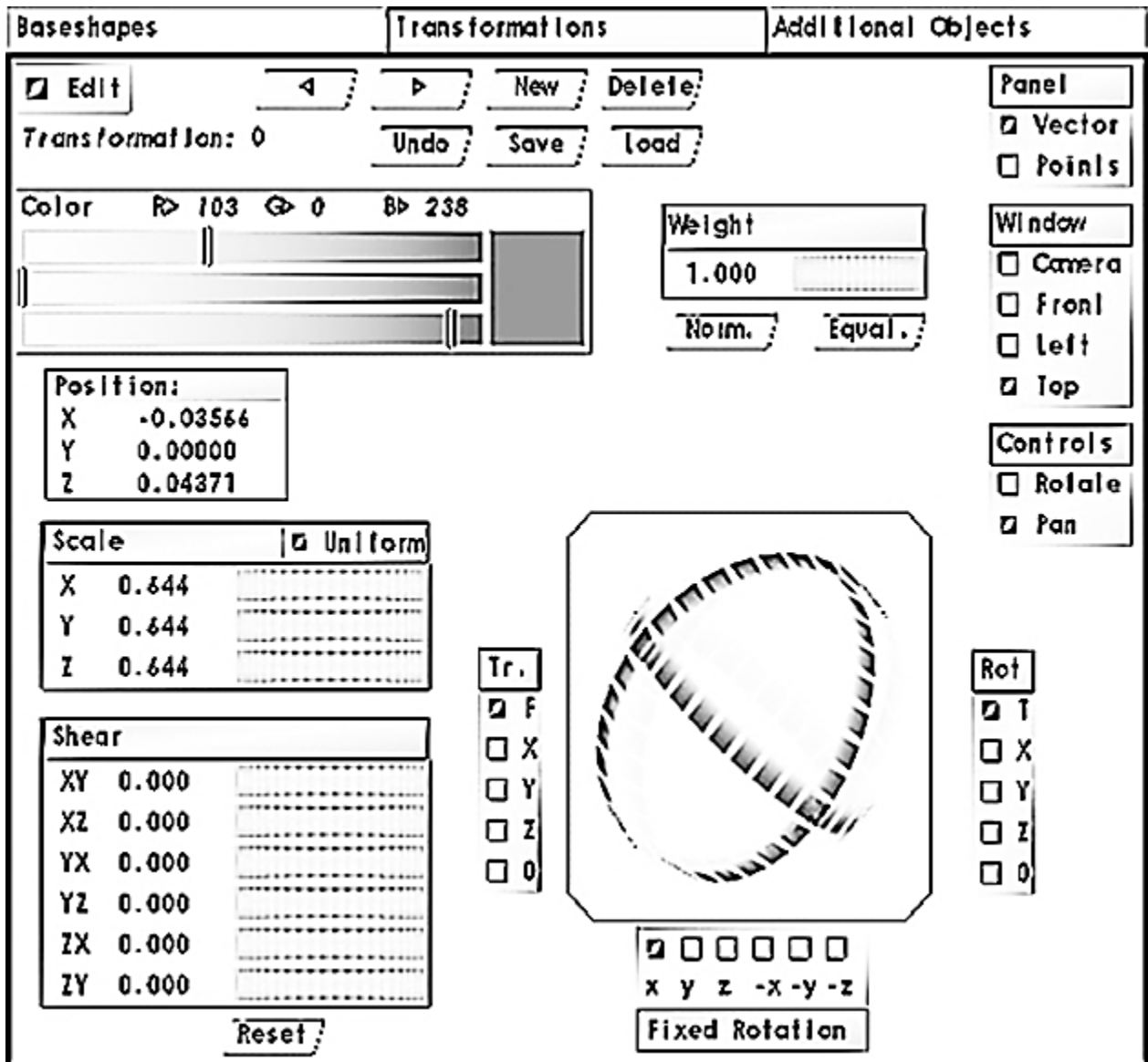


Рис. 2.28. Вікно вкладки Transformations

На вкладках Baseshapes і Transformations відображається Panel, яка має два режими – Vector (Векторний) і Point (Точковий). Ці два режими впливають на відображення фрактала у вікні попереднього огляду. Векторний режим зручний тим, що в ньому можна легко

побачити вибрану форму (вона підсвічується червоним кольором) і легко рухати її мишкою у вікні попереднього огляду. Але він не дозволяє побачити, як виглядатиме фрактал у кінцевому вигляді. Це дозволяє зробити точковий режим. Вибравши цей режим, можна подивитися, що приблизно становитиме майбутній фрактал. Панель Window дозволяє змінювати положення камери, завдяки чому можна подивитися на фрактал із різних сторін. Пункт Controls – змінює режим контролю переміщення фрактала Rotate – обертання, Pan – панорамування.

Вікно попереднього перегляду фрактала дозволяє подивитися, як виглядатиме фрактал (рис. 2.29).

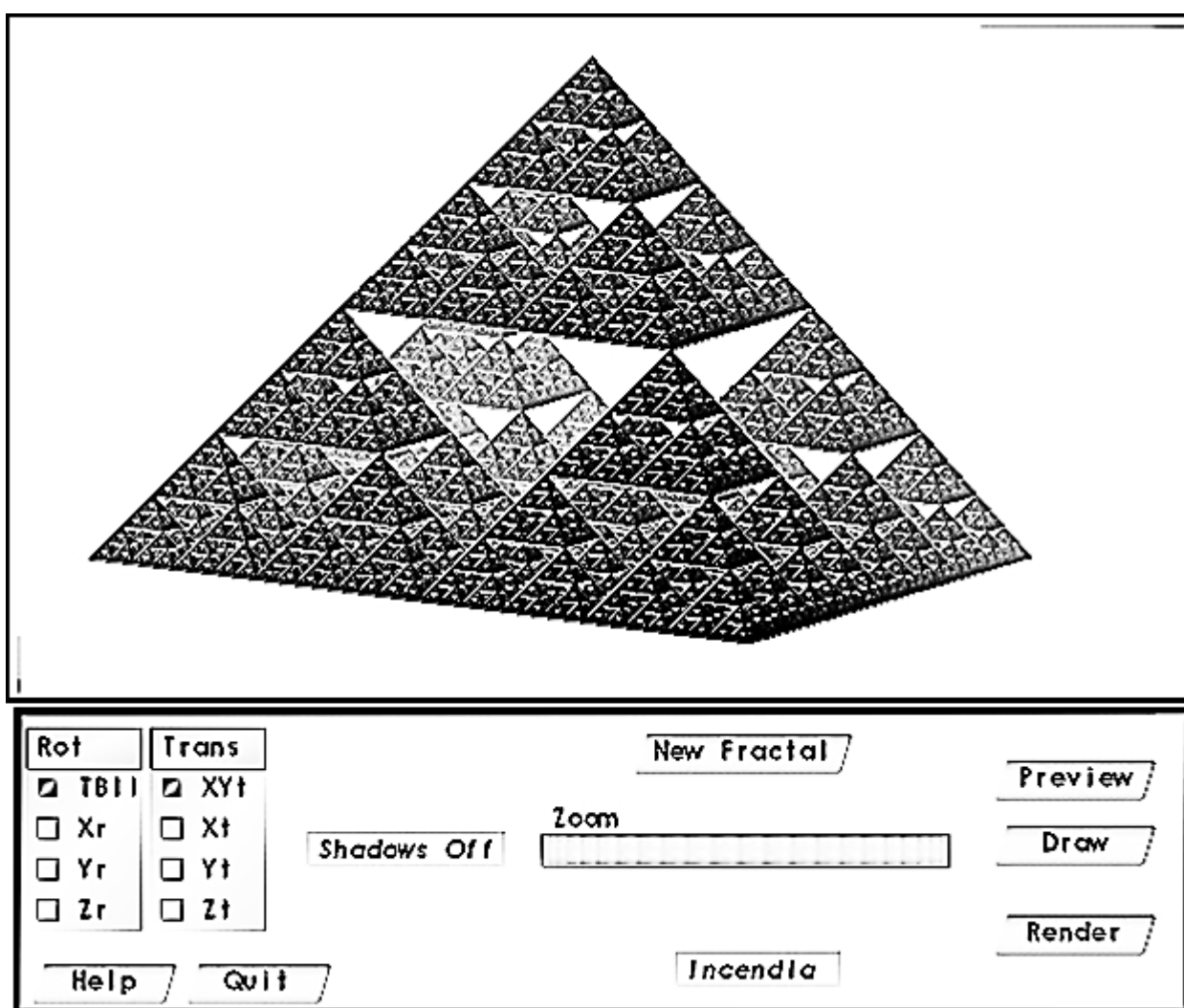


Рис. 2.29. Вікно попереднього перегляду

Кнопка New Fractal дозволяє створити новий фрактал на основі вибраного типу з випадковими параметрами.

Кнопки Preview і Draw використовуються для перегляду фрактала в двох режимах Draw (білі крапки) і швидкого попереднього перегляду.

У режимі Draw можна обертати і пересувати фрактал мишкою. Ліва кнопка миші управляє трекболом обертання, а права кнопка миші управляє переміщенням фрактала. Для зручності у пункті Rot можна обмежити обертання на різних осях, вибравши конкретний напрям:

TBlI – у будь-якому напрямку;

Xr – вгору, вниз;

Yr – управо, вліво;

Zr - за годинниковою стрілкою, проти годинникової.

Переміщення фрактала правою кнопкою миші також може бути обмежене конкретними напрямками у пункті:

Trans – XYt – у будь-якому напрямку;

Xt – по горизонталі;

Yt – по вертикалі;

Zt – по осі Z.

Повзунок Zoom контролює коефіцієнт масштабування. Масштабування також можна здійснювати коліщатком миші.

Примітка: кнопки перегляду зображення працюють тільки за умови, коли відключені параметри Baseshapes і Transformations на вкладці Editors, отже якщо не вдається подивитися на своє творіння, насамперед перевіряйте і вимикайте саме ці дані.

Кнопка Render запускає процес рендерингу. Якщо зупинити процес рендерингу і знову натиснути кнопку Render, з'явиться попередження, що попередній вміст буфера буде стертий.

2.6. Експорт фрактала з Incendia у 3ds max

Відкриваємо Incendia і вибираємо тип фрактала у вікні роботи з фракталом 1 3D Inverse Tree (рис. 2.24).

Після цього у вікні попереднього перегляду натискаємо кнопку New Fractal кілька разів, до ти, поки не з'явиться потрібний вид фрактала (рис. 2.29).

Далі у вікні роботи з фракталом (рис. 2.24) вибираємо вкладку Volume. У полі Volumetric Resolution вибираємо воксельное розширення і відповідно якість моделі, та швидкість прорахунку. Включаємо рендер (Start Volumetric Render) і після прорахунку натискаємо на кнопку Export Volume as OBJ (рис. 2.30).

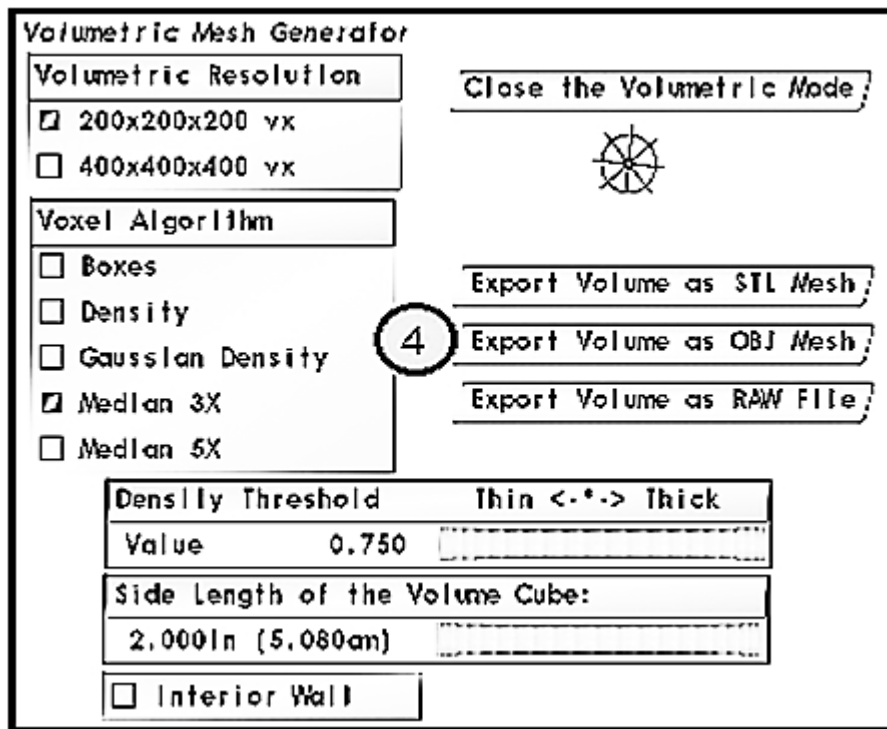


Рис. 2.30. Вкладка Volume вікна роботи з фракталом

3D-модель фрактала створена.

Після запуску 3ds max перетягуємо збережений у Incendia фрактал у його робоче вікно (рис. 2.31).

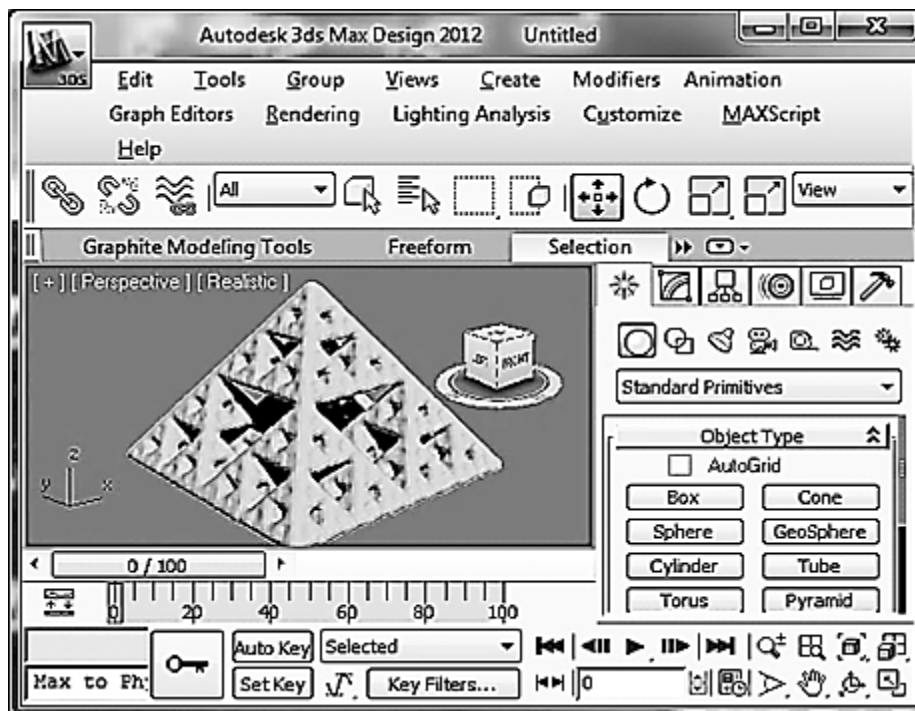


Рис. 2.31. Створений у Incendia фрактал у робочому вікні 3ds max

Тепер клацнемо десь поряд із фракталом і таким чином знімаємо з нього виділення.

У середовищі 3ds max можна модифікувати фрактал і призначити йому матеріал. Основні модифікатори, які можна застосувати до фрактала:

- Bend (вигин);
- Twist (скручування);
- Taper (загострення);
- Stretch (розтягання);
- Mirror (дзеркальна копія);
- Melt (плавлення);
- Push (надування);
- Ripple (брижі).

Застосуємо до піраміди Серпінського модифікатор Twist (рис. 2.32).

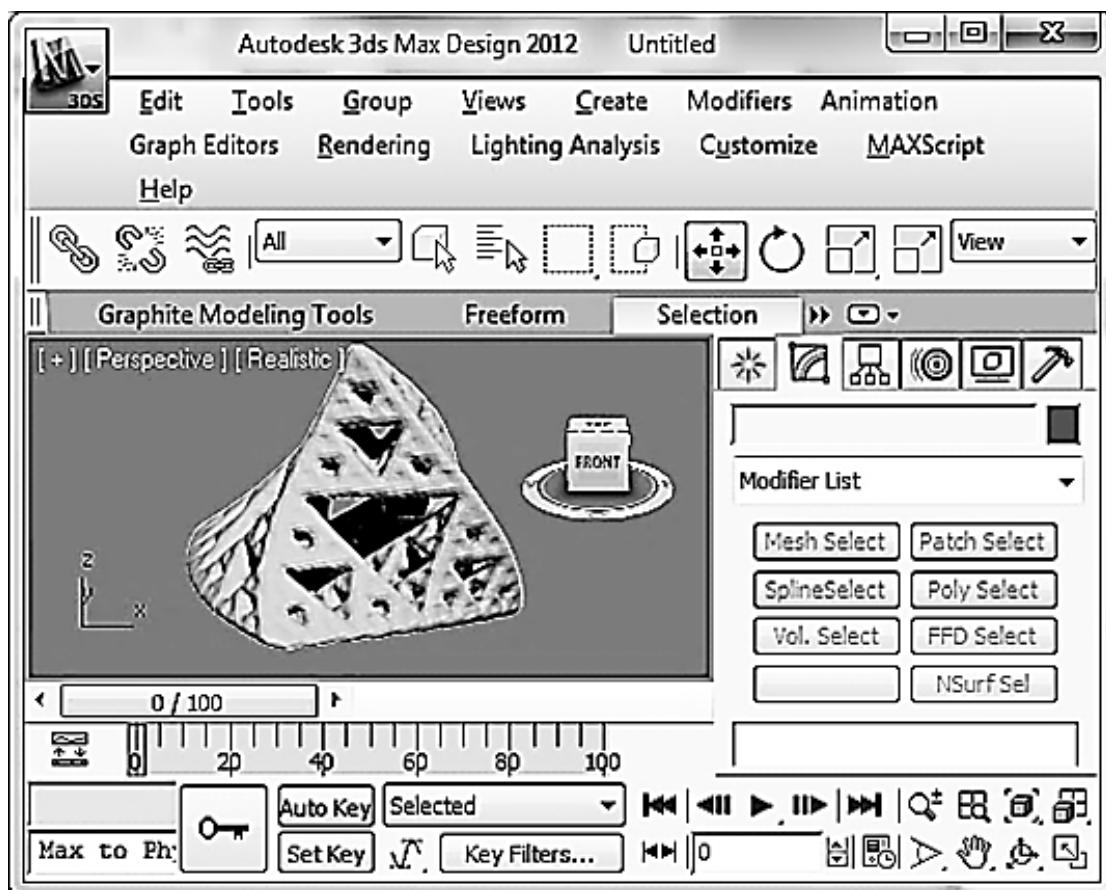


Рис. 2.32. Застосування модифікатора Twist до піраміди Серпінського

Після виконання всіх операцій у 3ds max виконується рендер і збереження 3D-фрактала.

2.7. Практичний приклад

Відкриваємо Incendia і вибираємо тип фрактала. Для побудови фантастичного міста найкраще підходить тип фрактала 10. Julia Set Z^3+C vs. IFS. Цей фрактал має різні види. Щоб отримати вид фрактала (рис. 2.33) треба кілька разів натиснути на кнопку New Fractal.

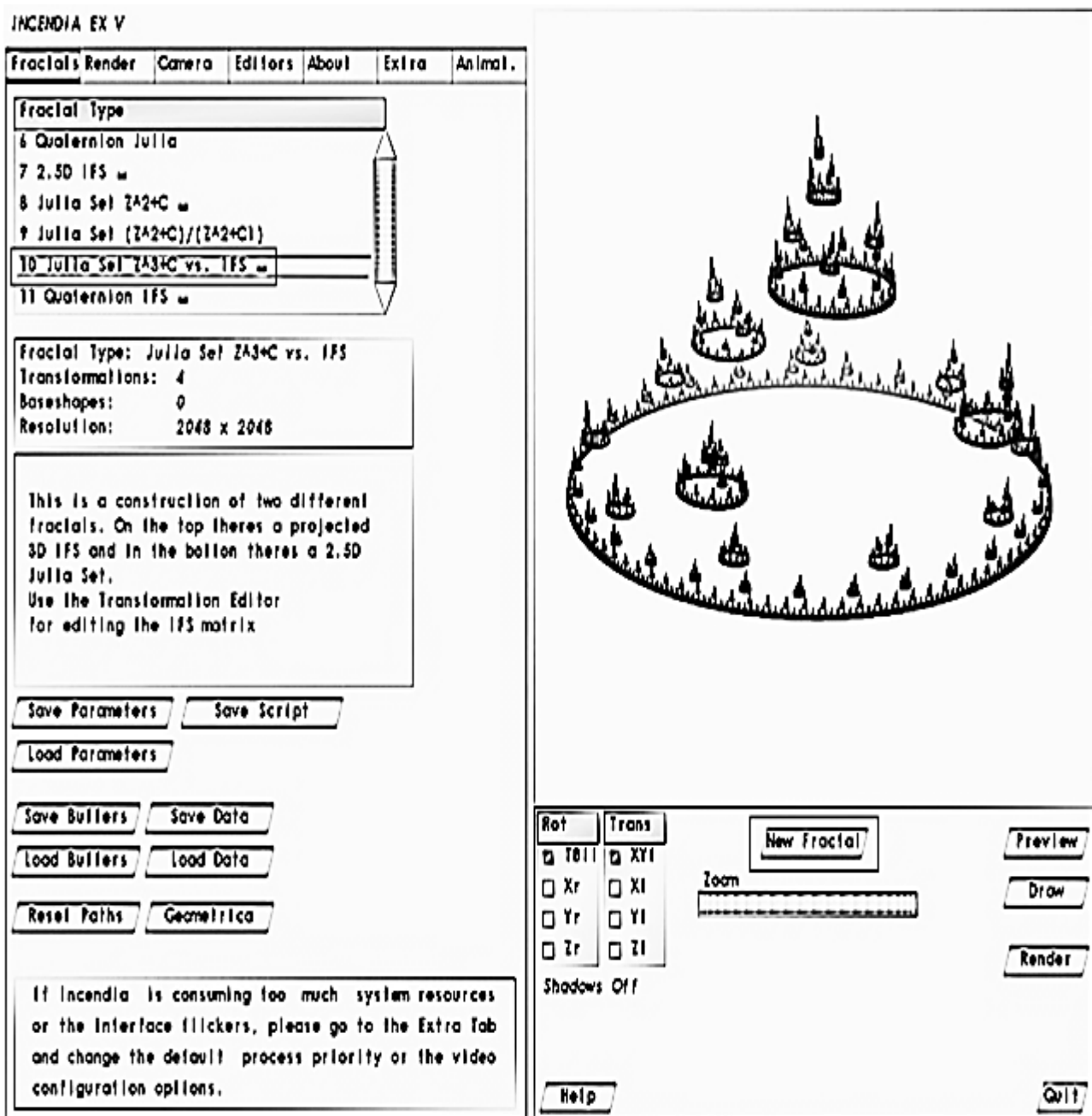


Рис. 2.33. Тип фрактала 10. Julia Set Z^3+C vs. IFS

Цей фрактал не має базових форм. Для побудови фантастичного міста їх потрібно додати. Для додавання потрібної форми переходимо

на вкладку Editors, далі вибираємо вкладку Baseshapes, натискаємо на кнопку Edit і далі натискаємо на кнопку New. На панелі форм Shape, що відкрилася, вибираємо потрібну і задаємо її параметри. У вікні перегляду форма виділена червоним кольором. Її за допомогою мишки можна переміщувати.

На рис. 2.34 показана перша вибрана форма, яка імітуватиме рослинність. Задамо для неї параметри і текстуру.

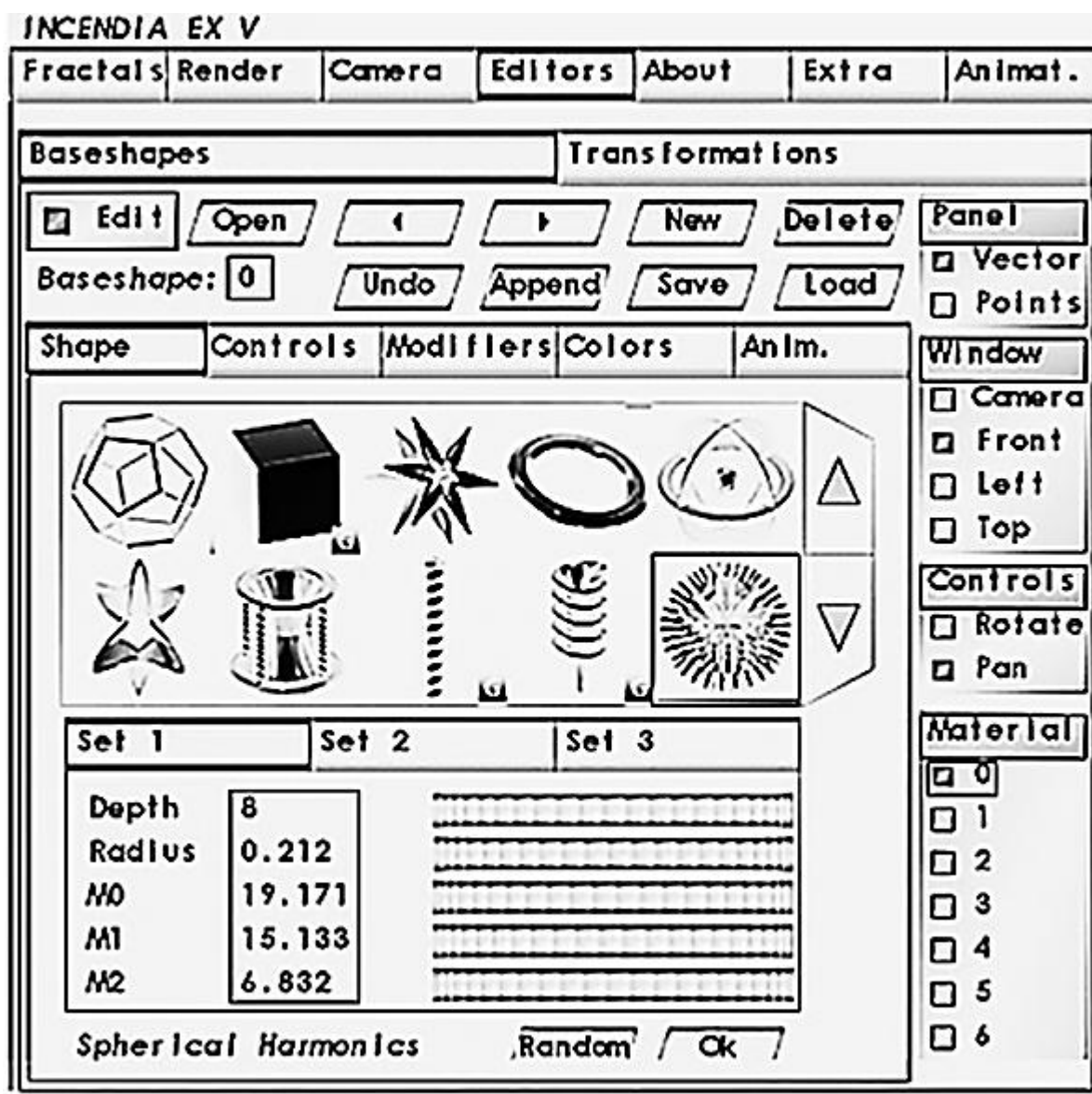


Рис. 2.34. Вибір форми із заданими параметрами для рослинності

Текстуру задають на вкладці Colors, активізують елемент управління Use Surface Texture і натискають на кнопку Load Texture. Далі вибирають потрібну текстуру (рис. 2. 35).

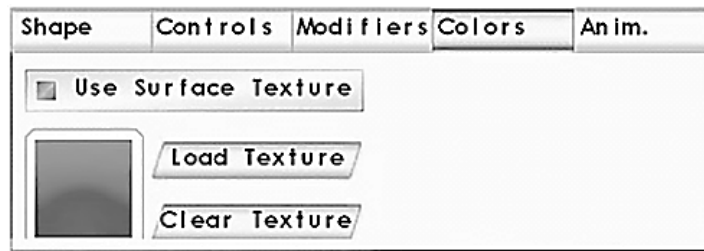


Рис. 2.35. Вікно вибору текстури для форми

Результат вибору текстури можна побачити у вікні перегляду натиснувши кнопку Points (рис. 2.36).

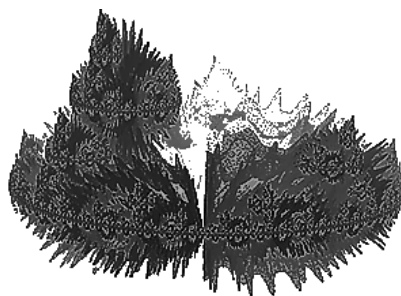


Рис. 2.36. Результат вибору текстури

Для створення стіни міста додамо форму із заданими параметрами рис. 2.37 і вкажемо для неї матеріал 1.

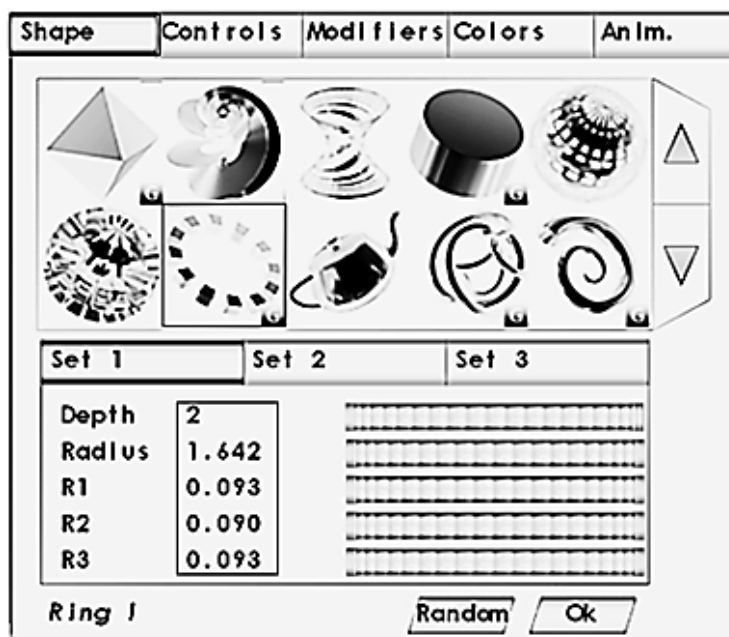


Рис. 2.37. Вибір форми із заданими параметрами для стін міста

Задаємо текстуру стіни у вигляді цегляної кладки.

Для побудови замку скористаємося двома формами із заданими параметрами (рис. 2.38).

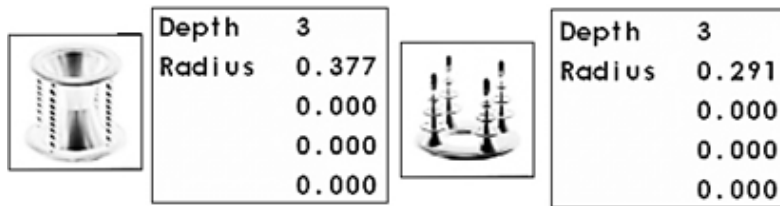


Рис. 2.38. **Форми для замку з заданими параметрами**

Перемикаємося на вкладку Controls і якщо необхідно, обертаємо форму навколо осі X, щоб колони було краще видно. Текстуру для цієї форми задати не можна, але матеріал вказуємо 1.

Далі додаємо вежу (рис. 2.39) Вказуємо для неї матеріал 2. Кількість ітерацій у неї Depth = 0, оскільки вежа в нас буде одна. Вежа у нас вузька і висока. Тому Radius зменшуємо, а висоту Height збільшуємо (можна задати будь-яке значення).

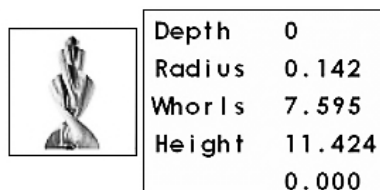


Рис. 2.39. **Форма для вежі з заданими параметрами**

Задаємо текстуру вежі.

Для додавання птахів використовується форма металевої зірочки із заданими параметрами (рис. 2.40). Щоб зірочка розлетілася, її потрібно поставити досить далеко від центра обертання фрактала.

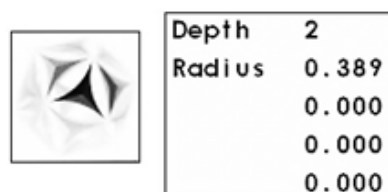


Рис. 2.40. **Форма для птахів з заданими параметрами**

А можна переключити панелі у режим Points і мишкою розтаскати частини зірочки так, як потрібно.

Спорудження фантастичного міста закінчене (рис. 2.41). У нього чотири трансформації і шість базових форм.

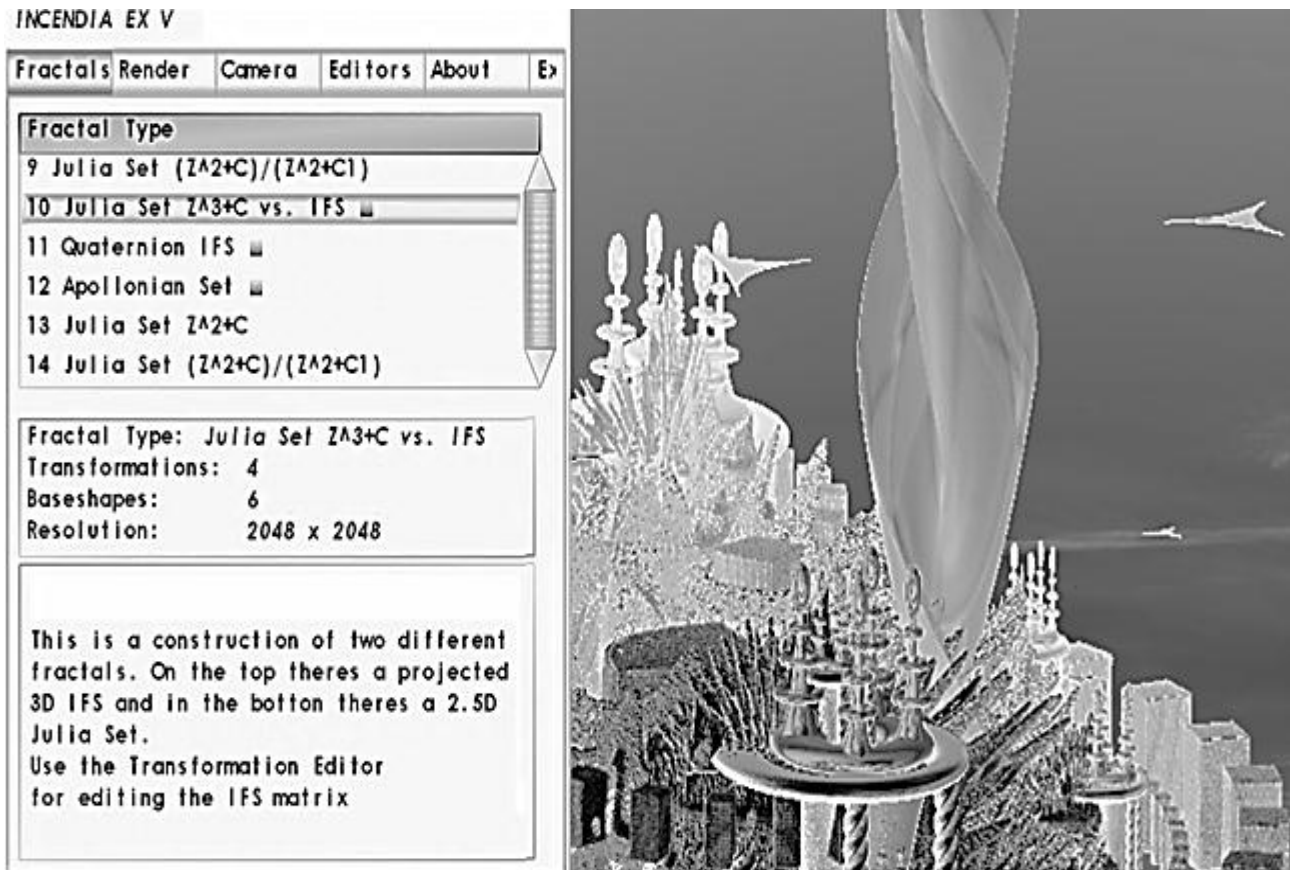


Рис. 2.41. Підсумкові характеристики фрактала фантастичного міста

В процесі створення фантастичного міста використовувалися форми, яким присвоювався номер матеріалу. Для вибору матеріалу переходимо на вкладку Render – Materials. Далі вибираємо шаблони матеріалу і дифузії, а також їхні параметри (рис. 2.42).

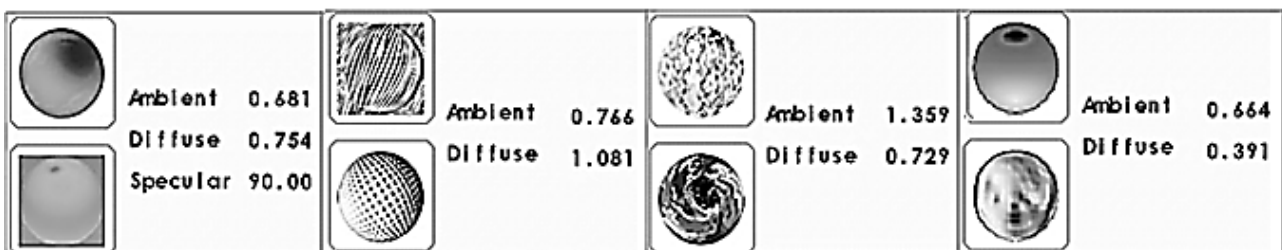


Рис. 2.42. Шаблони матеріалів, дифузій і їх параметрів

Для надання природного вигляду фрактальному місту необхідно налаштувати зовнішнє оточення. Для цього треба перейти на вкладку Environment і виконати налаштування:

ввімкнути туман Enable – Fog;

ввімкнути об'ємний туман Volumetric – Fog;

ввімкнути фон – Background;

в якості фону використовуємо природне освітлення (схід або захід сонця) – Skylight;

активізувати дозвіл використовувати фон як джерело світла для туману – Use Background as fog Color Source

встановити видимість у тумані – Fog R. A (чим більше значення, тим менший туман і тим далі ми бачимо);

встановити кількість пилу у повітрі – Sun N (чим вона більша, тим більш червоним буде схід або захід);

встановити колір туману – Vol fog Red, Vol fog Green, Vol fog.

Результати налаштування показані на рис. 2.43.

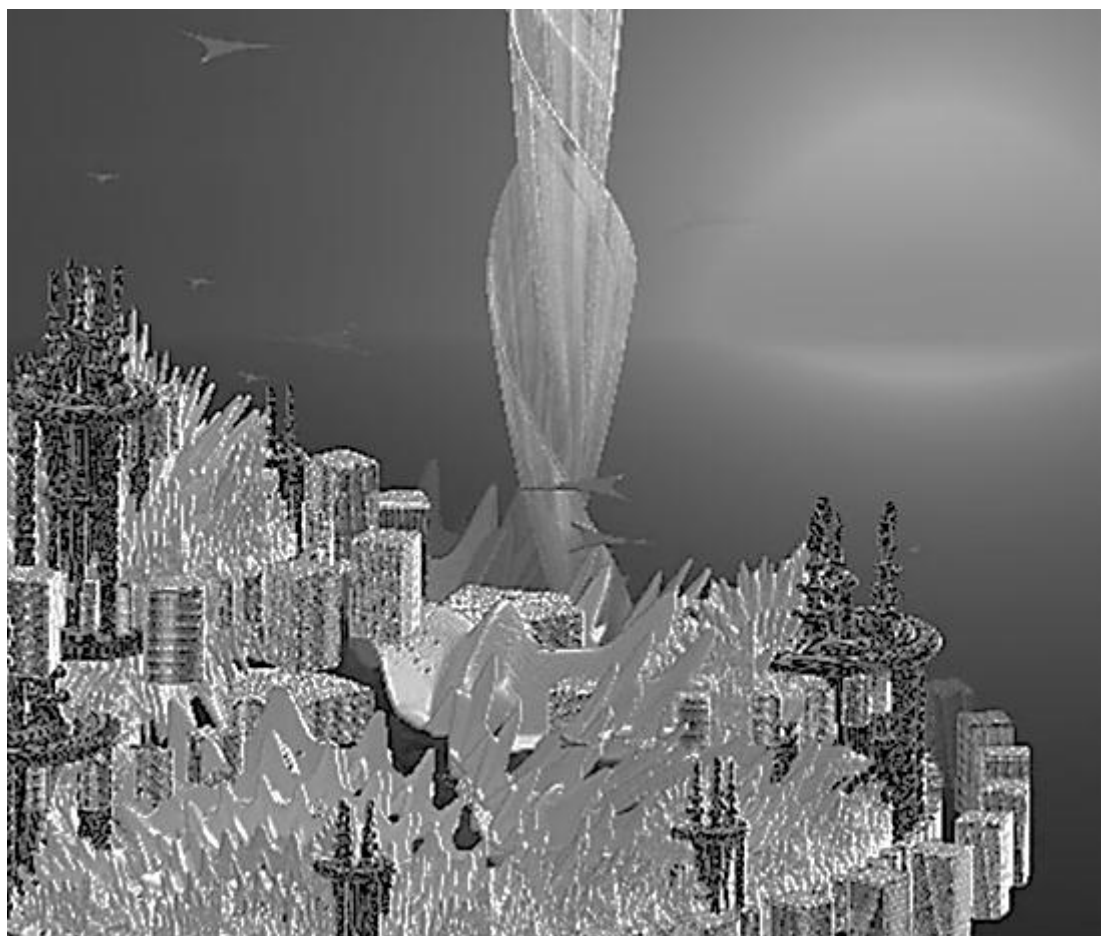


Рис. 2.43. Результати налаштування зовнішнього оточення

Для створення ефекту сходу сонця над горизонтом переходять на вкладку Camera і у вікні Light Direction мишкою тягнуть сонечко вниз, поки воно не зайде за горизонт і продовжують тягнути вниз, поки воно не почне підніматися. Далі вмикають тіні Shadow On.

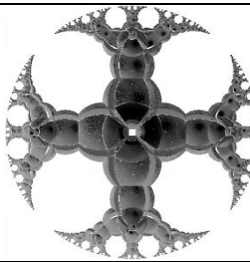



Після всіх налаштувань вмикають Render.

2.8. Задавання для самостійної роботи

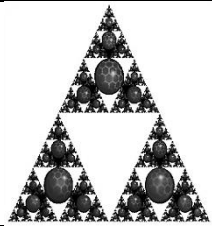
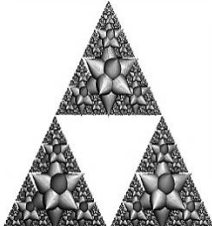

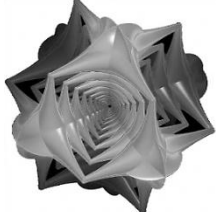
Використовуючи програми Incendia і 3ds max згенерувати 3D-фрактали, наведені у табл. 2.2.

Таблиця 2.2

Фрактали

| № | Вид фрактала | Ім'я фрактала | Операція модифікації у 3ds max |
|----|---|----------------------|--------------------------------|
| 1 | 2 | 3 | 4 |
| 1. |  | Cross.zdat | Ripple(брижі) |
| 2. |  | MandelbrotII.zdat | Melt (плавлення) |
| 3. |  | Mandelbrot.zdat | Push (надування) |
| 4. |  | Metallic Spiral.zdat | Ripple(брижі) |

Закінчення табл. 2.2

| 1 | | 2 | 3 |
|----|--|-----------------------------|---------------------------------|
| 5. |  | Simple Triangle II.zdat | Stretch (розтягання) |
| 6. |  | Simple Triangle III.zdat | Bend (вигин) |
| 7. |  | VolumeExample. zdat | Mirror (дзеркальна копія) |
| 8. |  | Hexahedral.zdat | Melt (плавлення) |

Контрольні запитання

1. Поняття фрактала.
2. Історія фракталів.
3. Класифікація фракталів.
4. 3D-фрактали.
5. Математичний опис фракталів.
6. Програми створення та редагування фракталів.
7. Редактор 3D-фракталів Mandelbulb3D.
8. Генератор тривимірних фракталів Incendia.
9. Інтерфейс Incendia.
10. Фрактали у 3ds max.

Розділ 2. Полігональні 3D-об'єкти

3. Полігональна 3D-графіка

3.1. Класифікація подання тривимірних геометричних об'єктів

Існують три способи подання тривимірних геометричних об'єктів (моделей):

- простий;
- граничний;
- об'ємний.

3.1.1. Простий спосіб задавання тривимірних об'єктів

До простих способів задавання тривимірних об'єктів належить точкове і дротяне (каркасне) подання. У точковому поданні об'єкт заданий сукупністю вершин, що належать поверхні об'єкта $V = \{V_1, \dots, V_n\}$.

Дротяна модель є розширенням попереднього способу. Об'єкти задаються сукупністю вершин і ребер (3.1), що їх сполучають:

$$V = \{v_1, \dots, v_n\}, E = \{v_i, v_j [f_k(u)]\}, \quad (3.1)$$

де $f_k(u)$ – це елемент, використовуваний, якщо ребро не є відрізком прямої, що сполучає вершини v_i та v_j . Як правило, для однієї моделі f_k є вектор-функцією одного і того ж типу.

Основна перевага цих способів простота подання. Тому вони застосовуються на проміжних стадіях роботи з геометричним об'єктом: попередня візуалізація або як початкові моделі для синтезу більш складних.

3.1.2. Граничний спосіб задавання тривимірних об'єктів

Граничне подання об'єкта передбачає точний математичний опис. Існують два основні способи опису поверхні: явний параметричний і у вигляді неявних функцій. Загальний вигляд параметричної поверхні наступний (3.2).

$$P(u, v) = \begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases}, \quad (3.2)$$

де для параметрів u і v , як правило, встановлюють область визначення або прямокутного ($u_a < u < u_b, v_a < v < v_b$), або трикутного ($u_a < u < u_b, v_a < u + v < v_b$) вигляду.

Можна виділити два основні типи граничних параметричних моделей:

полігональну модель, яка подана набором плоских граней;
патч-модель, або клаптева модель.

В останньому випадку гранями слугують частини поверхонь одного типу (білінійні, поверхні Кунса, бікубіческие поверхні, поверхні Без'є, поверхні на основі В-сплайнів).

Граничне задавання тривимірних об'єктів є найбільш поширеним, і для нього сформувався такі різновиди топологій:

Список вершин. У цій топології грань виражається через вершини:

$V = \{v_i\}$ – вершини $|V| = n$;

$F = \{(v_{j1}, v_{j2}, \dots, v_{jk} [f_j(u, v)])\}$ – грань (чи патч f_j), що складається з k вершин ($k \geq 3$).

Список ребер. Тут грань виражена через ребра:

$V = \{v_i\}$ – вершини $|V| = n$;

$E = \{e_k = (v_i, v_j [f_k(u)])\}$ – ребро. f_k – рівняння лінії;

$F = \{(e_{j1}, e_{j2}, \dots, e_{jk} [f_j(u, v)])\}$ – грань (чи патч f_j), що складається з k ребер ($k \geq 3$).

"Крилате" подання. Ця модель є розвитком моделі, заснованої на інформації про ребра. Відмінність полягає у тому, що у структуру, що описує ребра, додається інформація про взаємне розташування граней. Вона включає:

$V = \{v_i\}$ – вершини $|V| = n$;

$E = \{e_k = (v_{start}, v_{end}, n_{cw}, n_{ccw}, [f_k(u)])\}$ – ребро, де v_{start} – початок ребра, v_{end} – кінець ребра, n_{cw} – наступне (попереднє) ребро у тій грані, де e_k зустрічається у додатному напрямку обходу вершин, n_{ccw} – наступне (попереднє) ребро у іншій грані (де e_k зустрічається у від'ємному напрямку);

$F = \{(first_edge, sign [f_j(u, v)])\}$ – грань (чи патч f_j), де $first_edge$ –

перше ребро у ланцюжку подання грані, sign – знак (+/-), визначальний, у якому напрямку зустрічається ребро first_edge у цій грані.

"Крилате" подання є найбільш зручним для реалізації найважливіших алгоритмів із геометричними об'єктами:

перевірка правильності задавання;

алгоритми для полігональних моделей, пов'язані з обходом ребер (виділення плоских контурів, спрощення моделі шляхом видалення граней та інші).

Крім того, не більше, ніж за лінійний час, можна відновити будь-яку іншу топологію, йдучи ланцюжками зв'язків між елементами.

3.1.3. Об'ємний спосіб задавання тривимірних об'єктів

У об'ємному поданні базовими є об'ємні елементи або неявно подані примітиви. Найбільш відомі:

воксели;

метаболи;

суцільні конструктиви.

Основою воксельного подання слугує так званий воксель (чи осередок), що є кубічною областю простору. Тривимірний об'єкт визначається як масив вокселів. Можна виділити ті, що наслідують топологію воксельного об'єкта:

найпростіша – набір однакових вокселів, що апроксимує область простору, займану об'єктом;

$V = \{(\{L\} \times \{M\} \times \{N\}, \{1,0\})\}$ – елементу тривимірного масиву вокселів розміром $L \times M \times N$ ставиться у відповідність його заповнена (належність об'єкта);

октарное дерево – рекурсивний розподіл простору на 8 частин. Разом із тим встановлюється певний мінімальний розмір воксела. Лист дерева вважається заповненим, якщо він повністю належить об'єкта. Таким чином, топологія подана у вигляді дерева;

PM – октарние дерева – це гібрид октарного дерева і полігональної моделі для зменшення похибки апроксимації в разі досягнену мінімального розміру воксела у рекурсивному розподілі простору.

Воксельное подання є дуже зручним для реалізації просторових алгоритмів і теоретико-множинних операцій над об'єктами (об'єднання, віднімання, перетин), але має ряд недоліків, які обмежують сферу його

застосування:

низька точність для подання для більшості об'єктів;
великий об'єм займаної пам'яті.

Метаболи – це кулі різного радіуса (r), які можуть взаємодіяти залежно від близькості й радіуса взаємодії (R): Сфера = (координати, радіус, речовина). Взаємодія виражається через появу додаткової "матерії" між ними (рис. 3.1). Топологія як така тут відсутня.

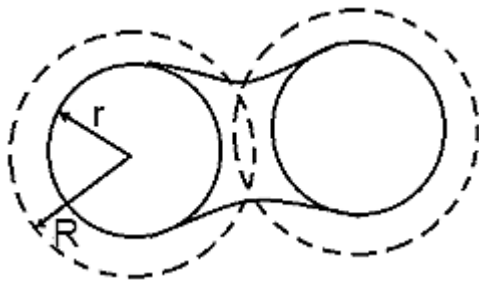


Рис. 3.1. Взаємодія двох метаболів

У процесі подання об'єкта у вигляді суцільних конструктивів використовують два набори:

базовий набір примітивів (паралелепіпед, сферу, конус, циліндр, тор, призму, піраміду і под.), що є структурними елементами об'єкта;

базовий набір теоретико-множинних операцій: унарного афінного перетворення (T) і бінарних операцій віднімання ($-$), перетину ($*$), об'єднання ($+$). Цей набір визначає топологію моделі, яка реалізується у вигляді формули теорії великих кількостей.

Наприклад, якщо наявні три примітиви A , B , C і формула $(A + T(B)) * C$, то це означає, що об'єднується об'єкт A з трансформацією об'єкта B і перетинається з об'єктом C . Перевага цього способу подання полягає у тому, що таким чином можна відносно легко моделювати досить складні об'єкти.

3.2. Моделювання за допомогою редагованих поверхонь

3.2.1. Полігональні сітки

Полігональна сітка (англ. *polygon mesh*) або неструктурована сітка це сукупність вершин, ребер і граней, які визначають форму багатогранного об'єкта в тривимірній комп'ютерній графіці й об'ємному

моделюванні. Гранями зазвичай є трикутники, чотирикутники або інші прості опуклі багатокутники (полігони). Вчення про полігональні сітки – великий підрозділ комп'ютерної графіки і геометричного моделювання. Різні подання полігональних сіток використовуються з різною метою і для різних застосувань. Безліч операцій, що проводяться над сітками, можуть включати Булеву алгебру, згладжування, спрощення і багато інших. Мережеві подання, такі як "потоківі" і "прогресивні" сітки, використовуються для передачі полігональних сіток мережею.

На рис. 3.2 зображений простий приклад полігональної сітки з чотирьох багатокутників з дев'ятьма вершинами і дванадцятьма ребрами.

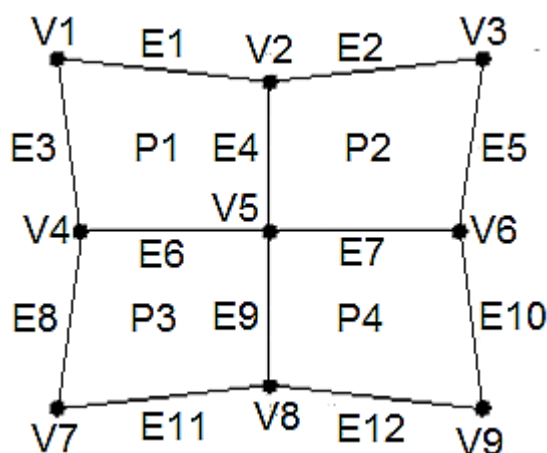


Рис. 3.2. Приклад полігональної сітки

На рис. 3.3 – 3.5 розглянуто декілька різних подань і наведені міркування щодо їхньої ефективності і зручності маніпулювання.

| P1 | P2 | P3 | P4 |
|----|----|----|----|
| V1 | V2 | V4 | V5 |
| V2 | V3 | V5 | V6 |
| V5 | V6 | V8 | V9 |
| V4 | V5 | V7 | V8 |

Рис. 3.3. Подання полігональної сітки з явним задаванням багатокутників

Компактно для одного багатокутника, але надмірно для набору, оскільки не існує загального опису загальних вершин і ребер.



Рис. 3.4. Подання полігональної сітки з покажчиками на списки вершин

Елементи списку покажчиків на вершини для кожного багатокутника посилаються на відповідні координатні дані для вершин. Це подання компактніше за попереднє, але складно знайти багатокутники із загальними ребрами.



Рис. 3.5. Подання полігональної сітки у вигляді списку ребер

Елементи списку ребер містять покажчики на вершини у списку вершин, що утворюють це ребро. Для забезпечення пошуку всіх вершин, що утворюють цей багатокутник, необхідно мати зворотні покажчики від вершини на одне з інцидентних до неї ребер.

Редаговану поверхню можна порівняти, умовно, звичайно, з глиною, з якої ви можете виліпити будь-яку форму (рис. 3.6).

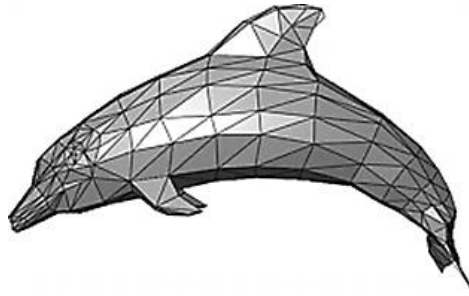


Рис. 3.6. Приклад полігональної сітки, що зображує дельфіна

Полігональні сітки можуть зберігатися у безлічі файлових форматів: FBX, 3DS, DXF, OBJ, PLY, STL, VRML, X3D.

3.3. Редаговані поверхні у 3ds max

У 3ds max є декілька типів редагованих поверхонь:

Editable Mesh (Редагована сітка) – поверхня, що складається з трикутних граней. У процесі роботі з редаговою сіткою можна використовувати режими редагування Vertex (Вершина), Edge (Ребро), Face (Грань), Polygon (Полігон) і Element (Елемент);

Editable Poly (Редагована полісітка) – поверхня, що складається з багатокутників. Для роботи з такими об'єктами можна використовувати режими редагування Vertex (Вершина), Edge (Ребро), Border (Межа), Polygon (Полігон) і Element (Елемент);

Editable Patch (Редагована патч-поверхня) – поверхня, що складається з клаптів трикутної або чотирикутної форми, які створюються сплайнами Без'є. Особливість цього типу редагової поверхні полягає у гнучкості управління формою створюваного об'єкта. Для роботи з Editable Patch (Редагована патч-поверхня) можна використовувати режими редагування Vertex (Вершина), Edge (Ребро), Patch (Патч), Element (Елемент) і Handle (Вектор);

NURBS Surface (NURBS-поверхня) – поверхня, побудована на NURBS-кривих. Цей метод створення поверхонь заснований на неоднорідних раціональних B-сплайнах (Non Uniform Rational B-splines, NURBS). Найчастіше цей спосіб використовують для моделювання органічних об'єктів, анімації обличчя персонажів. Цей метод є найскладнішим у освоєнні, але водночас найгнучкішим.

Редаговані поверхні відрізняються між собою здебільшого можливостями редагування моделі на різних рівнях подіб'єктів, а також

інструментами, які можна Разом із тим використовувати.

Практично будь-який об'єкт 3ds max можна перетворити у один із цих типів поверхонь. Для цього правою кнопкою миші викличе контекстне меню, клацніть на пункті Convert To (Перетворити) і у контекстному меню, що з'явилося, виберіть один з типів (рис. 3.7).

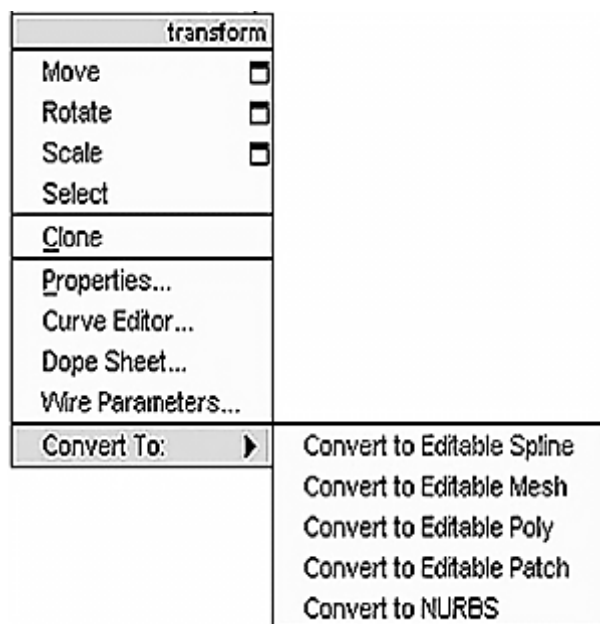


Рис. 3.7. Вибір типу поверхні у контекстному меню

Перемикаючись у різні режими редагування подоб'єктів, можна переміщати, масштабувати, видаляти і об'єднувати їх.

3.3.1. Основи полігонального моделювання

Як уже було сказано, значна частина об'єктів 3ds max може бути подана у вигляді сіток, що складаються з однотипних елементів або подоб'єктів, до яких належать вершини, ребра, грані, полігони й елементи (рис. 3.8).

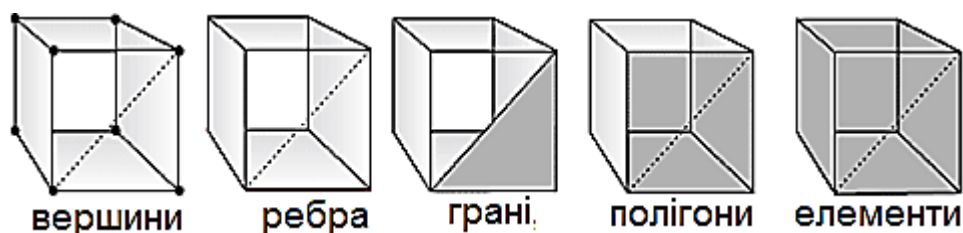


Рис. 3.8. Подоб'єкти

Призначені для роботи з подоб'єкти засоби скомпоновані за рівнями: Vertex, Edge, Face, Polygon, Element і Border..

Вершини – це точки, у яких сходиться і з'єднується один з одним будь-яка кількість ребер. Для роботи з вершинами призначений рівень Vertex (Вершина).

Ребра - це лінії границь між гранями. Ребра можуть бути видимими, якщо сусідні грані не лежать у одній площині (тоді вони відображаються суцільними лініями), або невидимими. За запитом користувача невидимі ребра можуть відображатися пунктирною лінією. За управління видимістю і положенням ребер відповідає рівень Edge (Ребро).

Грані – це ділянки площини трикутної форми, що є елементарними осередками сітки. В одній площині об'єкта може знаходитися безліч граней, які зовні будуть абсолютно невиразними. Для роботи з гранями призначений рівень Face (Грань).

Суміжні грані, що лежать у одній площині, можуть бути об'єднані у багатокутники-полігони. У разі перетворення об'єкта в тип Editable Poly процес формування полігонів з граней виконується автоматично. У звичайній сітці полігон – це просто подоб'єкт, що дозволяє виділити відразу всі суміжні грані, що лежать у одній площині. У полігональній сітці немає таких подоб'єктів, як грані, і вся вона складається тільки з полігонів, причому деякі полігони можуть бути і трикутними. За роботу з полігонами відповідає рівень Polygon (Багатокутник).

Крім того, сіткові об'єкти можуть редагуватися на рівні Element (Елемент), який використовується для роботи з групами граней, об'єднаними в елемент каркаса, що зручно, наприклад, під час вдавливання меж.

Редагування сітчастих об'єктів можна проводити як на рівні об'єкта загалом, так і на рівні подоб'єктів: граней, ребер або вершин. Щоб об'єкт став редагованим на рівні подоб'єктів і перетворився на редаговану сітку, необхідно виділити його і вибрати з контекстного меню команду Convert to > Convert to Editable Mesh (Конвертувати > Конвертувати у режим редагування сітки); можна також застосувати до об'єкта модифікатор Edit Mesh (Редагування сітки). Для перетворення об'єкта в тип Editable Poly з контекстного меню вибирається команда Convert to > Convert to Editable Poly (Конвертувати > Конвертувати у режим редагування полисетки). В обох випадках це призведе до появи на

панелі Modify цілої серії сувоїв:

Selection (Виділення) – відповідає за ввімкнення потрібного подоб'єктного рівня і управління режимами вибору подоб'єктів;

Soft Selection (М'яке виділення) – призначений для розширення можливостей виділення подоб'єктів і визначає закон поширення трансформацій за об'ємом редагованого каркаса;

Edit Geometry (Редагувати геометрію) – містить основні інструменти зміни геометрії подоб'єктів. Деякі інструменти однакові для усіх рівнів і для обох типів сіток, а інші є особливими для кожного рівня (і сітки). У перелік загальних інструментів входять, зокрема, такі:

Attach (Приєднати) – дозволяє додавати до редагованої моделі нові каркасні об'єкти, при цьому всі грані приєднується об'єкта виявляються об'єднаними в новий елемент;

Detach (Від'єднати) – відповідає за відділення вибраного подоб'єкта у окремий елемент або новий об'єкт

Remove Isolated Vertices (Видалити ізольовані вершини) – дозволяє видаляти окремо розташовані вершини об'єкта

View Align і Grid Align (Орієнтувати за поточним видом/Орієнтувати по сітці) – здійснює відповідну зміну орієнтації вибраних подоб'єктів;

Make Planar (Привести до площини) – встановлює площинну орієнтацію для вибраних подоб'єктів;

Collapse (Звести у точку) – виконує колапс (стискування) і об'єднання всіх вершин вибраних подоб'єктів у одну, розташовуючи її у геометричному центрі виділеної області;

Surface Properties (Властивості поверхні) – об'єднує такі інструменти налаштування властивостей поверхні, які призначені для кожного рівня.

Вибір потрібного рівня подоб'єктів здійснюється або підсвічуванням рівня в списках подоб'єктів Editable Mesh або Editable Poly, або клацанням по відповідній кнопці в світку Selection панелі Modify. Для вибору самих подоб'єктів використовують звичайні інструменти виділення: Select Object (Виділити об'єкт), Select and Move (Виділити і пересунути), Select and Scale (Виділити і масштабувати), Select and Rotate (Виділити і повернути) і Selection Region (Форма області виділення). Щоб послідовно виділити декілька об'єктів, під час виділення утримують клавішу Ctrl.

Для того щоб повернутися від редагування об'єкта на рівні

подоб'єктів до звичайного редагування, треба підсвітити у списку подоб'єктів рівень Editable Mesh або Editable Poly.

Щоб почати роботу з полігонами, потрібно конвертувати початковий об'єкт у Editable Poly. Це можна зробити, натиснувши на об'єкт правою кнопкою миші, потім, вибравши "Convert To:", натиснути "Convert to Editable Poly".

Для зручності роботи з об'єктом викличме контекстне меню у вікні Perspective і виберіть Edged Faces, тепер видно структуру об'єкта (у нашому випадку Box) з ребер і точок.

Після конвертації об'єкта з'явиться головна панель моделювання Editable Poly (рис. 3.9).

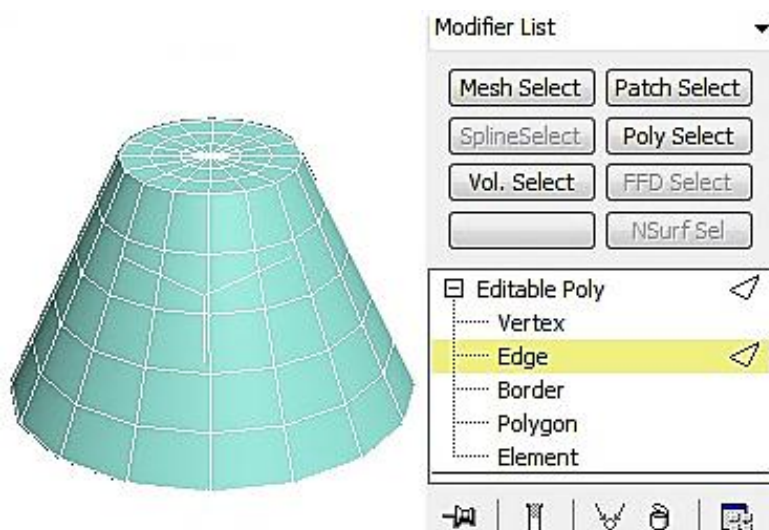


Рис. 3.9. Головна панель моделювання Editable Poly

Моделювання відбувається за допомогою зміни:

точки (Vertices) – найдрібніші частки, з яких складається об'єкт;

ребра (Edge) – лінії між двома точками;

полігону (Polygon) – частини об'єкта всередині якого немає точок, сторонами є ребра.

Точки, з яких складається об'єкт, можна виділяти, переміщувати, сполучати (рис. 3.10 а).

Для розмноження вибирається одна або більше точок і натискається кнопка біля команди Chamfer. З'являється вікно, де треба вибрати відстань між точками, що з'явилися. Розмноження йде по дотичних до точки ребрах (рис. 3.10 б).

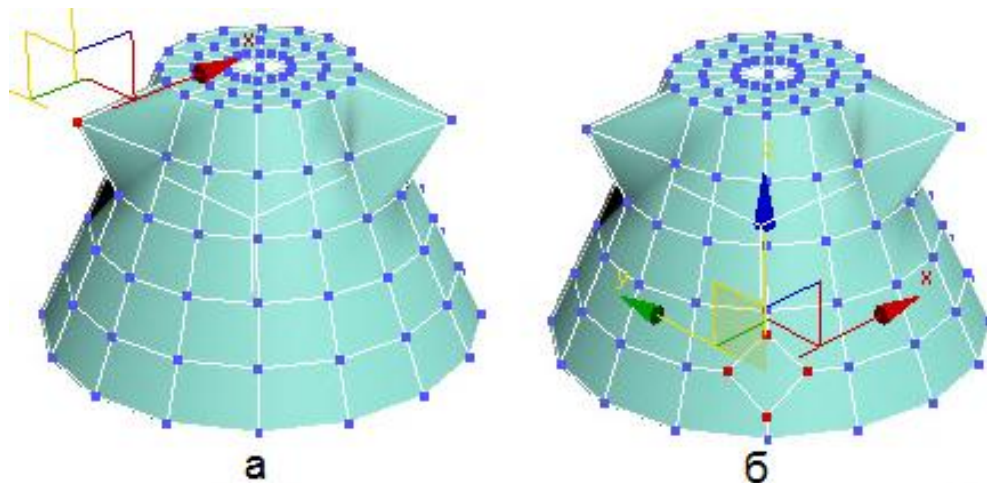


Рис. 3.10. **Переміщення і розмноження точок**

Видалення точок можна виконувати за допомогою команди Delete і Remove. У першому випадку віддаляються всі сусідні полігони, а в другому випадку точка зникає, але полігони залишаються на місці (рис. 3.11 а, б).

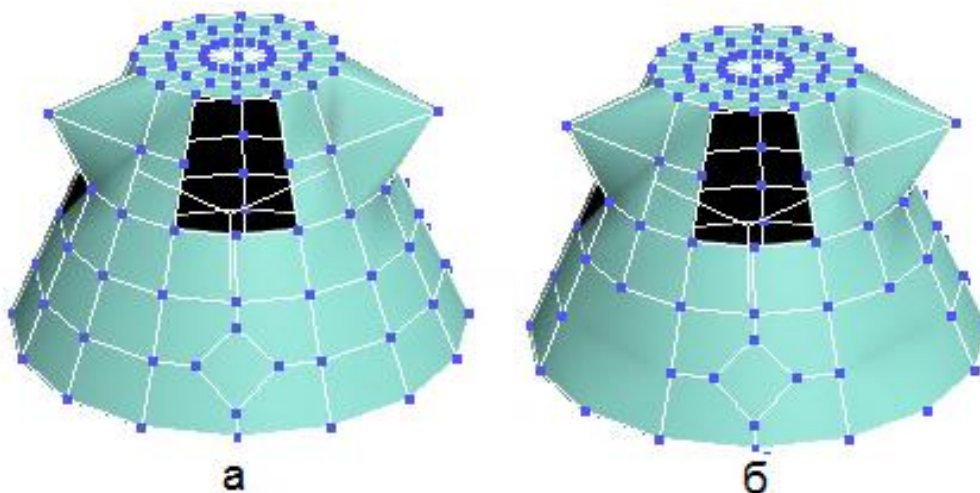


Рис. 3.11. **Видалення точок**

Для редагування геометрії використовується закладка Edit Geometry, яка дозволяє:

- Create – створювати точки, які не прив'язані полігонами до об'єкта;
- Collapse - з'єднує всі вибрані точки в одну так само, як і Weld, тільки без відстані;
- Attach – приєднувати до об'єкта будь-який інший об'єкт;

Detach - від'єднувати за допомогою обраної точки все сусідні полігони від об'єкта;

Slice Plane – створювати зріз і відповідні йому полігони;

Cut – створювати додаткові точки;

MSmooth – згладжувати сусідні полігони;

Tessellate – розбивати сусідні полігони на дрібніші.

Для згладжування об'єкта використовується вкладка Subdivision Surface. На вкладці Subdivision Surface поставимо галочку перед Use NURMS Subdivision. Початковий об'єкт прийме більш округлу форму (рис. 3.12).

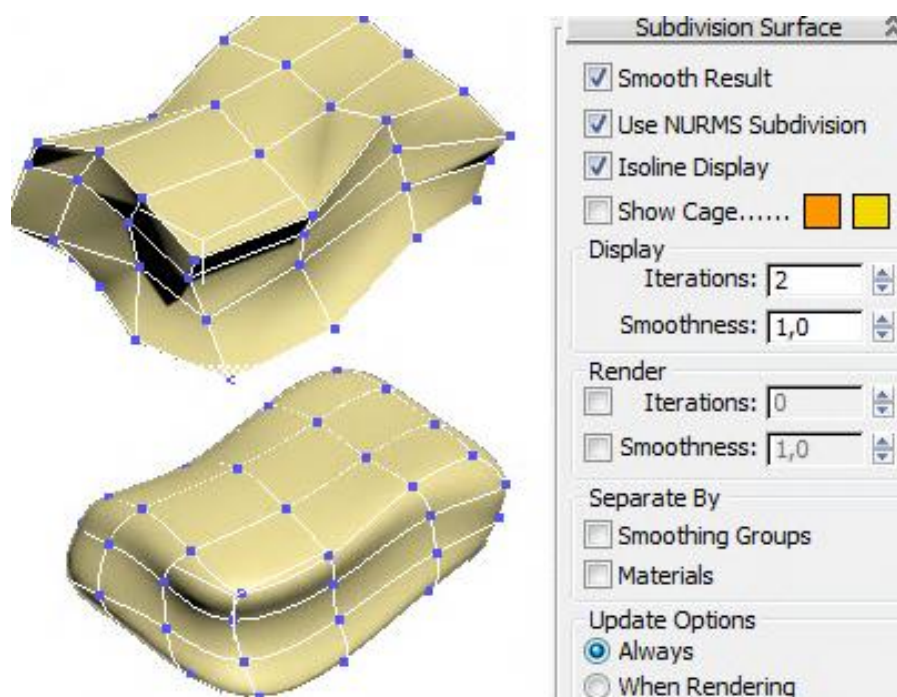


Рис. 3.12. Згладжування об'єкта

У процесі згладжування додаються нові точки і ребра. Щоб їх побачити, потрібно прибрати галочку з опції Isoline Display. Кількість точок і ребер так само залежить від параметра Iteration у групі Display.

Якщо поставити галочку перед Use NURMS Subdivision в Display, Iteration поставити 0, а в render Iteration поставити 2, то додаткових точок і ребер не буде, але при рендер (швидкий доступ "F9") об'єкт буде виглядати згладженим.

Для роботи з ребрами скористаємося об'єктом Plane, конвертований об'єкт у Editable Poly і виділимо подоб'єкт ребра. Далі

натискаємо клавішу Shift і перетягуємо ребро у інше місце або застосуємо команду Extrude (рис. 3.13).

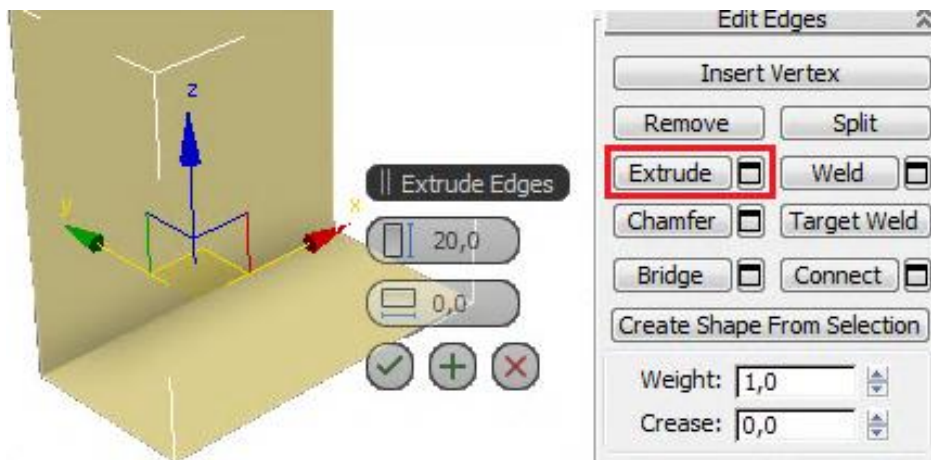


Рис. 3.13. Переміщення ребра

До полігонів застосовуються фактично ті ж операції, що і до ребер.

3.4. Полігональне моделювання на конкретних прикладах

3.4.1. Чашка з ручкою

Створимо кухоль з ручкою за допомогою Editable Poly (Редагована полісетка).

Створимо циліндр із заданими параметрами (рис. 3.14).

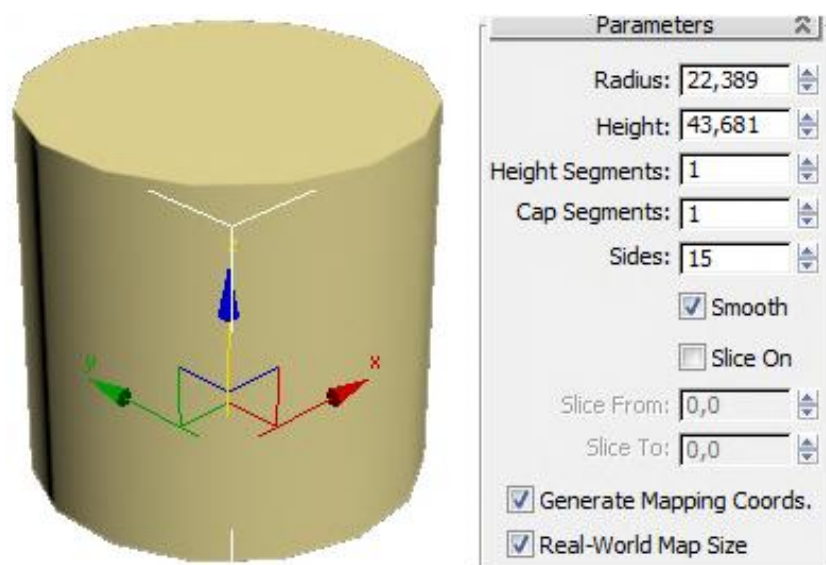


Рис. 3.14. Циліндр із заданими параметрами

За допомогою контекстного меню перетворимо циліндр у Editable Poly (рис. 3.15).

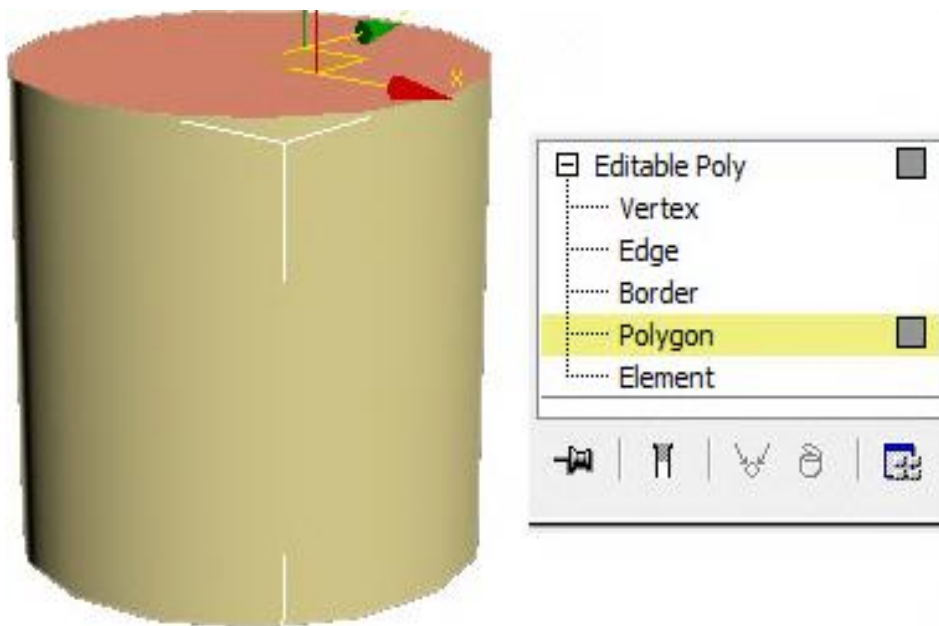


Рис. 3.15. Перетворення циліндра у Editable Poly

Далі додамо полігон у верхній частині циліндра за допомогою команди Inset (рис. 3.16).

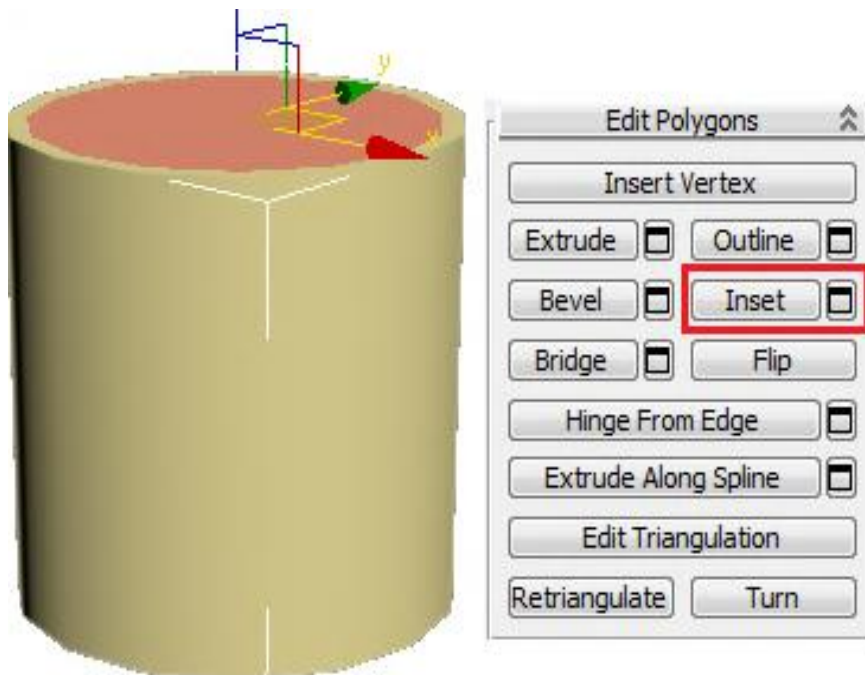


Рис. 3.16. Додавання полігону у верхню частину циліндра

За допомогою команди Extrude формуємо внутрішню частину чашки. На спливаючій командній панелі -40 означає, що витискання виконується вниз на 40 одиниць вимірювання (рис. 3.17).

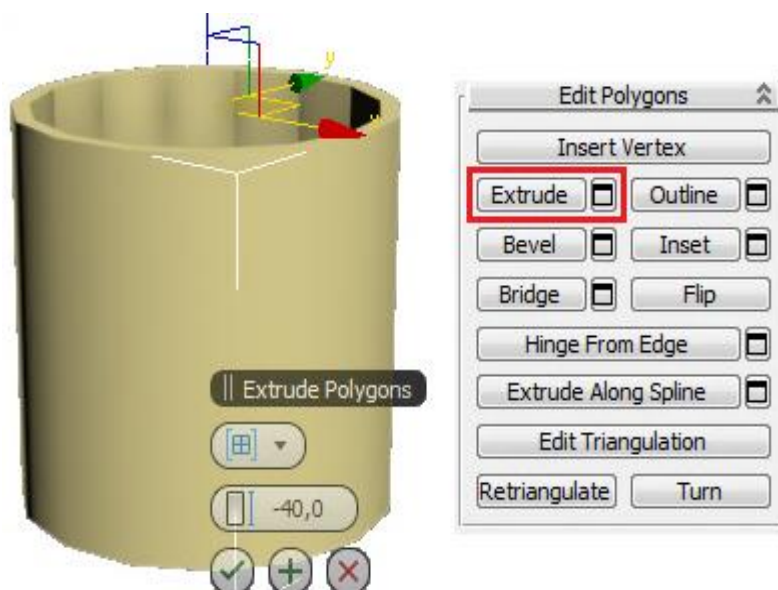


Рис. 3.17. Формування внутрішньої частини чашки

Якщо полігон вдавився занадто сильно або навпаки занадто слабо, то не знімаючи виділення, одразу вибираємо інструмент переміщення Move, "беремося" точно за вертикальну вісь і рухаємо полігон. Наскільки полігон втиснувся можна побачити на вигляді спереду або збоку.

Для того щоб сформувати ручку створимо додаткові полігони на бічній поверхні чашки. Для цього вибираємо рівень Edge (Ребра). Виділяємо всі вертикальні ребра зовні. Найпростіше виділити одне з ребер, а потім затиснути на клавіатурі кнопку Shift і клацнути по сусідньому. У правій панелі натискаємо маленьку кнопку поряд із кнопкою Connect.

На спливаючій командній панелі перша цифра означає кількість сегментів Segments: 2, друга цифра (Pinch) змінює відстань від одного сегменту до іншого і третя цифра (Slide) змінює положення сегментів. Операцію треба виконати двічі для нижньої і верхньої частин чашки (рис. 3.18).

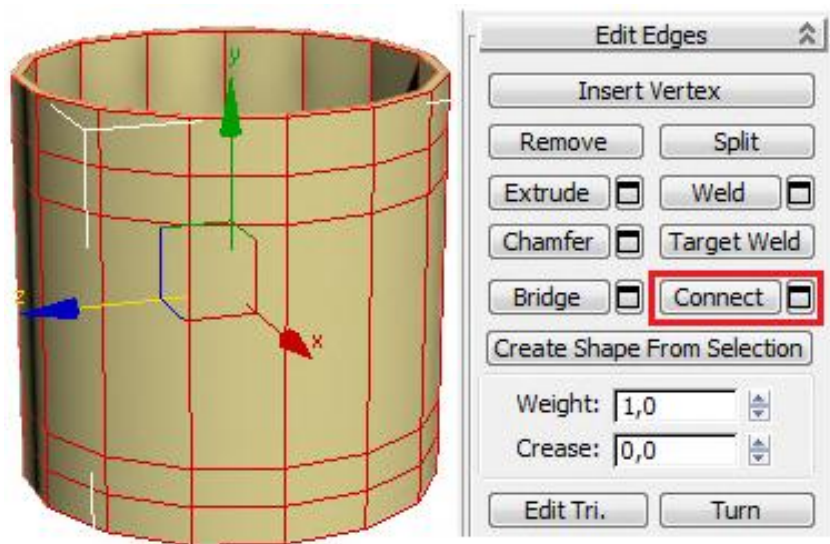


Рис. 3.18. Формування додаткових полігонів

Далі виберемо рівень полігонів (Polygon) і виділимо полігон, до якого прикріплюватиметься ручка. До виділеного полігону застосовуємо команду Extrude із заданими параметрами (рис. 3.19).

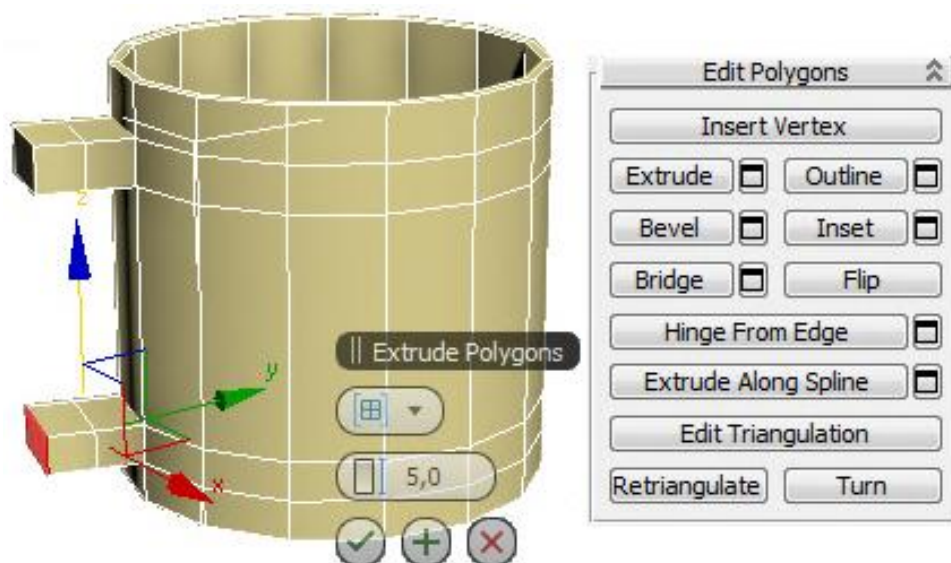


Рис. 3.19. Початкова стадія формування ручки

На рис. 3.19 бачимо два виступи, на кожному з яких є два сегменти.

У модифікаторі EditPoly виділимо рівень вершин (Vertex) і на вигляді збоку посунемо вершини так, щоб полігони на кінці виступів були приблизно один навпроти одного (рис. 3.20).

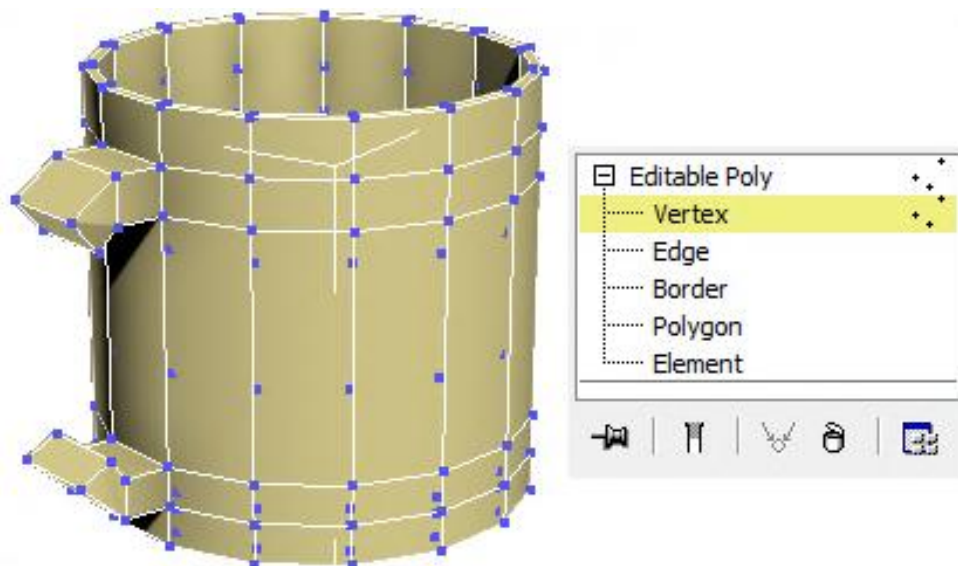


Рис. 3.20. Поворот полігонів

На завершуючому етапі формування ручки виберемо рівень полігонів (Polygon), виділимо два полігони один навпроти одного (клацнемо мишею по одному, затиснемо на клавіатурі кнопку Ctrl, клацнемо по другому). У правій панелі натиснемо кнопку Bridge, виділені полігони з'єднаються (рис. 3.21).

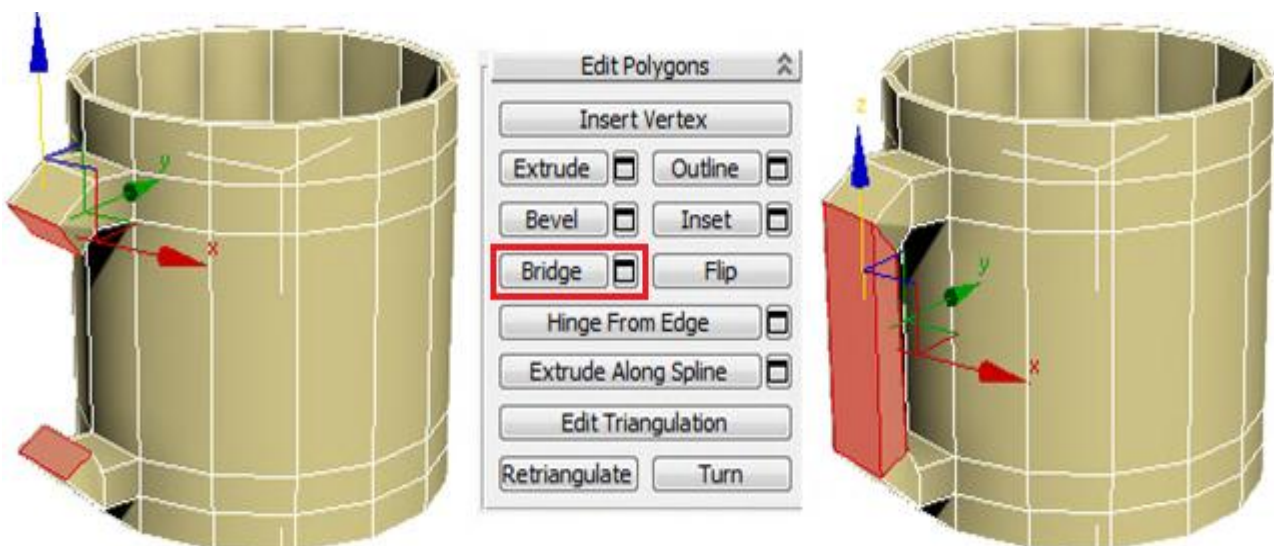


Рис. 3.21. Остаточне формування ручки

Далі застосуємо до об'єкта модифікатор MeshSmooth або TurboSmooth (він працює дещо швидше).

Встановлюємо рівень згладжування Iterations: 2.

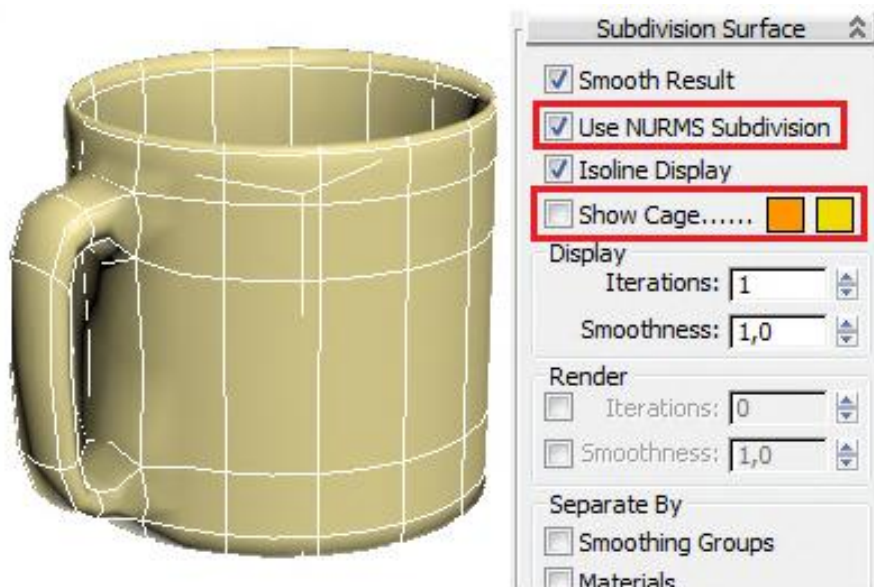


Рис. 3.22. Застосування операції згладжування

На рис. 3.22 показаний варіант згладжування об'єкта за допомогою ввімкнення Use NURMS Subdivision на вкладці Subdivision Surface.

3.4.2. Склянка

Розглянемо процес, як за допомогою EditPoly можна створити грановану склянку. Створимо циліндр із заданими параметрами (рис. 3.23).

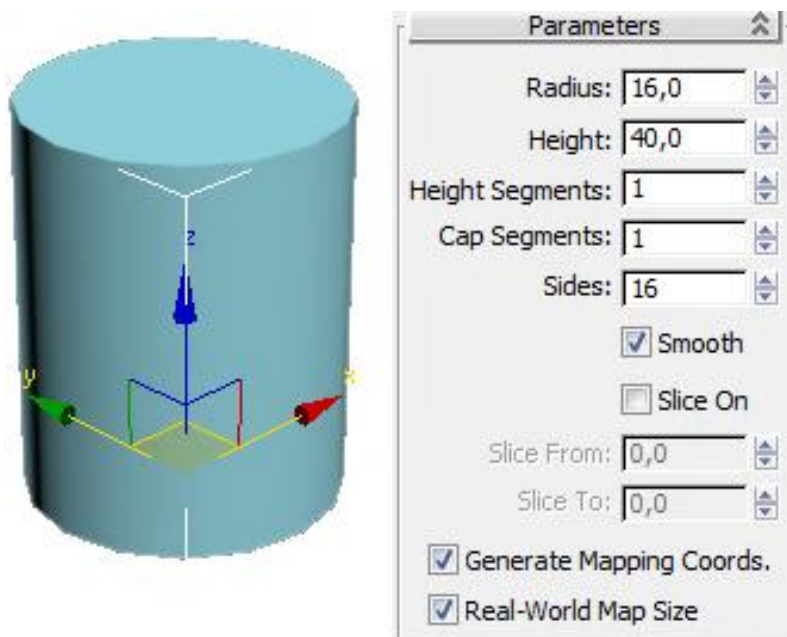


Рис. 3.23. Створення циліндра

Далі додамо полігон у верхній частині циліндра за допомогою команди Inset і за допомогою команди Extrude формуємо внутрішню частину склянки (рис. 3.24).

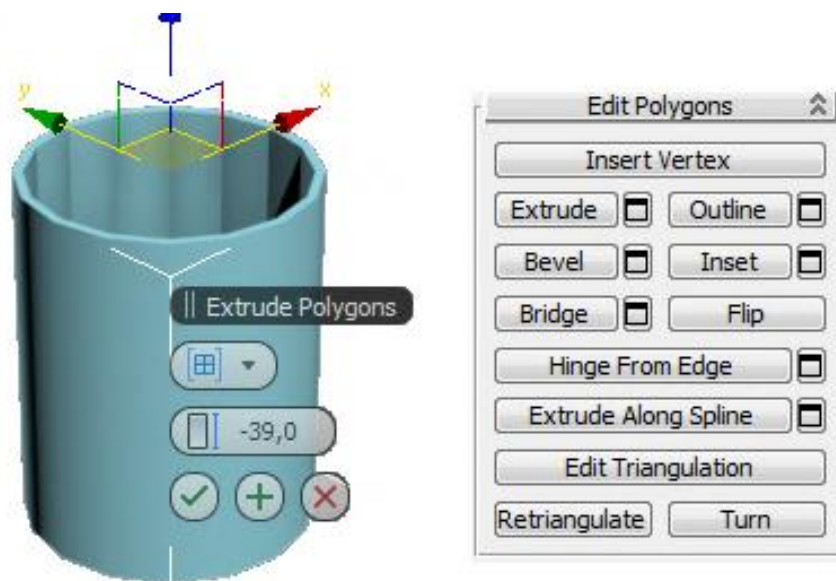


Рис. 3.24. Формування внутрішньої частини склянки

Виділимо верхню кромку склянки і за допомогою інструменту масштабування (Scale) збільшимо діаметр верхньої кромки склянки (рис. 3.25).

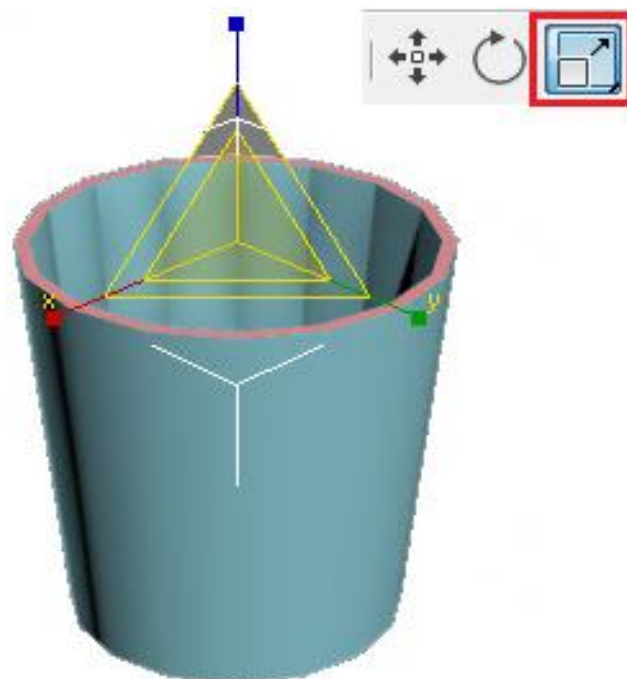


Рис. 3.25. Масштабування верхньої кромки склянки.

Згори і знизу додалися сегменти, так що під час згладжування форма об'єкта краще зберігається, точніше, стали чіткішими верхня і нижня межі (рис. 3.26).

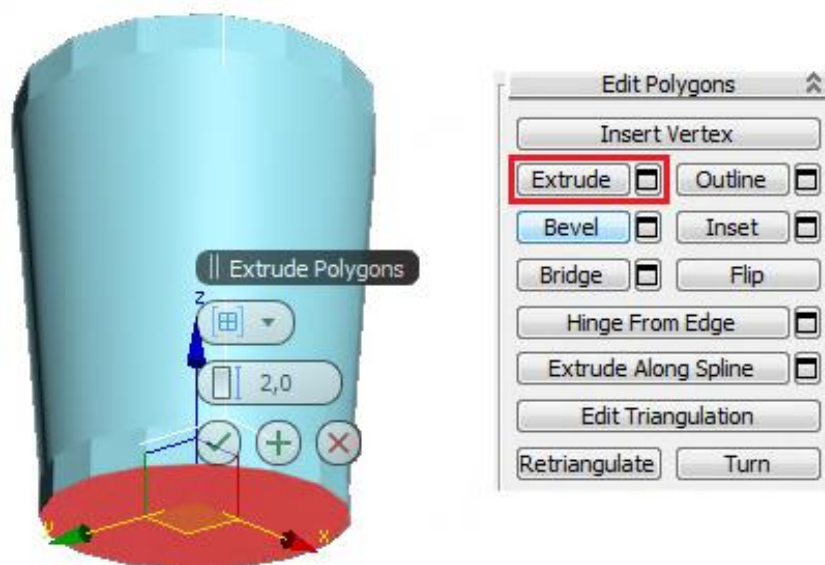


Рис. 3.26. Створення кромки у верхній і нижній частинах склянки

Для того щоб грані були чітко видно, треба виділити великі полігони збоку. Далі до виділених полігонів застосуємо команду Inset. З'являється спливаюча панель. Ставимо тип By Polygon і збільшуємо цифру, щоб з'явилися нові полігони з маленьким відступом. Натискаємо галочку ОК (рис. 3.27).

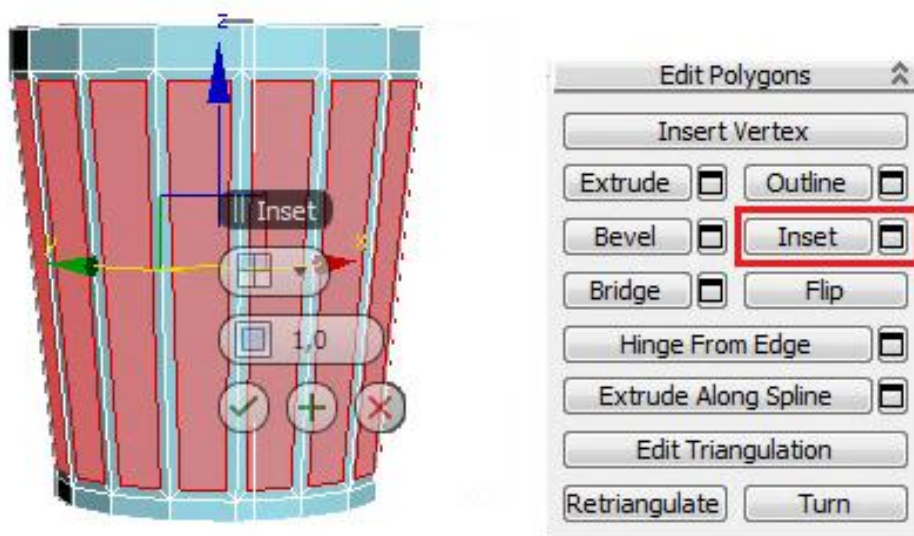


Рис. 3.27. Формування граней склянки.

Подивимося, що виходить у результаті згладжування (рис. 3.28).

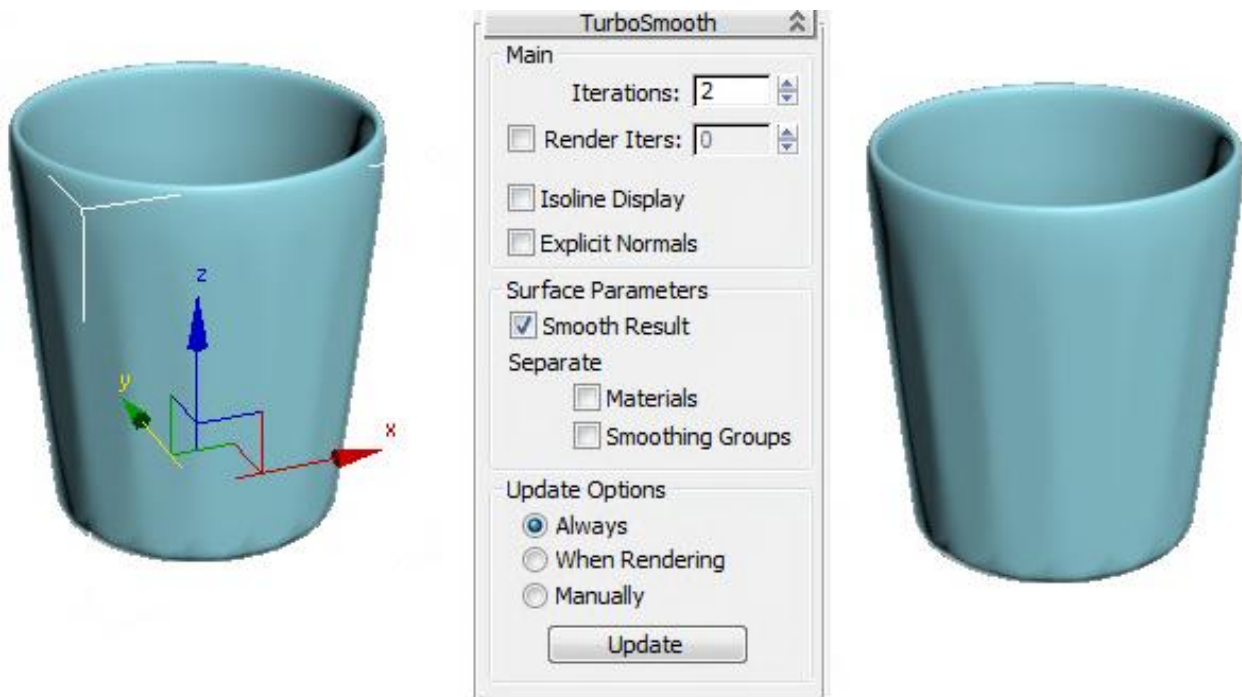


Рис. 3.28. Застосування модифікатора TurboSmooth

3.4.3. Попільничка

Створимо круглу попільничку з чотирма виїмками збоку. Сформуємо об'єкт циліндр з параметрами (рис. 3.29).

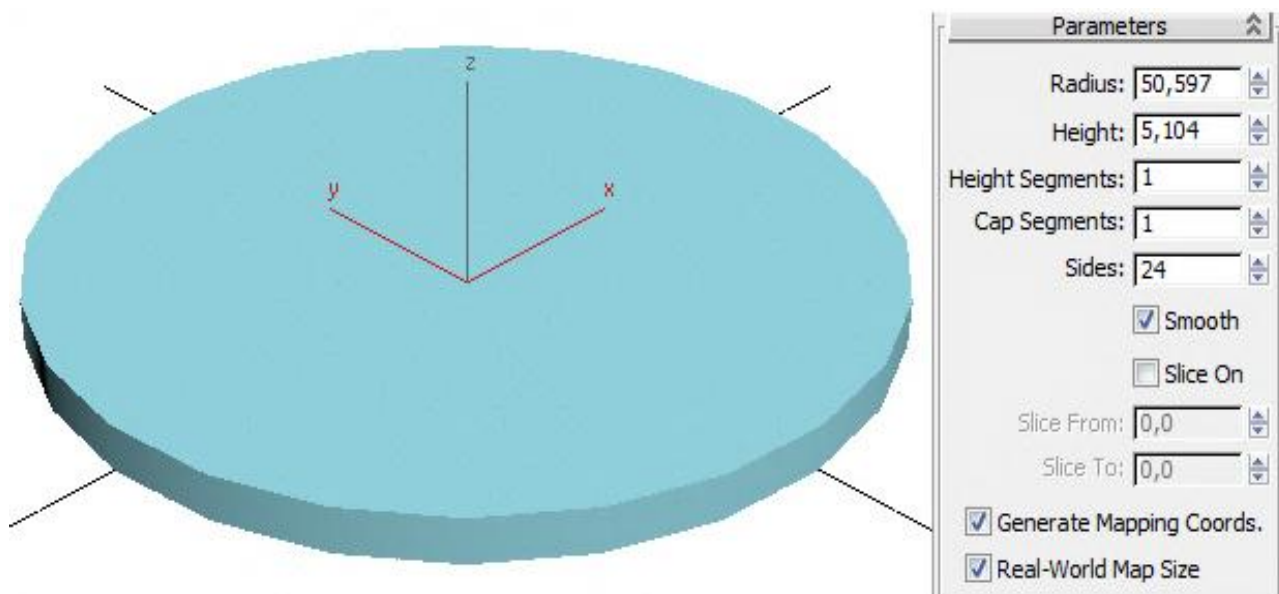


Рис. 3.29. Заготовка для попільнички

За допомогою контекстного меню перетворимо циліндр у Editable Poly і виділимо верхній полігон. Далі виберемо інструмент масштабування (Scale) і збільшимо його.

До виділеного верхнього полігону застосуємо операцію Inset (рис. 3.30).

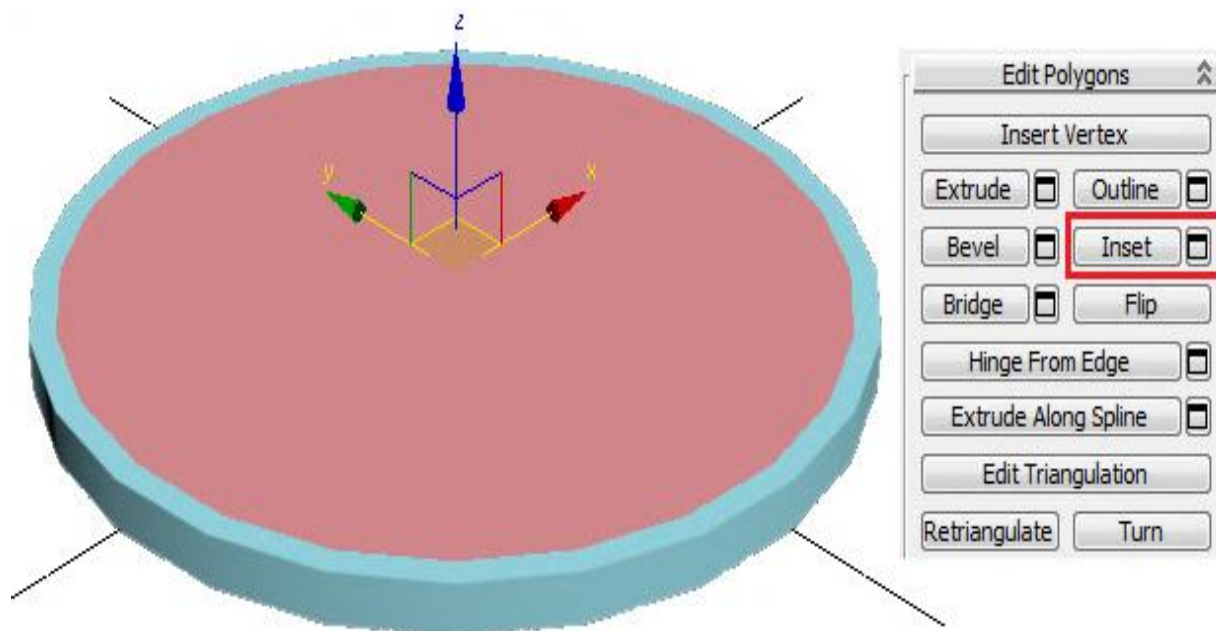


Рис. 3.30. Створення полігонів у верхній частині попільнички

Тепер потрібно виділити всі полігони на верхній частині, окрім великого центрального. Можна затиснути Ctrl і клацати по них, а можна скористатися командою Ring для полігонів: виділити один полігон, затиснути Shift і клацнути по сусідньому.

Тепер застосовуємо команду Extrude і відтискаємо все виділені полігони. Вийшов тонкий борт по краю.

Затискаємо на клавіатурі кнопку Alt (прибрати з виділення) і клацаємо по чотирьох полігонах, де будуть виїмки.

Знову робимо видавлювання Extrude виділених полігонів. Виходить борт з чотирма виїмками (рис. 3.31).

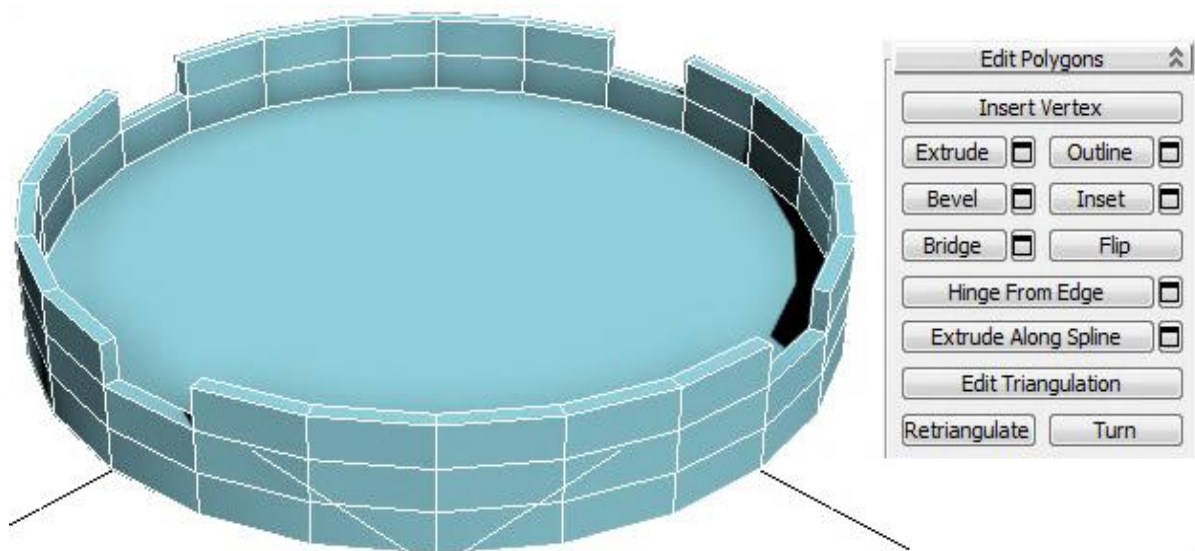


Рис. 3.31. Створення борту попільнички

Застосуємо до попільнички модифікатор MeshSmooth або TurboSmooth (рис. 3.32). Кількість рівнів згладжування Iterations: 2.

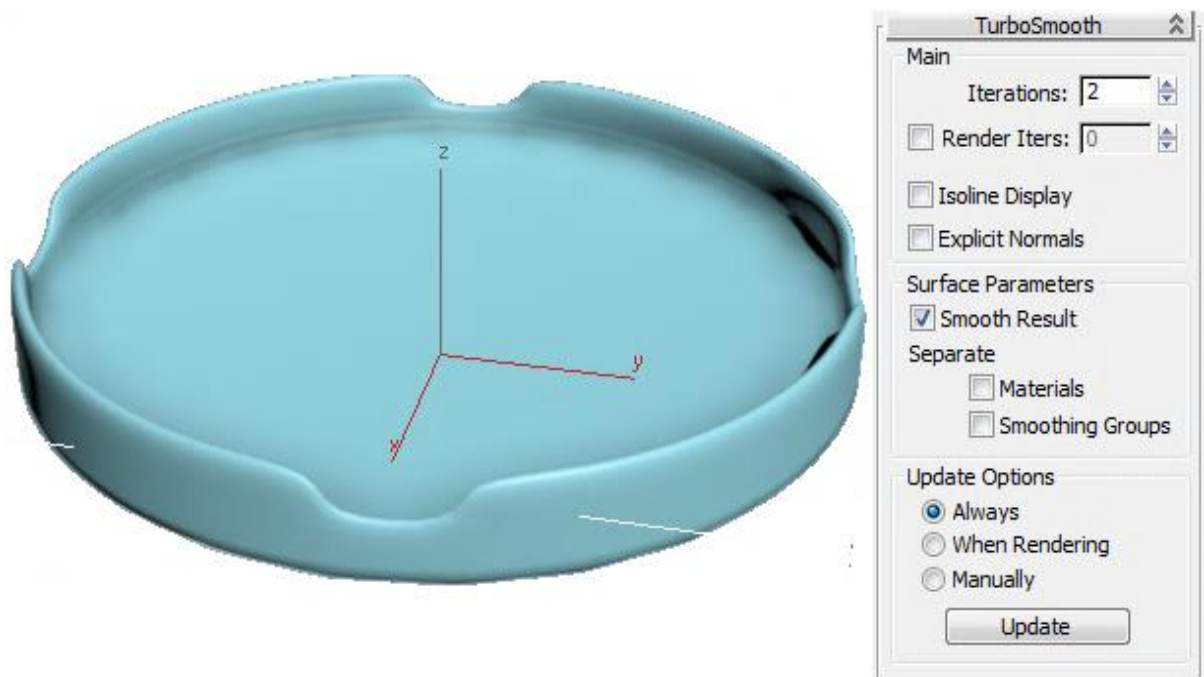


Рис. 3.32. Застосування модифікатора TurboSmooth до попільнички

3.4.4. Ложка

Спершу створимо площину – об'єкт Plane. Можна також відразу ввімкнути показ усіх сегментів (за активного вигляду перспективи слід

натиснути кнопку F4). Далі заходимо у Modify і дивимося у налаштування площини. Довжину площини можна задати такою, що дорівнює довжині майбутньої ложки (рис. 3.33).

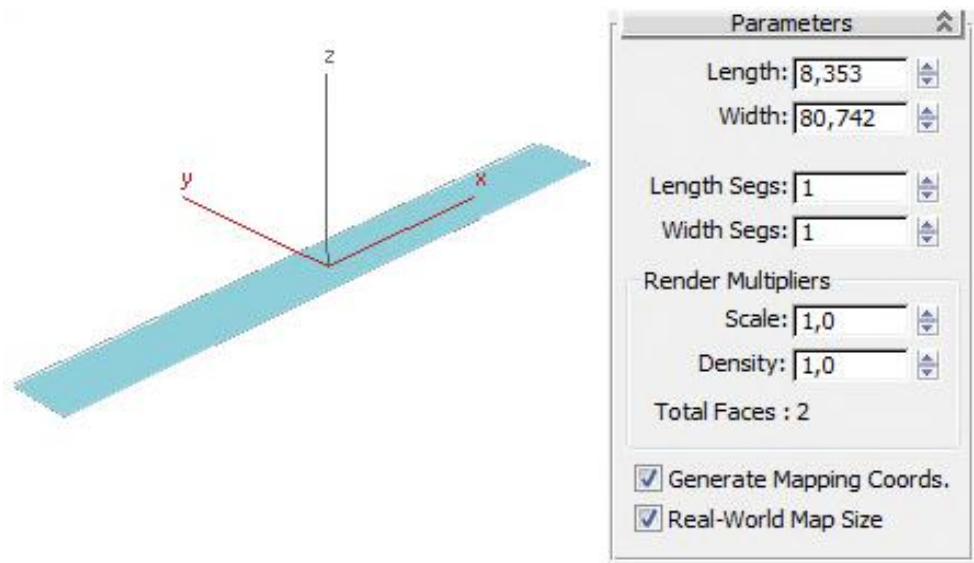


Рис. 3.33. Створення площини із заданими параметрами

За замовчуванням кількість сегментів в площині варто 4 по обох напрямках. Змінюємо кількість сегментів на 1.

Застосовуємо до площини модифікатор EditPoly і вибираємо режим Edge. Додаємо 10 сегментів по довжині за допомогою команди Connect (рис. 3.34).

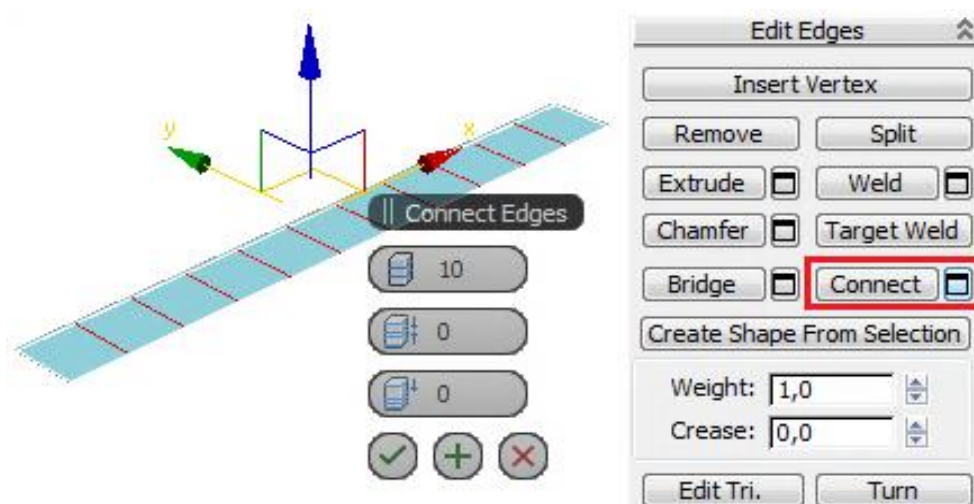


Рис. 3.34. Додавання сегментів по довжині площини

Далі виділяємо Vertex (вершини) і на вигляді зверху (чи на тому, де площина повернена на нас) починаємо рухати вершини. Вершини рухаємо тільки з одного боку, тобто робимо половину ложки, з іншого боку вершини краще навіть не виділяти, щоб випадково їх не зрушити (рис. 3.35 а).

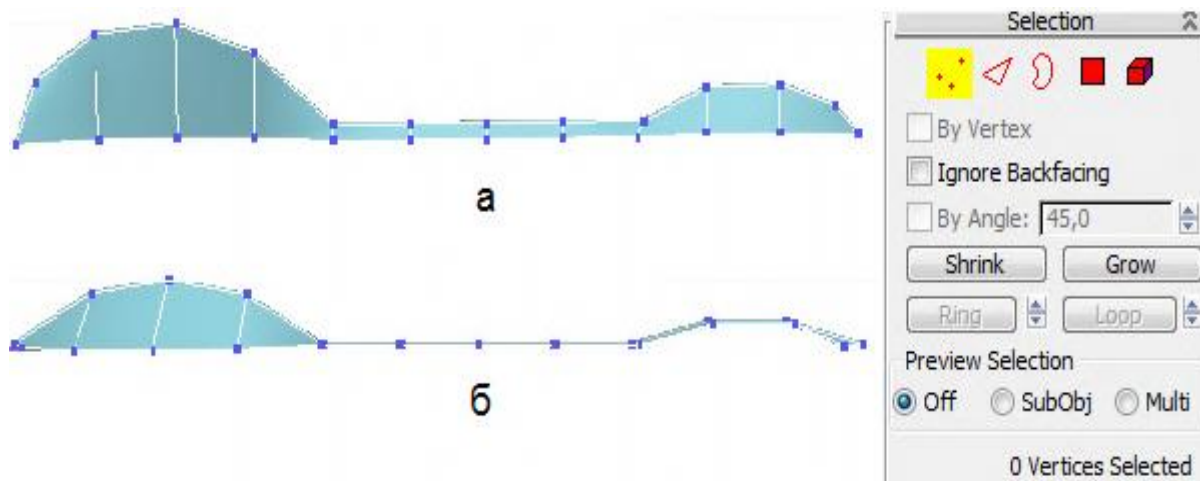


Рис. 3.35. Редагування (переміщення) вершин

Але поки що ложка виходить абсолютно плоскою. Треба виділити вершини в центрі моделі (ймовірно, в цьому випадку зручно використовувати вид перспективи) і зрушити їх вниз.

На вигляді збоку відразу видно, що вершини пішли вниз. Також видно, що ручка ложки абсолютно плоска, а ложка має бути злегка вигнута. Тому на цьому ж вигляді збоку можна виділяти вершини і зрушувати їх вгору або вниз (рис. 3.35 б).

Виділяємо одну з вершин на центральній лінії (можна вибрати будь-яку) і застосовуємо модифікатор Symmetry.

Якщо спочатку відбилосся не так, як потрібно, не треба одразу відмінити дію. У правій панелі є три галочки: X, Y, Z і галочка flip. X, Y, Z змінюють напрям площини симетрії, а flip – міняє сторону (тобто, умовно кажучи, можна віддзеркалити або праву сторону на ліву, або ліву на праву). У наведеному варіанті потрібно поставити галочку flip (рис. 3.36).

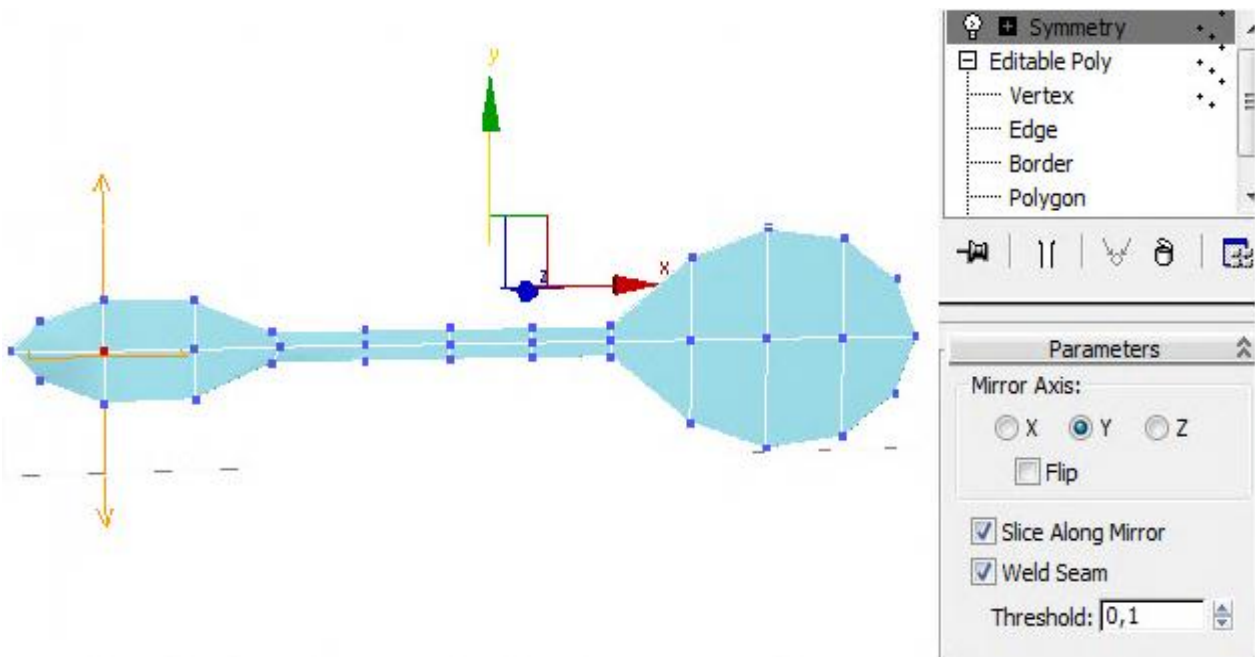


Рис. 3.36. Результат застосування модифікатора **Symmetry**

Додамо товщини ложці за допомогою модифікатора **Shell** (рис 3.37).

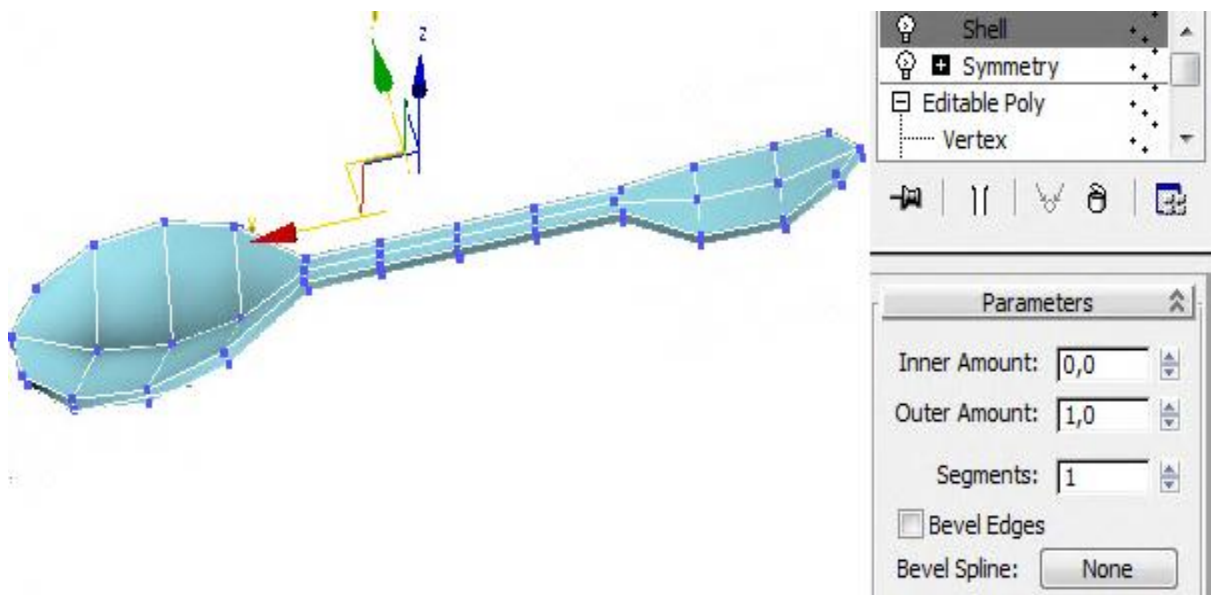


Рис. 3.37. Результат застосування модифікатора **Shell**

У параметрах модифікатора є дві цифри, що відповідають за товщину об'єкта, який вийшов: **Inner Amount** і **Outer Amount**. Вони визначають, у яку сторону (у цьому випадку вгору або вниз) додається

товщина.

Можна поставити значення, пропорційні зображеній ложці, щоб вона не виглядала ні тонкою як папір, ні занадто товстою.

Для згладжування застосовують або модифікатор MeshSmooth, або модифікатор TurboSmooth. Різниця у тому, що TurboSmooth є спрощеною версією MeshSmooth. Вона має менше налаштувань, за рахунок цього він працює швидше.

Рівні згладжування (Iterations) поставимо 2 (рис. 3.38).

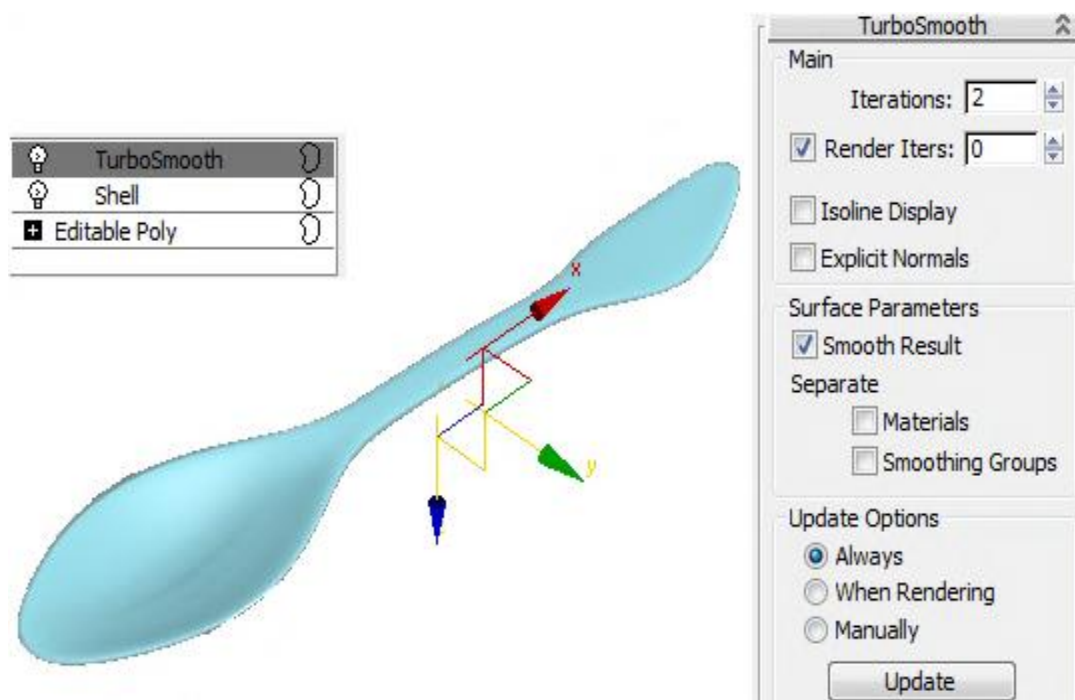


Рис. 3.38. Результат застосування модифікатора TurboSmooth

Аматори експериментувати з налаштуваннями і дивитися що виходить, збільшують iterations, після чого комп'ютер зависає від раптово з'явилися мільйонів полігонів. На великому об'єкті вже за iterations: 3 комп'ютер починає пригальмовувати, зазвичай 2 оптимальне число за співвідношенням швидкість/якість.

3.4.5. Плитка шоколаду

Почнемо з того, що створимо об'єкт Vox приблизно за габаритними розмірами плитки шоколаду (рис. 3.39).

Всі сегменти в усіх напрямках дорівнюють 1. Також можна ввімкнути показ усіх сегментів (при активному виді перспективи

натискаємо кнопку F4).

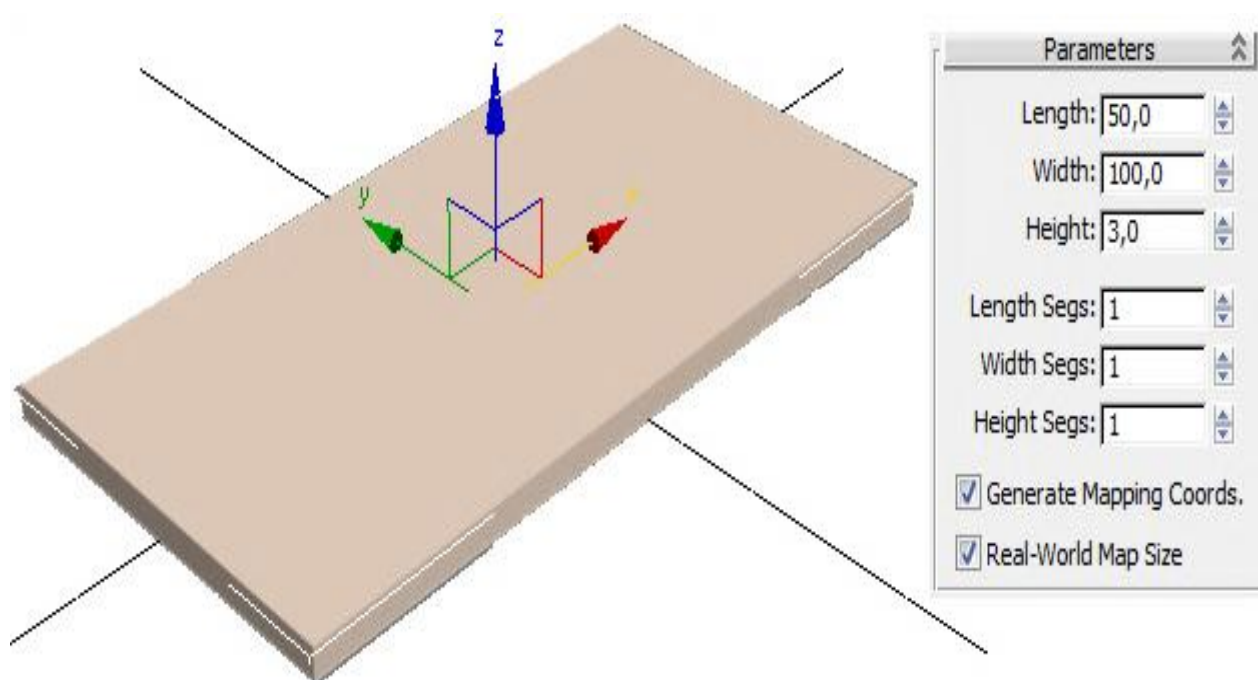


Рис. 3.39. Заготівля для плитки шоколаду

Застосуємо модифікатор EditPoly і виберемо ребра (edge). Далі виділимо 4 паралельних ребра. Є два варіанти, як це зробити. Перший: на вигляді зверху потрібно обвести мишкою середину об'єкта, не захоплюючи краю. Мають виділитися тільки 4 паралельних ребра. Другий: виділити одне з паралельних ребер, потім у правій панелі натиснути кнопку Ring (кільце). Паралельні ребра виділяться. В 3ds max 2011 або новіших додатках замість команди Ring можна виділити одне ребро, затиснути Shift і клацнути по паралельному йому ребра. Дія аналогічна до команди Ring. Після виділення 4 ребер натискаємо в правій панелі маленьку кнопочку поруч з кнопкою Connect. З'являється табличка з настройками. Перша цифра задає кількість сегментів, які додадуться між виділеними ребрами.

Далі додамо сегменти в іншому напрямку. Дії такі ж, як і в першому випадку.

Кількість сегментів у напрямках задають розміри часточок шоколаду. У прикладі в першому напрямку встановлено 5 сегментів в іншому 3 (рис. 3.40).

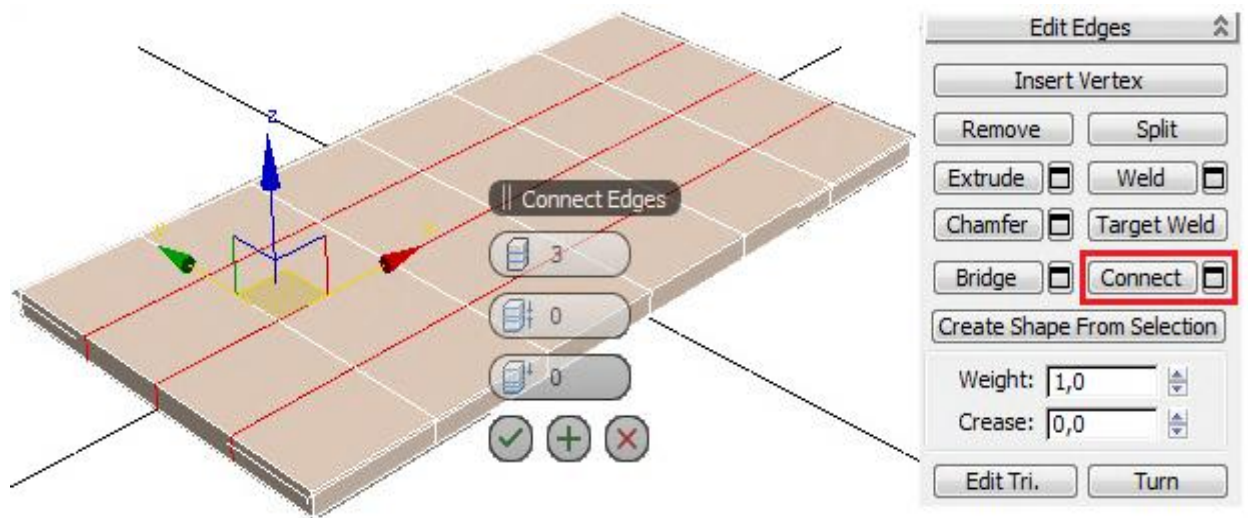


Рис. 3.40. Розподіл прямокутника на сегменти

У правій панелі у EditPoly вибираємо Polygon і виділяємо всі верхні полігони. Можна затиснути CTRL і на виді перспективи виділити всі верхні полігони.

Далі до усіх виділених полігонів застосовують команду Bevel. З'являється вікно з параметрами. Вгорі можна змінити варіанти витискання: Group (всі полігони у одному напрямку), Local Normal (кожен полігон у своєму напрямку), By Polygon (кожен полігон у своєму напрямку і окремо від інших). Вибираємо там By Polygon. Висоту (Height) збільшуємо, міру загострення (Amount) зменшуємо. Виходять часточки (рис. 3.41).

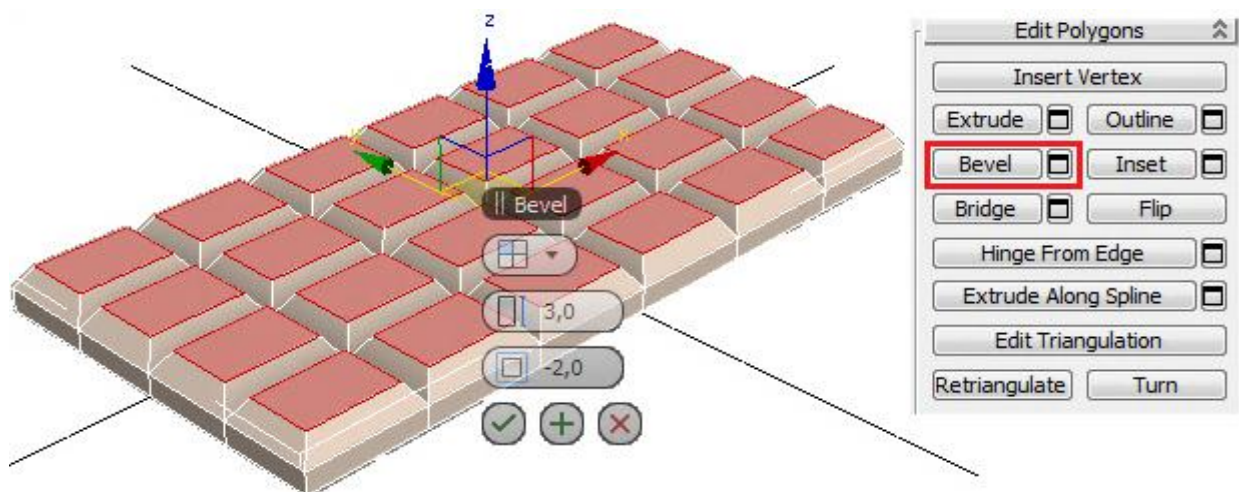


Рис. 3.41. Результат застосування операції Bevel із заданими параметрами

Застосуємо модифікатор згладжування MeshSmooth або TurboSmooth. У процесі згладжування квадрати перетворюються на круги, прямокутники у еліпси, а куби у сфери (рис. 3.42).

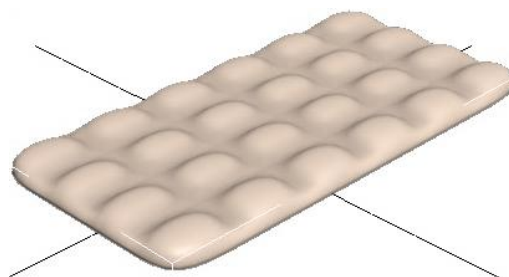


Рис. 3.42. Застосування модифікатора згладжування MeshSmooth

Повернемося до EditPoly, виберемо Edge і перевіримо відімкнення кнопки Show End Result (показати кінцевий результат), щоб знову бачити усе у не згладженому вигляді.

Затискаємо CTRL і виділяємо паралельні ребра. У цьому прикладі досить виділити 6 штук, після чого у правій панелі потрібно натиснути кнопку Ring, щоб виділилися всі паралельні ребра. Потім застосовують команду Connect. У вікні, що з'явилося, перший параметр (кількість сегментів) ставимо 2, другий параметр (Pinch) відповідає за те, ближче чи далі один від одного розташовані нові сегменти, збільшують його до 80 – 90 (але менш від 100). Чим більший параметр Pinch, тими контрастнішими будуть краї під час згладжування. Натискаємо ОК, щоб команда застосувалася. Між виділеними паралельними ребрами додалися сегменти (рис. 3.43).

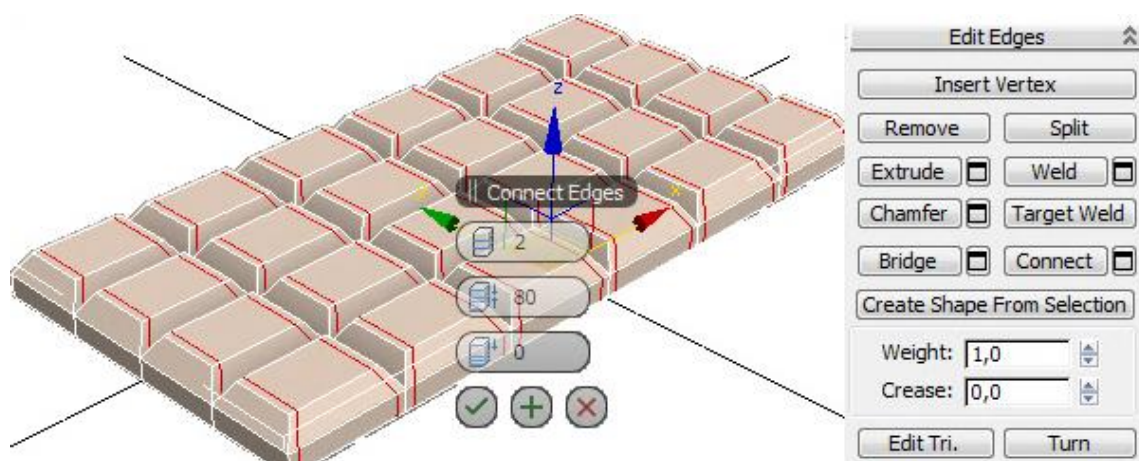


Рис. 3.43. Додавання додаткових сегментів

В одному напрямку у нас тепер більше сегментів, і у цьому напрямку форма об'єкта після застосування згладжування краща. У цьому можна переконатися, клацнувши мишею по назві модифікатора TurboSmooth (нового модифікатора застосовувати не потрібно, потрібно клацнути по вже наявному), або натиснувши кнопку Show End Result. Тепер потрібно додати більше сегментів і у іншому напрямку.

Тепер у результати згладжування виходить нормальний результат (рис. 3.44).

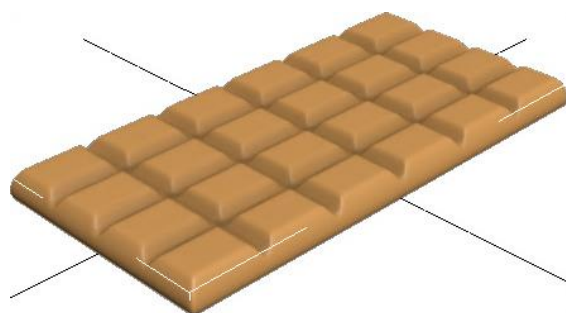


Рис. 3.44. Застосування модифікатора згладжування MeshSmooth

3.4.6. Подушка

У цьому прикладі подивимося як зробити форму подушки. Почнемо з того, що створимо об'єкт Box приблизно за габаритами майбутньої подушки (рис. 3.45).

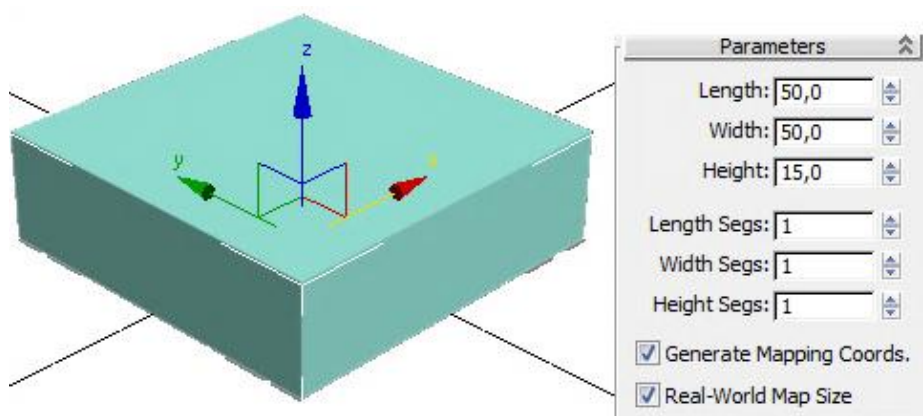


Рис. 3.45. Заготівля для подушки

Застосуємо модифікатор EditPoly і виберемо подіб'єкт Edge

(ребро). Виділимо паралельні ребра у одному з напрямків: виділяємо одне ребро, потім у правою панелі натискаємо кнопку Ring.

До виділених ребер застосуємо команду Connect з параметром Segments: 1, інші параметри дорівнюють нулю і натискаємо ОК. Так само додамо сегменти у іншому напрямку (рис. 3.46).

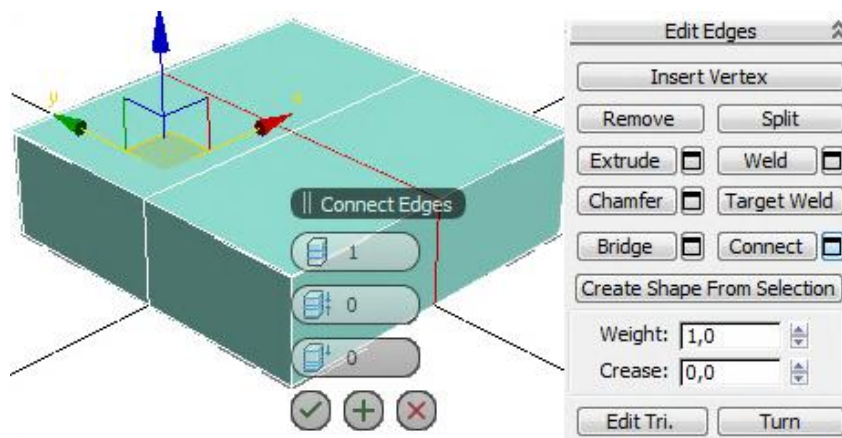


Рис. 3.46. Додавання сегментів

Як відомо, чим більше сегментів, тим реалістичнішим виходить об'єкт під час згладжування. Якщо згладити Box, то ми отримаємо в результаті щось овальне. А з додатковими сегментами кути згладяться, але форма стане вже не такою округлою.

Виділяємо у модифікаторі EditPoly подоб'єкт Vertex (вершини). У правій панелі натискаємо кнопку TargetWeld. За натиснутої кнопки TargetWeld клацаємо по верхній вершині, потім по нижній. Верхня вершина приєднується до нижньої. Повторюємо операцію для всіх кутів. У кожному куті має залишитися тільки одна вершина (рис. 3.47).

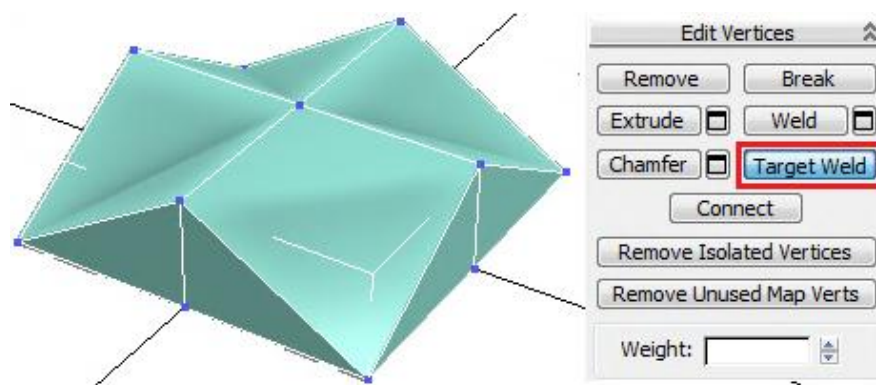


Рис. 3.47. Результат застосування команди TargetWeld

Подивимося, що виходить під час згладжування. Застосуємо модифікатор MeshSmooth або TurboSmooth. Міра згладжування Iterations: 2 (рис. 3.48).

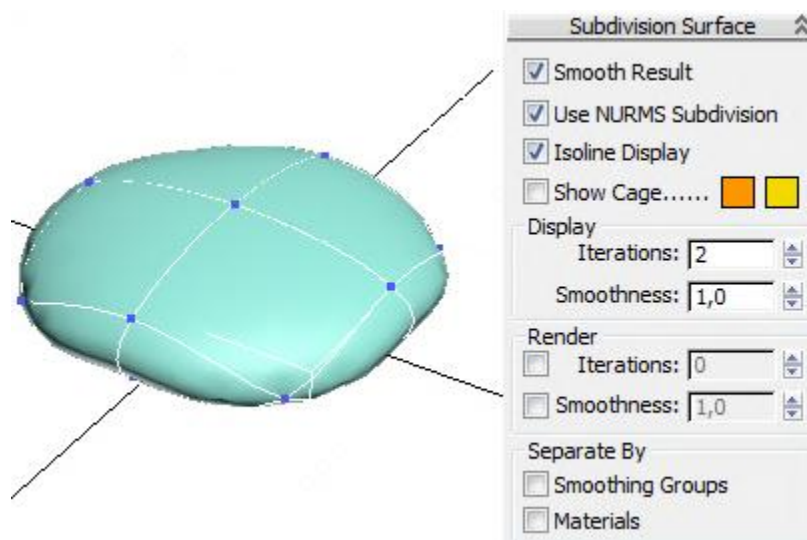


Рис. 3.48. Результат застосування модифікатора згладжування

Поки що результат не схожий на подушку.

Як відомо, чим більше сегментів, тим реалістичніша форма, тому додамо більше сегментів у кутах.

Виділяємо чотири вершини в кутах (використовуємо клавішу Ctrl). Застосуємо до виділених вершин команду Extrude з параметрами (рис. 3.49).

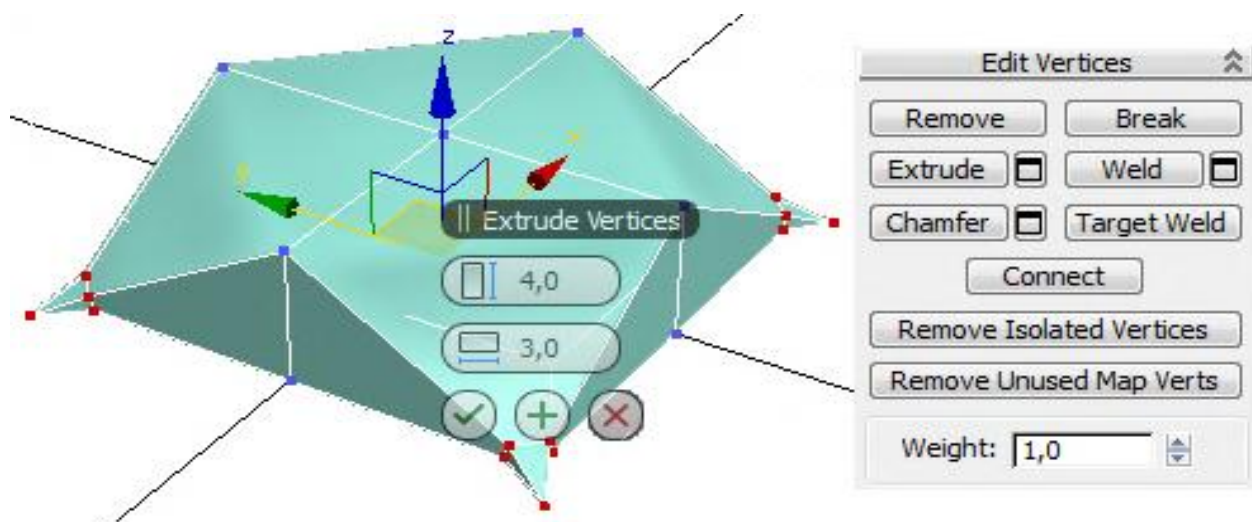


Рис. 3.49. Результат застосування команди Extrude до вершин

Дещо змінимо форму, взявши інструмент переміщення Move і рухаючи вершини. Спочатку виділимо всі вершини по кутах і дещо зрушимо кути вгору. Результат згладжування до і після переміщення вершин наведений на рис. 3.50.

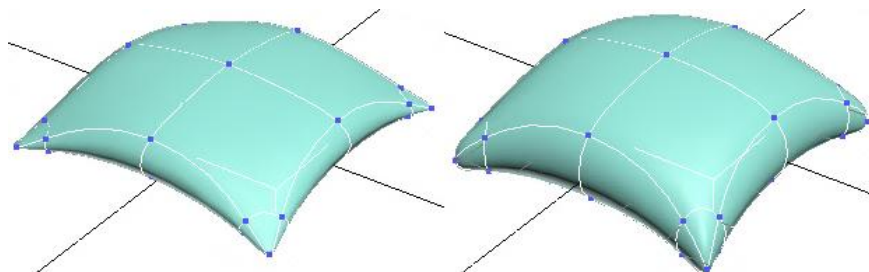


Рис. 3.50. Результат застосування модифікатора згладжування.

Якщо декілька таких подушок скласти один на одного, то вони будуть виглядати неприродно. Та і якщо одна подушка лежить на поверхні, то їй добре було би опустити кути. В модифікаторе EditPoly вибираємо подоб'єкти Vertex (вершини) виділяємо їх і рухаємо.

Також зручно при переміщенні вершин натиснути кнопку Show End Result і включити галочку Show Cage, щоб під час переміщення вершин відразу бачити кінцевий результат.

3.4.7. Тюбик

Створимо 3D модель тюбика. Хай це буде тюбик з фарбою. Почнемо, як завжди з об'єкта Box. Створимо Box за розмірами майбутнього тюбика. Головне, на що треба звернути увагу – ширина і висота повинні бути однаковими. Оскільки під час згладжування квадрати перетворюються в кола, то переріз має бути квадратним

Застосуємо модифікатор EditPoly і оберемо подоб'єкти полігон (Polygon). У правій панелі натиснемо кнопку Extrude і клацнемо по передньому полігону, щоб виділити його. Далі підведемо курсор до нього і, коли полігон поміняє вигляд, затиснемо ліву кнопку миші, зрушимо курсор дещо вгору і відпустимо кнопку миші. Додався один маленький сегмент. Тепер виберемо у верхній панелі інструмент масштабування (Scale) і зменшимо видавлений полігон.

Знову видавлюємо передній полігон і дещо зменшуємо його (або Scale, або Bevel).

Потім вибираємо Extrude (якщо використовували Bevel), підводимо

курсор до виділеного полігону, коли курсор поміняв вид, затискаємо ліву кнопку миші і ведемо курсор вниз. Полігон вдавлюється всередину (рис. 3.51).

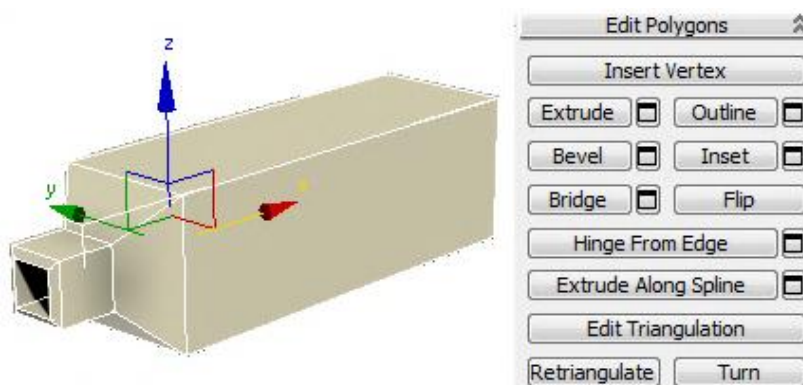


Рис. 3.51. Створення шийки тюрбика за допомогою команди Extrude і масштабування

Тепер додамо сегмент посередині. У правій панелі виберемо подоб'єкт ребро (Edge) і виділимо 4 довгі ребра. Можна виділити одне з них, а потім у правій панелі натиснути на кнопку Ring. До виділених ребер застосуємо команду Connect. Між виділеними ребрами з'явився новий сегмент. У цьому випадку параметри можна не міняти, залишивши один сегмент, і натиснути кнопку ОК.

Далі розгорнемо вигляд так, щоб бачити задню частину тюрбика, виділимо полігон на задній частині. За допомогою інструменту масштабування (Scale) стискуватимемо полігон і застосуємо до нього команду Extrude (рис. 3.52).

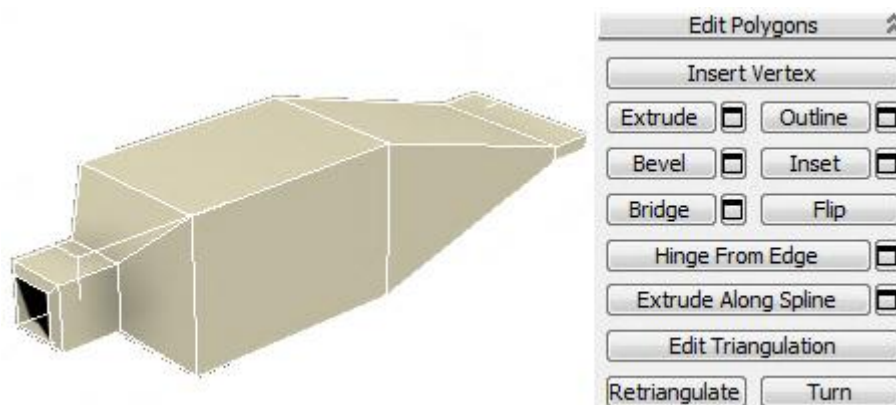


Рис. 3.52. Створення заготівлі тюрбика

Застосуємо модифікатор згладжування TurboSmooth (чи MeshSmooth) до заготівлі тубика. Рівень згладжування Iterations поставимо 2 (рис. 3.53).

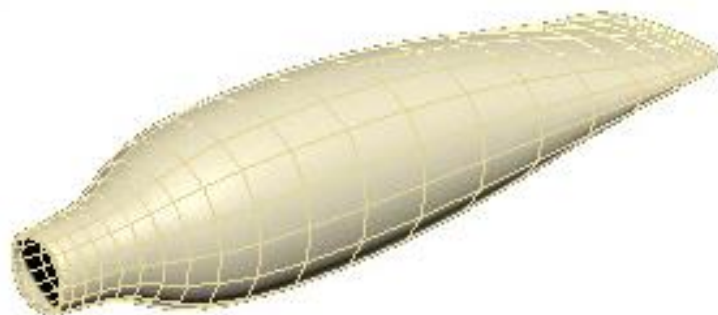


Рис. 3.53. Застосування операції згладжування до заготівлі тубика

У TurboSmooth можна також для наочності ввімкнути галочку Isoline Display. Кнопка Show End Result має бути вимкнена, щоб під час повернення на EditPoly об'єкт показувався не згладженим.

Згладжений об'єкт згладився занадто сильно. Але відомо, що чим більше сегментів, тим об'єкт контрастніший. Повертаємося до модифікатора EditPoly і вибираємо подіб'єкт ребро (Edge). Виділяємо 4 довгих прямих ребра з краю. Можна виділити одне з них, а потім правій панелі натиснути кнопку Ring. Коли всі 4 ребра виділено, у правій панелі застосуємо до них команду Connect. Третій параметр (Slide) міняємо так, щоб новий сегмент додався поряд із краєм (-90), чим ближче сегмент до краю, тим контрастніша межа (рис. 3.54).

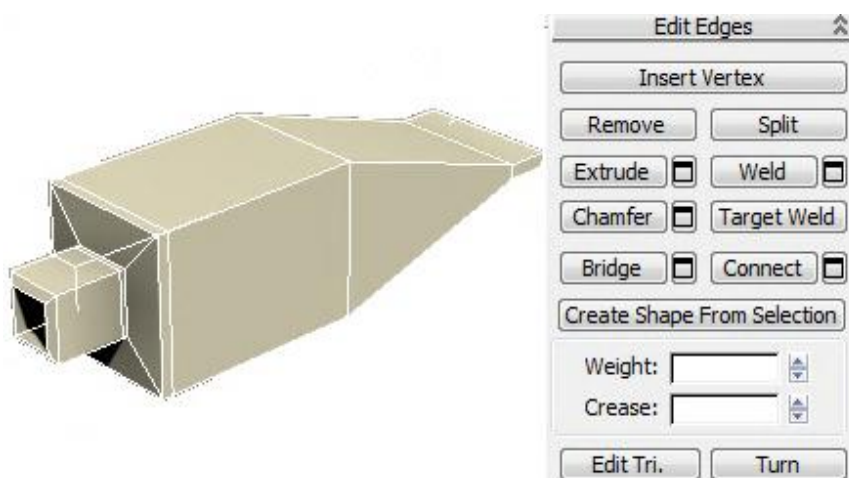


Рис. 3.54. Додавання сегменту

Після додавання сегменту межа стане контрастнішою.

Тепер додамо витікаючу фарбу. Створимо об'єкт Box таких розмірів, щоб у процесі згладжування кулька, що вийшла, співпадала за розміром з отвором шийки тюрбика (рис. 3.55).

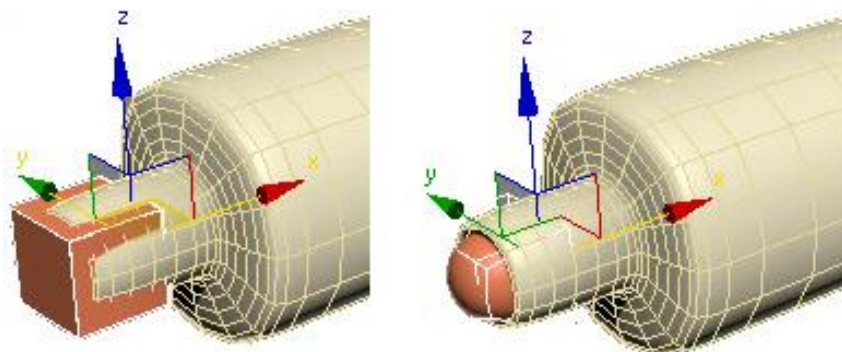


Рис. 3.55. Заготівля для витікаючої фарби

Застосуємо модифікатор EditPoly, виберемо подоб'єкт полігон (Polygon), виділимо передній полігон і командою Extrude витиснемо його назовні. Далі витиснутий полігон спрямуємо вниз. На готовому об'єкті найпростіше працювати на рівні вершин. Виділяємо подоб'єкт Vertex і на вигляді збоку рухаємо вершини так, щоб полігон був спрямований вниз (рис. 3.56 а). Полігон, що вийшов, витискаємо ще раз, щоб він торкнувся умовної поверхні, на якій лежить тюрбик (рис. 3.56 б). Тепер виділяємо бічний полігон і командою Extrude витискаємо його вбік (рис. 3.50 в).

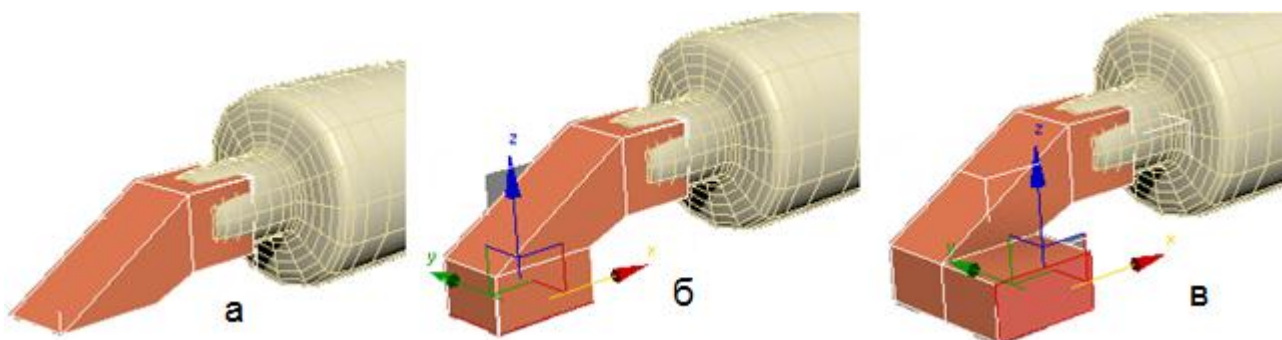


Рис. 3.56. Послідовність формування витікаючої фарби

Виділяємо два передні полігони (клацаємо по них мишею, затиснувши кнопку Ctrl) і теж витискаємо їх (рис. 3.57 а). Виділяємо два

полігони з іншого боку і також витискаємо їх (рис. 3.57 б). Далі витиснемо один із полігонів спереду і збільшимо його інструментом масштабування (рис. 3.57 в).

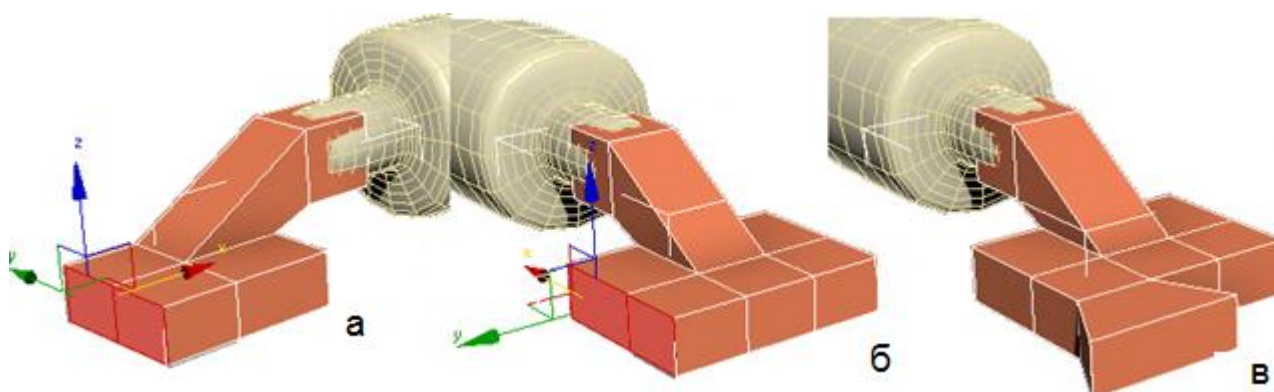


Рис. 3.57. Продовження послідовності формування витікаючої фарби

Форма в даному випадку потрібна дуже приблизна, тому що для додання реалістичності все одно треба буде рухати вершини, щоб уникнути дуже гладкої форми.

Подивимося, що виходить у процесі згладжування. Не обов'язково, щоб фарба, що розтеклася, була плоскою, можна також посувати вершини в горизонтальній площині (рис. 3.58).

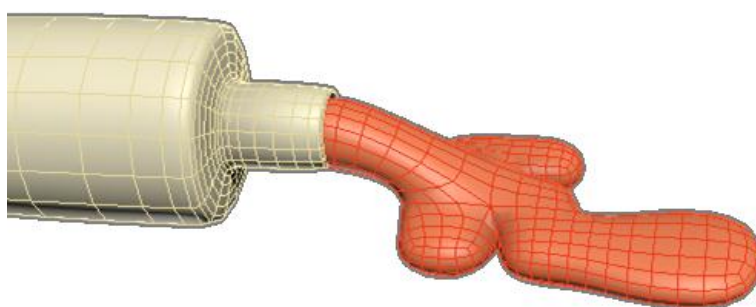


Рис. 3.58. Застосування модифікатора згладжування до витікаючої фарби

Якщо знизу, де "фарба" розтікається по поверхні, вона виходить не занадто реалістичною, то можна додати додатковий сегмент.

Тепер додамо кришку для тубика. Створимо циліндр (Cylinder).

У кількості сегментів по висоті Height Segments замість 5

ставимо 1. Застосовуємо модифікатор EditPoly.

Виділяємо всі бічні полігони і застосовуємо до них команду Bevel. З'являються параметри. Вгорі вибираємо варіант By Polygon (тобто кожен полігон видавлюватиметься по отдельности).

Збільшуємо висоту і дещо зменшуємо міру загострення, натискаємо ОК. Згладжування, як завжди, без додаткових сегментів вийшло занадто сильним. Вибираємо подоб'єкт ребро (Edge), виділяємо всі бічні ребра і застосовуємо команду Connect. Кількість сегментів: 2, Pinch більше 80-90, Slide: 0.

Тепер під час згладжуванні форма вже не така округла.

Вибираємо рівень полігонів, виділяємо верхній полігон і за допомогою команди Inset (чи Bevel з маленькою висотою) робимо невеликий відступ усередину. Потім за допомогою Bevel дещо видавлюємо полігон вниз і зменшуємо його.

Загальний результат набуде такого вигляду (рис. 3.59).

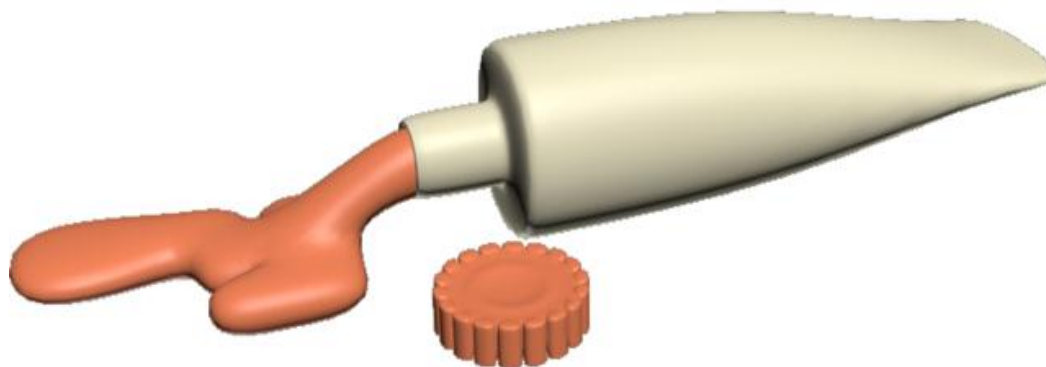


Рис. 3.59. Тюбик з витікаючою фарбою

3.4.8. Колона

У цьому прикладі подивимося, якими засобами можна створити колону. Процес створення колони дуже нагадує створення гранованої склянки.

Створимо циліндр приблизно за розмірами колони. Кількість сегментів по висоті Height Segments: 1. Кількість сторін Sides дорівнює кількості граней. Також для зручності відображення можна ввімкнути на вигляді перспективи показ сегментів (за активного вигляду перспективи натиснути кнопку F4).

Застосуємо модифікатор EditPoly, виберемо подоб'єкт – полігон

(Polygon) і виділимо верхній полігон.

В даному прикладі, у основному, використовувалася команда Bevel, Extrude. За допомогою цих команд формується верхня і нижня частині колони (рис. 3.60).

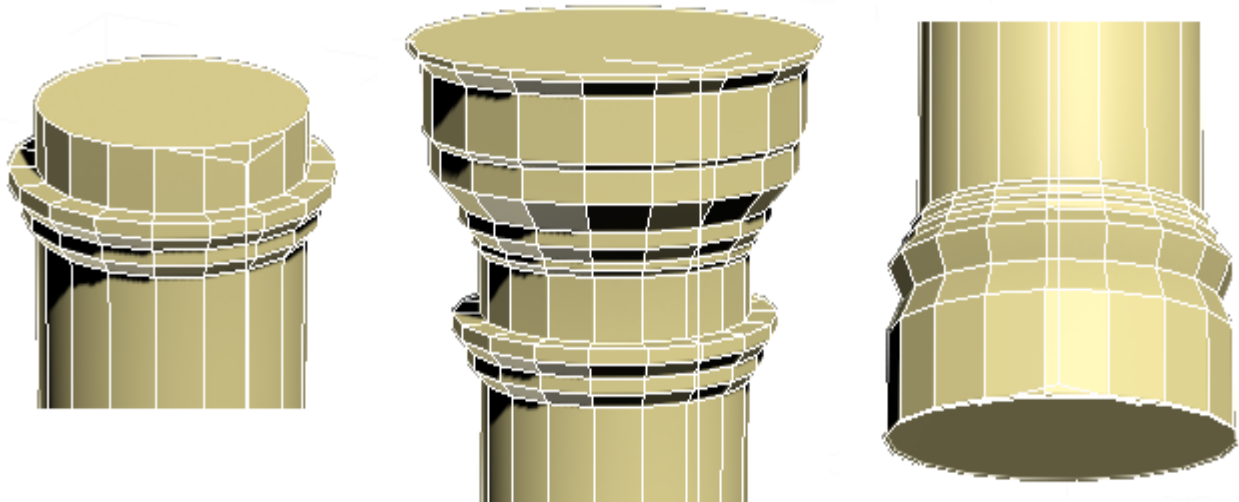


Рис. 3.60. Верхня і нижня частині колони

Щоб профіль колони був контрастнішим, треба створити додаткові сегменти виступів і западин.

Виділимо всі довгі вертикальні полігони: у модифікаторі EditPoly виберемо подоб'єкт полігон (Polygon), а далі або обведемо мишею середину колони, або виділимо один полігон, затиснемо Shift і клацнемо мишею по сусідньому (практично аналогічно до команди Ring для ребер).

У правій панелі натиснемо маленьку кнопку поряд із кнопкою Bevel (з'явиться панель налаштування витискання).

У верхній частині налаштувань виберемо тип витискання By Polygon, висоту витискання залишаємо нульовою, розмір загострення (Bevel Amount) дещо зменшуємо. Розмір загострення маленький, чим ближче параметр загострення до нуля, тим гостріший буде результат під час згладжування.

Натиснемо внизу на галочку ОК, щоб результат зафіксувався.

Застосовуємо модифікатор згладжування MeshSmooth або TurboSmooth. Перевіряємо вимкнення кнопки Show End Result, рівні згладжування Iterations залишаємо такими, що дорівнюють 2 (рис. 3.61).

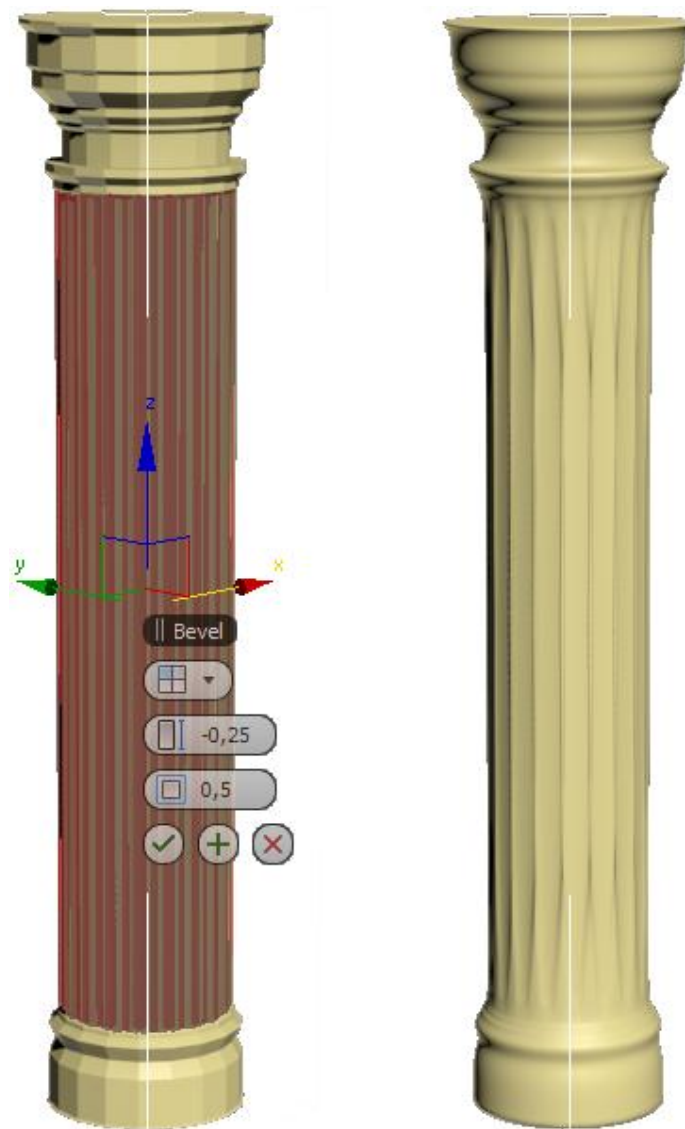


Рис. 3.61. Середня частина колони

Контрольні запитання

1. Основні типи 3D-моделей.
2. Способи подання моделей геометричних об'єктів.
3. Полігональна модель і патч-модель 3D-об'єктів.
4. Параметричне задавання геометричних об'єктів.
5. Редагована поверхня Editable Mesh.
6. Редагована поверхня Editable Poly.
7. Редагована поверхня Editable Patch.
8. Редагована поверхня NURBS Surface.
9. Перетворення об'єктів 3ds max у різні типи поверхонь.
10. Режими редагування поверхонь.

4. Сплайн 3D-графіка

4.1. Інтерполяція сплайна

Крива, що описує деформацію гнучкої лінійки, зафіксованої у окремих точках, є сплайном. Інакше кажучи, сплайн використовується для визначення функції профілю між точками даних.

Сплайни мають виключно гарні апроксимативні властивості, універсальності і забезпечують простоту реалізації обчислювальних алгоритмів, отриманих на їх основі. Тобто сплайн – не вигадана математична абстракція, а в багатьох випадках розв'язання диференціальних рівнянь, що описують цілком реальні фізичні процеси (наприклад, у механіці це деформація гнучкої пластини або стрижня, зафіксованого у окремих точках).

Сплайни – це гладкі (що мають декілька безперервних похідних) кусково-поліноміальні функції, які можуть бути використані для подання функцій, заданих великою кількістю значень, для яких непридатна апроксимація одним поліномом. Оскільки сплайни гладкі, економічні й легкі у роботі, вони використовуються під час побудови довільних функцій для:

- моделювання кривих;
- апроксимації даних за допомогою кривих;
- виконання функціональних апроксимацій;
- розв'язання функціональних рівнянь.

Розглянемо завдання проведення гладких кривих за заданими граничним точкам, або завдання інтерполяції. Оскільки через дві точки можна провести скільки завгодно багато гладких кривих, то для вирозв'язання цього задавання необхідно обмежити клас функцій, які визначатимуть шукану криву. Математичними сплайнами називають функції, використовувані для апроксимації кривих. Важливою їх властивістю є простота обчислень. На практиці часто використовують сплайни у виді поліномів третьої міри. З їх допомогою досить зручно проводити криві, які інтуїтивно відповідають людському суб'єктивному поняттю гладкості.

Розглянемо спочатку функцію сплайна для побудови графіку функції однієї змінної. Нехай на площині задана послідовність точок $\{x_i, y_i\}$, $i = 0, 1, 2, \dots, m$, $x_0 < x_1 < \dots, x_{m-1} < x_m$. Визначимо шукану функцію $y = S(x)$ (рис. 4.1), причому поставимо дві умови:

функція має проходити через всі точки $y_i = S(x_i)$, $i = 0, 1, 2, \dots, m$;
 функція має двічі безперервно диференціюватися, тобто мати безперервну другу похідну на усьому відрізку $[x_0, x_m]$.

На кожному з відрізків $[x_i, x_{i+1}]$, $i = 0, 1, 2, \dots, m - 1$, шукатимемо функцію у вигляді полінома третьої міри (4.1):

$$S_i(x) = \sum_{j=0}^3 (x - x_i)^j \cdot a_{i,j}. \quad (4.1)$$

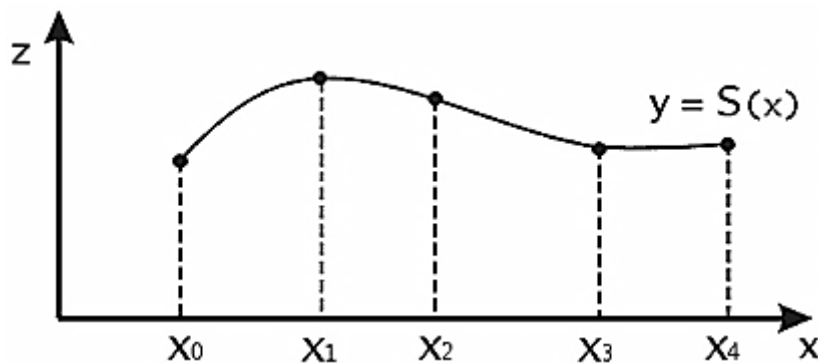


Рис. 4.1. Функція сплайна

4.2. Поліноміальна сплайн-інтерполяція

Задавання побудови полінома зводиться до знаходження коефіцієнтів a_{ij} . Оскільки для кожного з відрізків $[x_i, x_{i+1}]$ необхідно знайти 4 коефіцієнти a_{ij} , а всього кількість шуканих коефіцієнтів дорівнюватиме $4m$. Для знаходження всіх коефіцієнтів визначимо відповідну кількість рівнянь. Перші $(m - 1)$ рівнянь отримуємо з умов збігу значень функції у внутрішніх вузлах x_i , $i = 1, 2, \dots, m - 1$. Такі $2(m - 1)$ рівнянь отримуємо аналогічно з умов збігу значень перших і других похідних у внутрішніх вузлах. Разом з першою умовою отримуємо $m - 1 + m - 1 + m - 1 + m + 1 = 4m - 2$ рівнянь. Відсутні два рівняння можна отримати завданням значень перших похідних в кінцевих точках відрізка $[x_0, x_m]$. Так можуть бути задані граничні умови.

Перейдемо до складнішого випадку – задавання кривих у тривимірному просторі. У разі функціонального задавання кривої $y = f(x)$, $z = f(x)$ можливі багатозначності в разі самопересечень і незручності коли значення похідних дорівнюють ∞ . Зважаючи на це шукатимемо функцію у параметричному вигляді. Нехай t – незалежний

параметр, такий, що $0 \leq t \leq 1$. Кубічним параметричним сплайном назовемо наступну таку систему рівнянь (4.2):

$$\begin{aligned} x(t) &= a_x \cdot t^3 + b_x \cdot t^2 + c_x \cdot t + d_x; \\ y(t) &= a_y \cdot t^3 + b_y \cdot t^2 + c_y \cdot t + d_y; \\ z(t) &= a_z \cdot t^3 + b_z \cdot t^2 + c_z \cdot t + d_z. \end{aligned} \quad (4.2)$$

Координати точок на кривій описуються вектором $[x(t), y(t), z(t)]$, а три похідні задають координати відповідного дотичного вектора в точці. Наприклад, для координати x (4.3):

$$\frac{d}{dt}x = 3a_x t^2 + 2 \cdot b_x \cdot t + c_x. \quad (4.3)$$

Одним із способів задавання параметричного кубічного сплайна є зазначення координат початкової і кінцевої точок, а також векторів, дотичних до них. Такий спосіб задавання називається формою Ерміта. Позначимо кінцеві точки P_1 і P_4 , а дотичні вектори у них R_1 і R_4 . Індекси обрані з урахуванням подальшого викладення матеріалу.

Вирішуватимемо завдання знаходження четвірки коефіцієнтів a_x, b_x, c_x, d_x , оскільки для двох рівнянь, що залишилися, коефіцієнти знаходяться аналогічно. Запишемо умову для побудови сплайна (4.4):

$$x(0) = P_{1x}, x(1) = P_{4x}, x'(0) = R_{1x}, x'(1) = R_{4x} \quad (4.4)$$

Перепишемо вираз для (4.2) у векторному вигляді (4.5).

$$x(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}_x. \quad (4.5)$$

Позначимо вектор рядок – T і вектор стовпець коефіцієнтів – C_x , тоді $x(t) = TC_x$. З (4.4) і (4.5) випливає (4.6):

$$\begin{aligned}x(0) &= P_{1x} = (0 \ 0 \ 0 \ 1) \cdot C_x; \\x(1) &= P_{4x} = (1 \ 1 \ 1 \ 1) \cdot C_x.\end{aligned}\tag{4.6}$$

Для дотичних отримаємо (4.7):

$$\begin{aligned}x'(0) &= R_{1x} = (0 \ 0 \ 1 \ 0) \cdot C_x; \\x'(1) &= R_{4x} = (3 \ 2 \ 1 \ 0) \cdot C_x.\end{aligned}\tag{4.7}$$

Звідси матимемо векторно-матричне рівняння (4.8):

$$\begin{pmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \cdot C_x.\tag{4.8}$$

Розв'язуємо систему відносно C_x (4.9):

$$C_x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} P_{1x} \\ P_{4x} \\ R_{1x} \\ R_{4x} \end{pmatrix} = M_h \cdot G_{hx}.\tag{4.9}$$

Тут M_h – ермітова матриця, G_{hx} – геометричний вектор Эрмита. Підставимо вираз C_x для знаходження $x(t)$, $y(t)$ $z(t)$ і (4.10):

$$\begin{aligned}x(t) &= T \cdot M_h \cdot G_{hx}; \\y(t) &= T \cdot M_h \cdot G_{hy}; \\z(t) &= T \cdot M_h \cdot G_{hz}.\end{aligned}\tag{4.10}$$

Випишемо у явному вигляді формули для обчислення координат точок сплайна (4.11):

$$\begin{aligned}
 x(t) &= T \cdot M_H \cdot G_{HX} = \\
 &= P_{1X}(2 \cdot t^3 - 3 \cdot t^2 + 1) + P_{4X}(-2 \cdot t^3 + 3 \cdot t^2) + \\
 &+ R_{1X}(t^3 - 2 \cdot t^2 + t) + R_{4X}(t^3 - t^2).
 \end{aligned}
 \tag{4.11}$$

Чотири функції у дужках називаються функціями сполучення.

Форму кривої, заданої у формі Ерміта, легко змінювати, якщо врахувати, що напрям вектора дотичною задає початковий напрям, а модуль вектору дотичної задає міру витягнутості кривої у напрямку цього вектора, як показано на рис. 4.2.

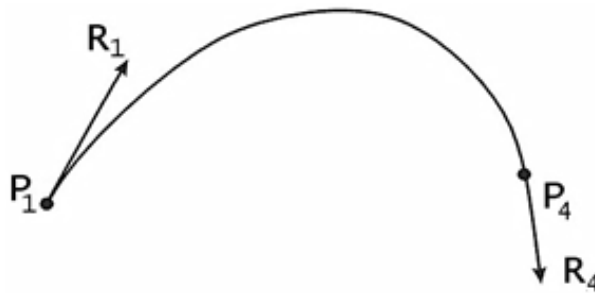


Рис. 4.2. Параметричний сплайн у формі Ерміта

Витягнутість кривої вправо забезпечується тим, що $|R_1| > |R_4|$.

Розглянемо форму Без'є, яка відрізняється від форми Ерміта способом задавання граничних умов, а саме замість векторів R_1 і R_4 вводяться точки (і відповідні їм радіус-вектори) P_2 і P_3 , як показано на рис. 4.3, такі, що задовольняють умови: $P'(0) = R_1 = 3(P_2 - P_1)$ і $P'(1) = R_4 = 3(P_4 - P_3)$.

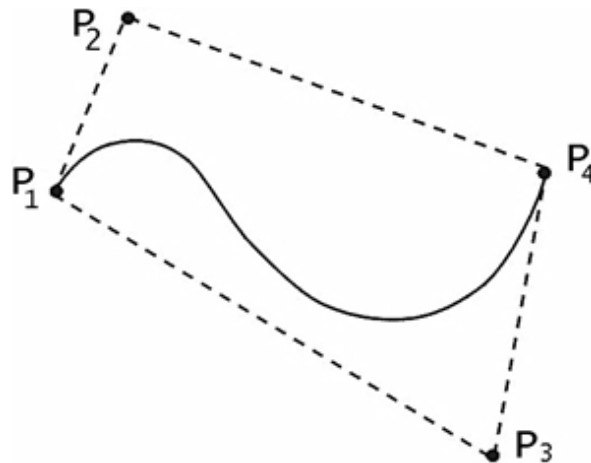


Рис. 4.3. Параметричний сплайн у формі Без'є

Перехід від форми Ерміта до форми Без'є виконується шляхом перетворення (4.12):

$$G_h = \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix} \cdot \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} = M_{hb} \cdot G_b, \quad (4.12)$$

де G_b – геометричний вектор Без'є. Підставляючи отримані дані у вираз для $x(t)$, отримуємо (4.13):

$$\begin{aligned} x(t) &= T \cdot M_h \cdot G_{hx} = \\ &= T \cdot M_h \cdot M_{hb} \cdot G_{bx} = \\ &= P_1(1 - t^3) + P_2 3t(t - 1)^2 + \\ &+ P_3 3t^2(1 - t) + P_4 t^3. \end{aligned} \quad (4.13)$$

Корисною властивістю сплайнів у формі Без'є є те, що крива завжди лежить усередині опуклої оболонки, утвореної чотирикутником $(P_1 P_2 P_3 P_4)$. Цю властивість можна довести, користуючись тим, що у вираз (4.12) коефіцієнти набувають значень від 0 до 1 і їхня сума дорівнює одиниці.

Слід зауважити, що матриця вигляду (4.14) – називається матрицею Без'є.

$$M_h \cdot M_{hb} = M_b = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \quad (4.14)$$

Дещо складніший тип інтерполяції – так звана поліноміальна сплайн-інтерполяція, або інтерполяції В-сплайнами. На відміну від звичайної сплайн-інтерполяції, зшивання елементарних В-сплайнів проводиться не у точках (t_i, x_i) , а у інших точках, координати яких зазвичай пропонується визначити користувачеві. Таким чином, вимога

рівномірного дотримання вузлів під час інтерполяції В-сплайнами відсутній і з їх допомогою можна наближати розрізнені дані.

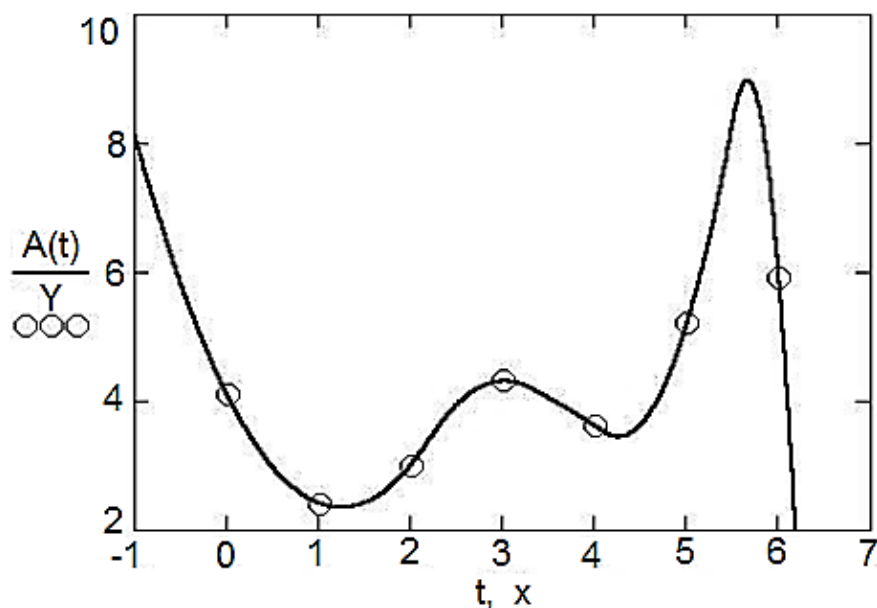


Рис. 4.4. Інтерполяція В-сплайнами

Сплайни можуть бути поліномами першого, другого або третього ступеня (лінійні, квадратичні або кубічні). Застосовується інтерполяція В-сплайнами точно так само, як і звичайна сплайн-інтерполяція, відмінність полягає тільки у визначенні допоміжної функції коефіцієнтів сплайна.

4.3. NURBS

Розглянуті методи досить просто узагальнюються на випадок апроксимації двовимірних функцій. Для конструювання криволінійних поверхонь за допомогою стандартних параметричних поліномів, поліномів Бернштейна і NURBS у сучасних системах геометричного моделювання застосовують три основні методи:

- тензорного добутку (tensor product surfaces);
- каркасний (lofting surfaces);
- булевої суми (transfinite method).

Водночас раціональні параметричні криві Безье є частковим випадком NURBS.

Узагальнення методів Без'є і В-сплайнів на початку 70-х років дозволило отримати один із найпотужніших і універсальних засобів

геометричного моделювання криволінійних обводів – NURBS-технологію (Non-Uniform, Rational B-Spline) – нерівномірний раціональний B-сплайн. Базова геометрична фігура (геометричний примітив) використовується для опису кривих поверхонь.

На сьогодні існує безліч способів опису геометричних об'єктів. Проте з точки зору співвідношення можливостей і витрат обчислювальних ресурсів найпотужнішим інструментом є так зване NURBS-моделювання. В основі методу NURBS лежить опис об'єктів за допомогою так званих раціональних поліноміальних функцій. Цей підхід дає максимальну економію обчислювальних ресурсів і абсолютну гнучкість у створенні об'єктів довільної форми.

Моделі NURBS створюються з декількох кривих. При цьому для настройки викривлення досить змінити власну вагу вершин і визначити сплайни. Поверхні, побудовані на основі NURBS-кривих, називають NURBS-об'єктами. Вони виглядають плавними і згладженими, легко піддаються редагуванню. Прорахунок таких моделей є більш прискореним, ніж у традиційних полігональних мереж.

4.4. Моделювання сплайна у 3ds max

Щоб краще зрозуміти, що таке сплайни, відкриємо меню Create і знайдемо іконку Shapes. У випадному меню виберемо Spline (рис. 4.5).

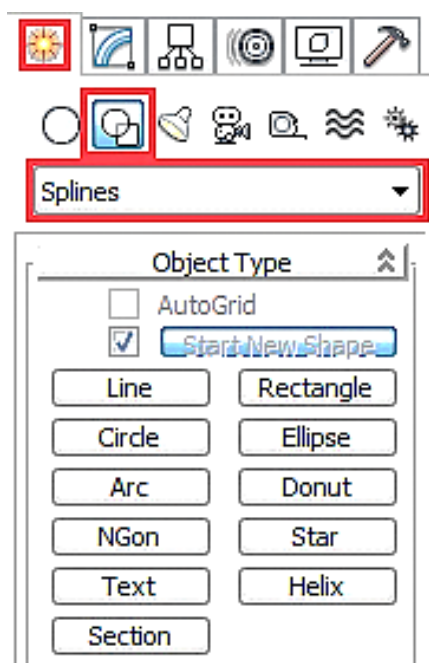


Рис. 4.5. Вкладка примітивів сплайнів

З цього меню починається будь-яка робота зі сплайнами (spline) в 3ds max. Вкладка Object Type містить 11 примітивів (рис. 4.6) сплайнів.

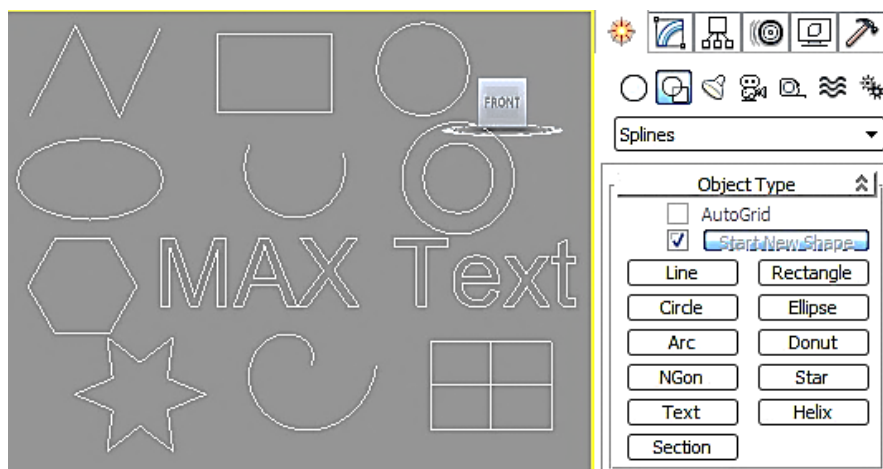


Рис. 4.6. Зовнішній вигляд примітивів сплайнів

Кожна з форм має ряд своїх налаштувань: наприклад, прямокутнику можна задати довжину сторін, зірці – кількість і глибину променів. Line є найгнучкішим інструментом, оскільки дозволяє побудувати абсолютно будь-який контур.

Виберемо примітив сплайна Create/Shapes/Line. Створимо з його допомогою довільну форму на проекції Top (рис. 4.7).

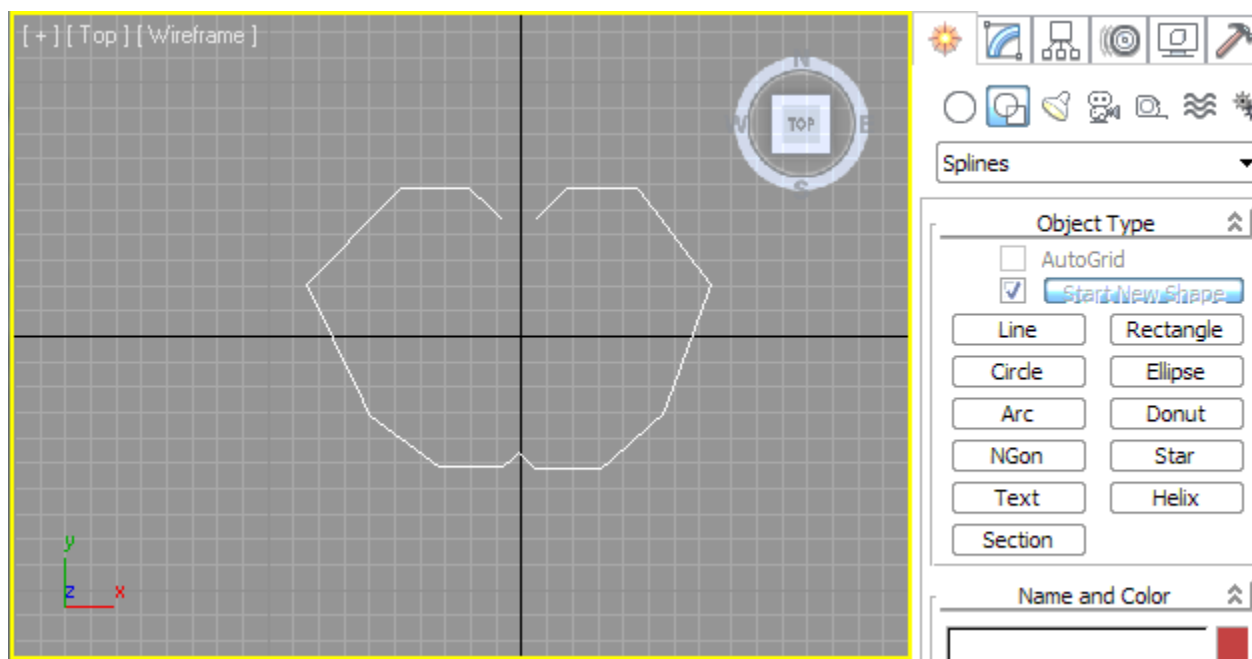


Рис. 4.7. Формування сплайна з окремих ліній

Після завершення малювання форми треба натиснути праву кнопку мишки.

Як і під час роботи з полігонами, робота зі сплайнами має декілька режимів редагування подоб'єктів. Їх можна побачити в свитці Selection або натиснувши на плюстик біля напису Line (рис. 4.8).

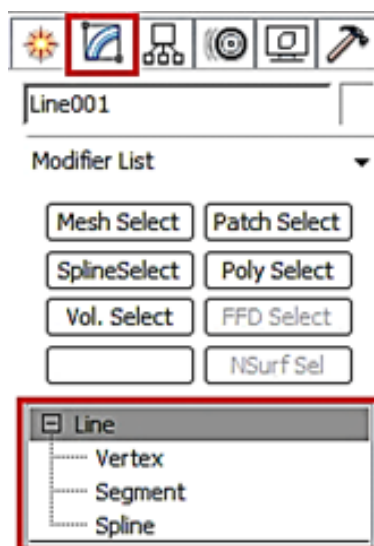


Рис. 4.8. Режими редагування подоб'єктів сплайна

З рисунка видно три режими редагування:

Перший – це Vertex, тобто редагування по точках, вершинах. Вертексами називають вузли, на скриншоті їх добре видно, червона точка – виділений (активний) вертекс (рис. 4.9).

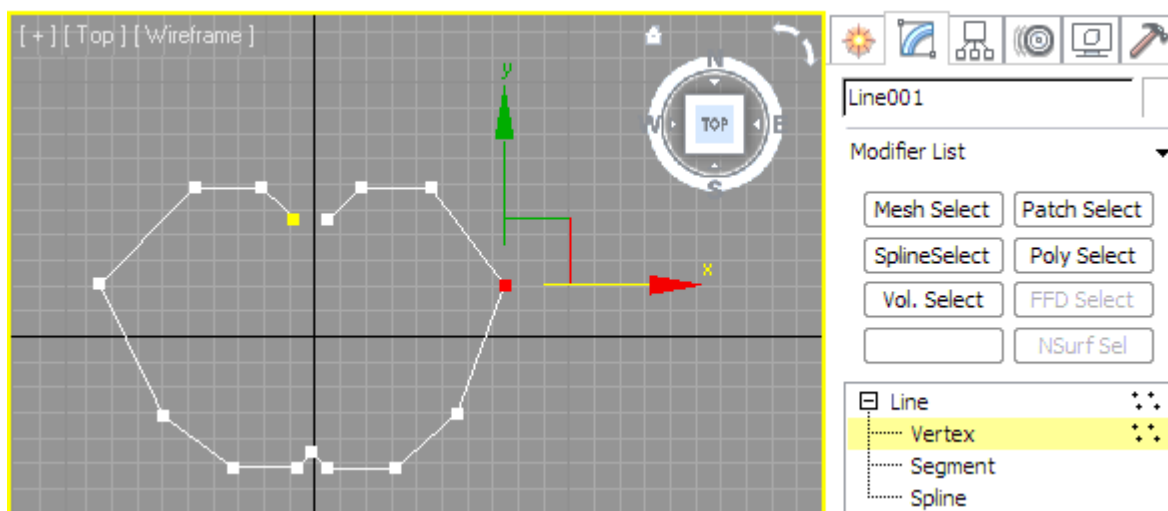


Рис. 4.9. Виділення і редагування точки сплайна

Другий – Segment – редагування у сегментах. Сегмент – відрізок, обмежений двома вертексами (рис. 4.10).

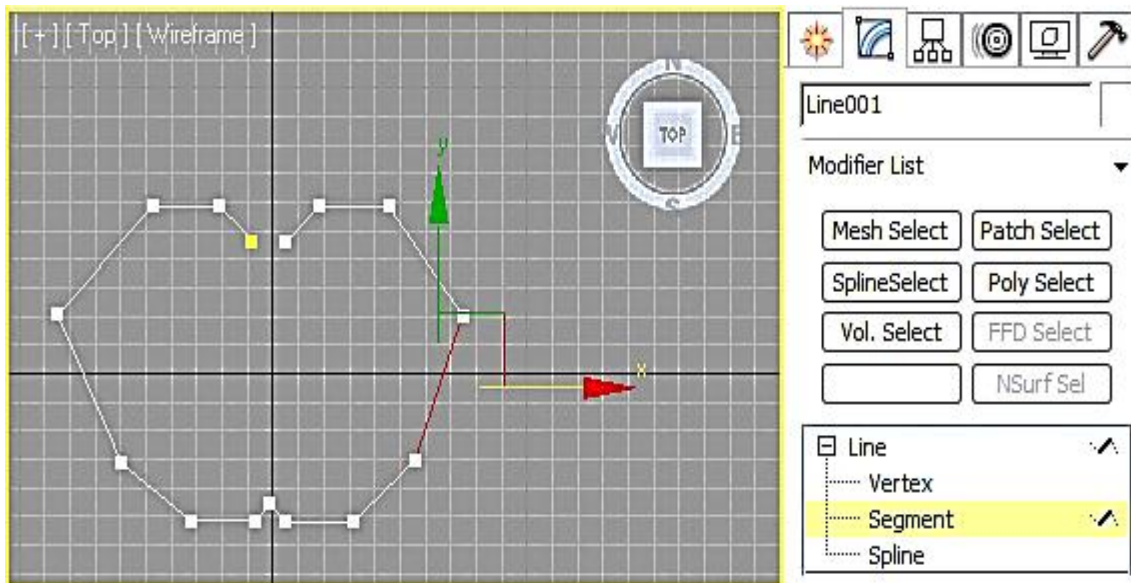


Рис. 4.10. Виділення сегменту

Третій – Spline – редагування по всьому сплайну. Сплайн виділяється повністю: від першої до останньої точки (рис. 4.11).

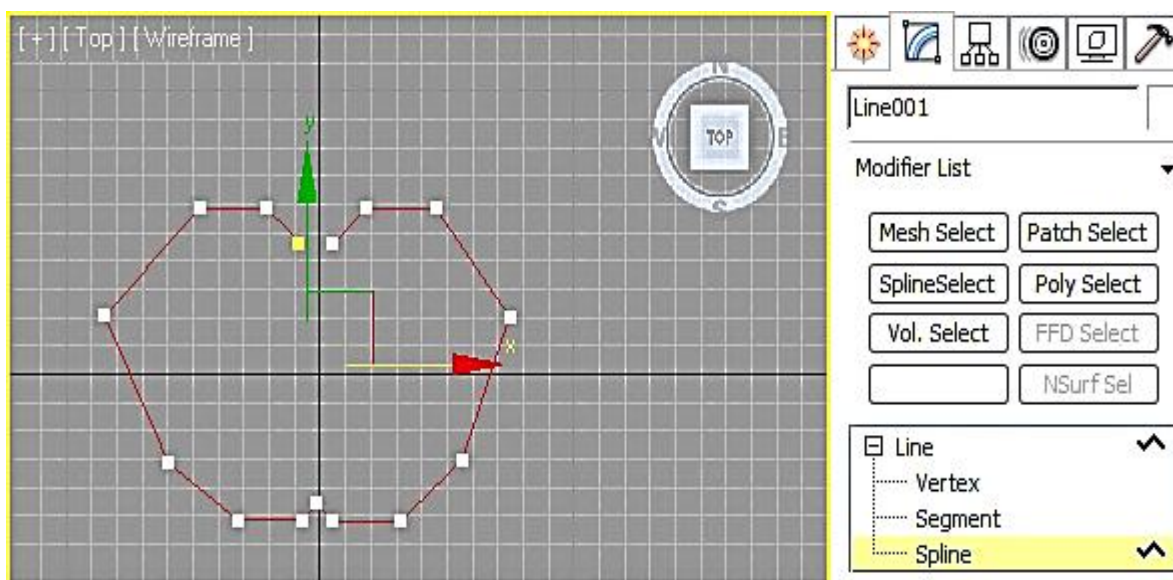


Рис. 4.11. Виділення всіх точок сплайна

Виділимо точку і викличемо контекстне меню. У ньому є блок, у якому позначений тип виділеної точки (рис. 4.12).

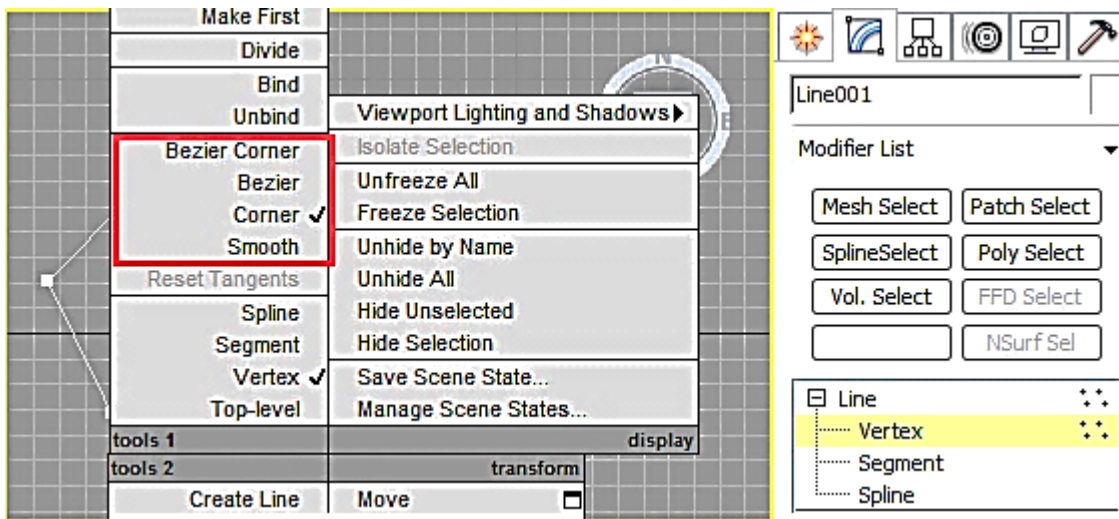


Рис. 4.12. Типи точок сплайнів

Виділяють такі точки сплайнів:

Bezier Corner – кут із кривою. У процесі застосування виходить кут, який матиме два не залежних один від одного маніпулятори;

Bezier – криву із симетричними маркерами-маніпуляторами, що згинаються разом;

Corner – кут, ламану лінія;

Smooth – автоматичне згладжування, що не має маніпуляторів.

Виділимо всі точки і виберемо тип точки Smooth (рис. 4.13).

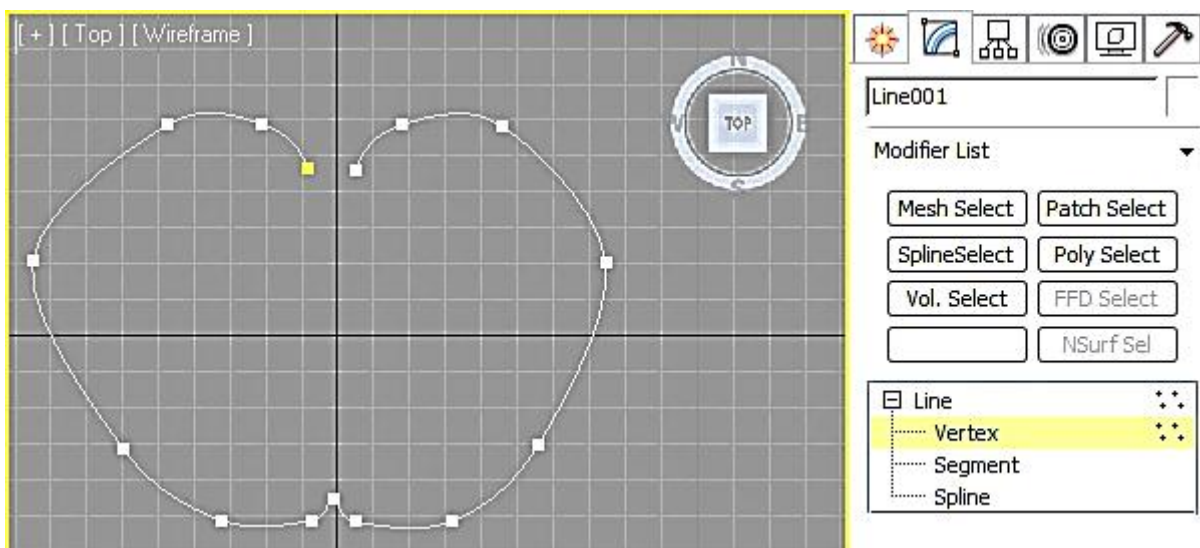


Рис. 4.13. Згладжування кривої сплайна

Вуглувата форма кривої стала згладженою.

Далі можна виділити декілька точок і перемістити їх. Форма кривої помітно зміниться (рис. 4.14).

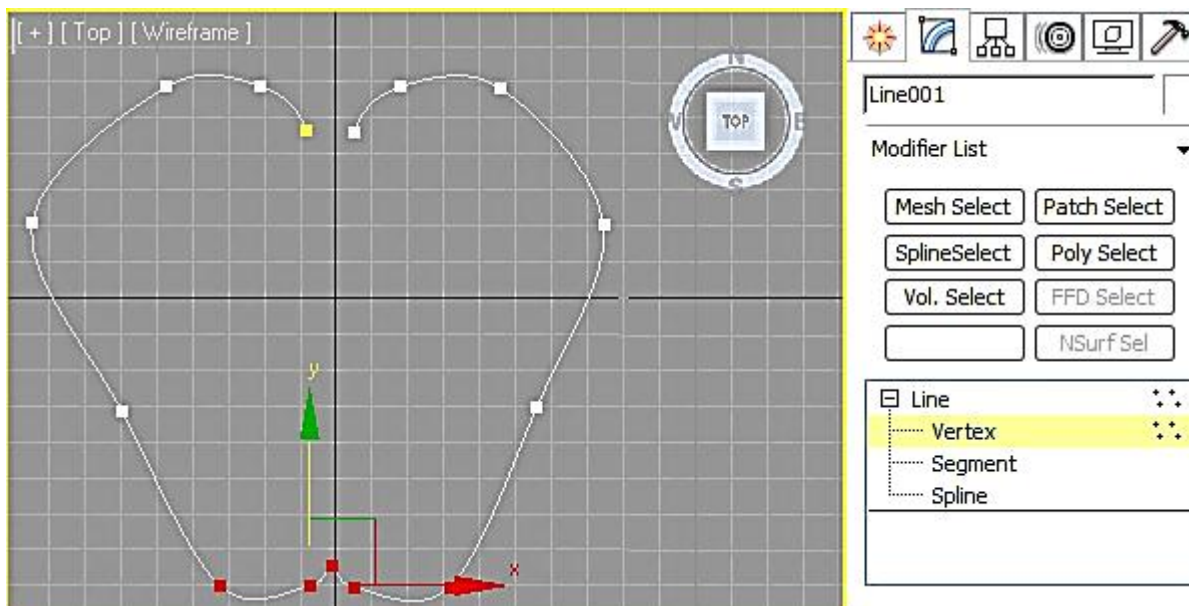


Рис. 4.14. Виділення декількох точок на сплайні

Для того щоб додати точку на сплайні, треба на вкладці Geometry натиснути кнопку Refine, а потім розмістити точки у потрібному місці на лінії сплайна (рис. 4.15).

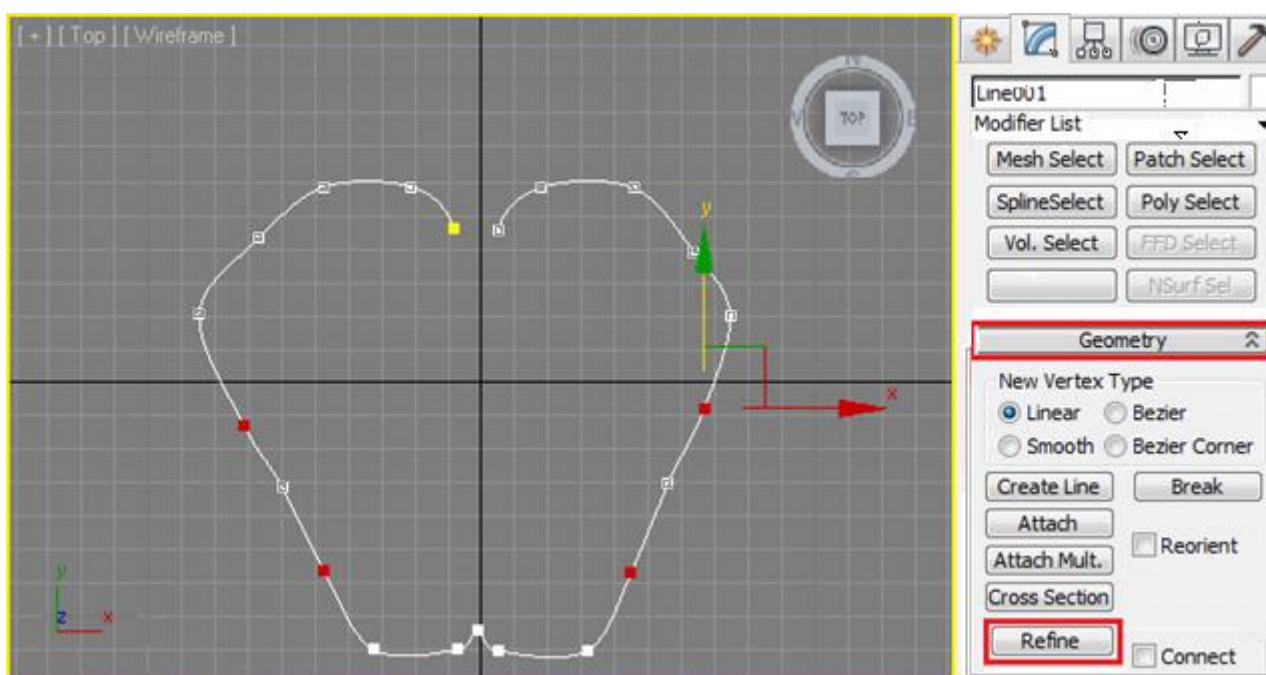


Рис. 4.15. Додавання точок на сплайні

Для видалення точки треба її виділити і натиснути Del на клавіатурі.

Для того щоб з'єднати дві точки на сплайні, треба виділити точки і вкладці Geometry, натиснути кнопку Fuse (об'єднати), а потім натиснути кнопку Weld (рис. 4.16).

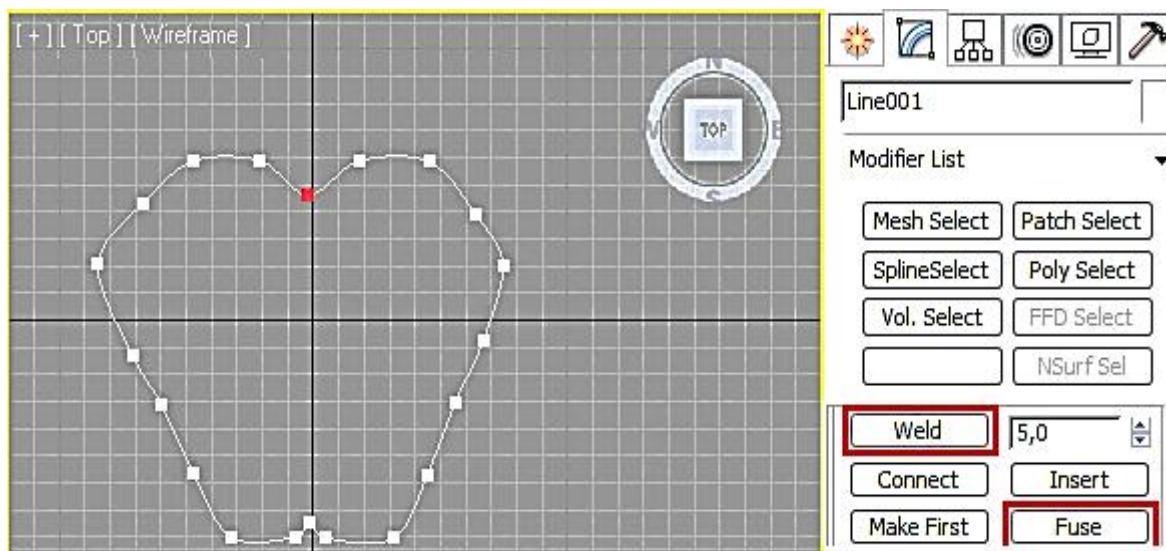


Рис. 4.16. З'єднання двох точок сплайна

У цьому місці можна поставити тип точки Corner Bezier.

Якість згладжування форми можна поліпшити, збільшивши кількість Steps (кроків) в світці Interpolation (рис. 4.17).

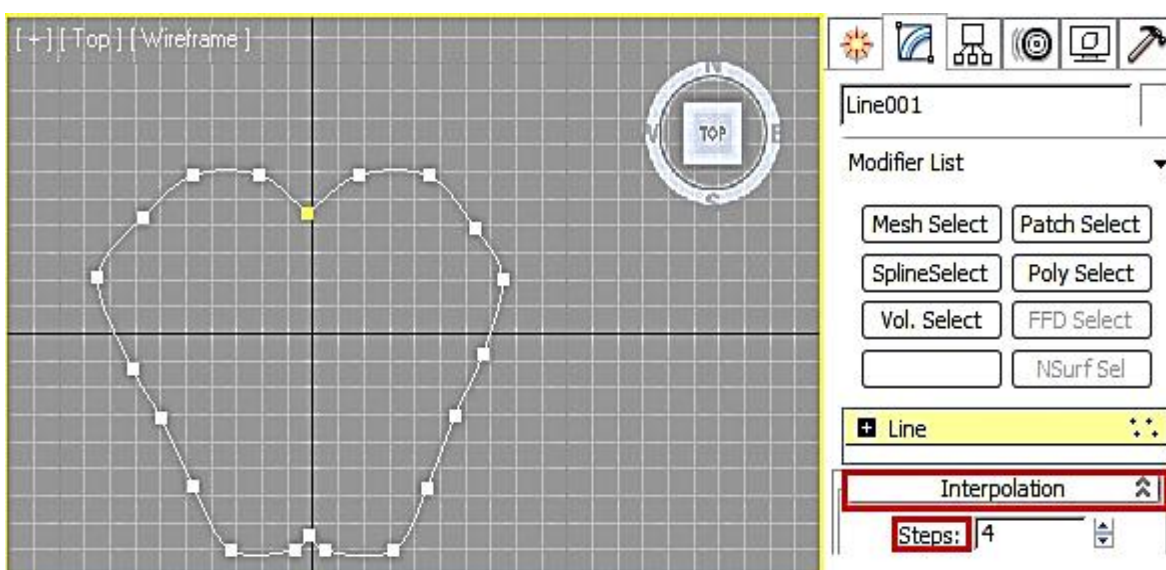


Рис. 4.17. Зміна параметра Steps для згладжування сплайна

Часто виникає потреба в об'єднанні сплайнів між собою. Для того щоб продемонструвати це, створимо яблуку листочок (рис. 4.18).

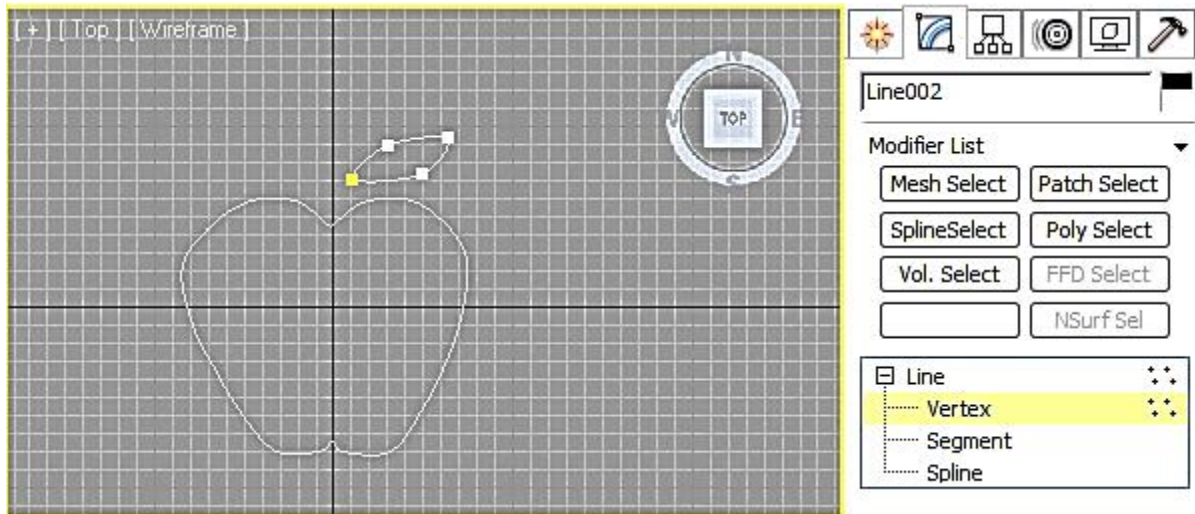


Рис. 4.18. Формування другого сплайна сцени

При замиканні першої точки з останньої, в процесі створення листочка, на питання «Close spline?» потрібно відповідати ствердно. Щоб з'єднати листок і яблуко, вибираємо одну з фігур, натискаємо кнопку Attach і вибираємо те, що хочемо приєднати. Після цього, в процесі переходу у режим подоб'єктів добре видно, що точки можна редагувати і на листку, і на самому яблуці (рис. 4.19).

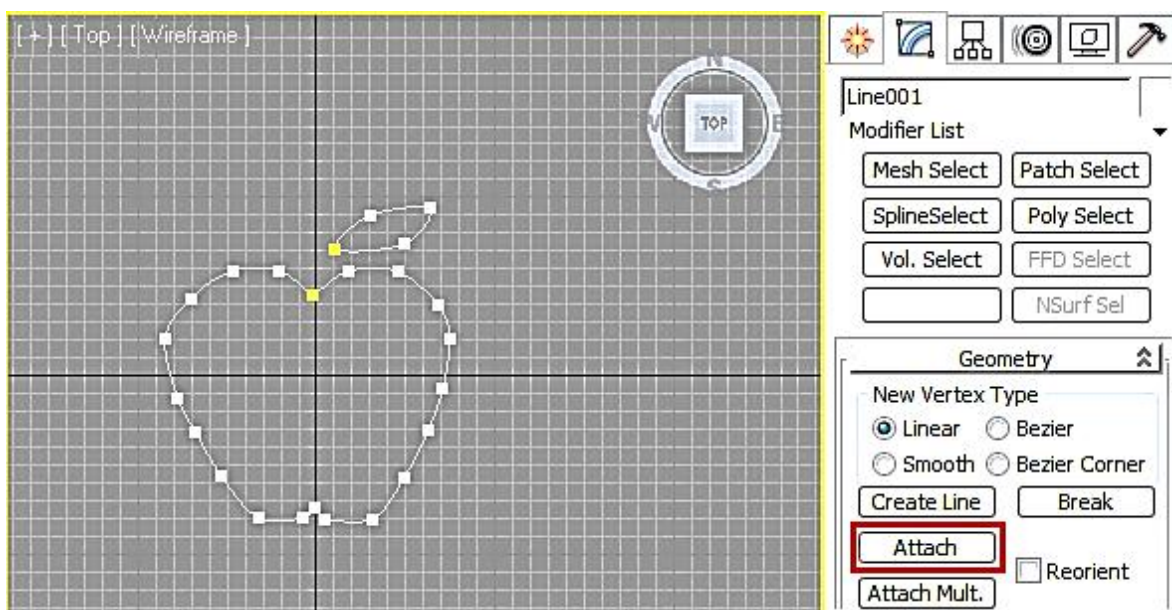


Рис. 4.19. Об'єднання двох сплайнів

Для завершення частково намальованого сплайна (рис. 4.20 а) використовується кнопка Insert на вкладці Geometry (рис. 4.20 б). Далі приєднаємо гілочку за допомогою кнопки Attach (рис. 4.20 в).

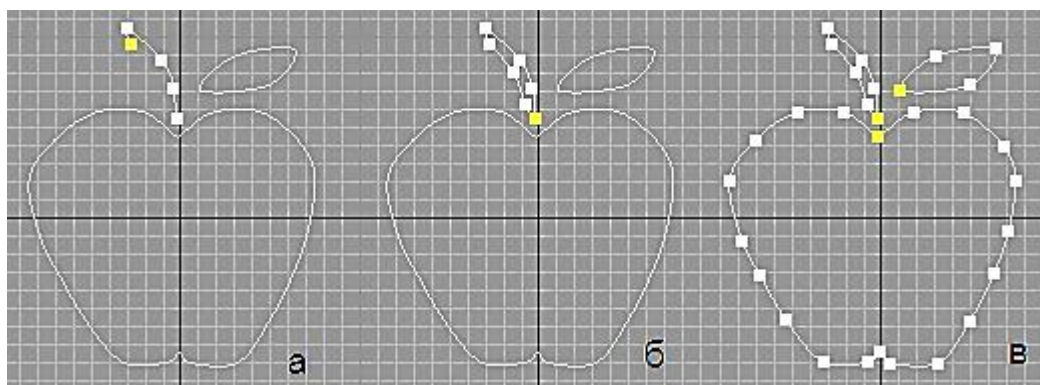


Рис. 4.20. Завершення частково намальованого сплайна

Для надання товщини лінії сплайна досить поставити два прапорці Enable in Renderer і Enable in Viewport у вкладці Rendering (рис. 4.21).

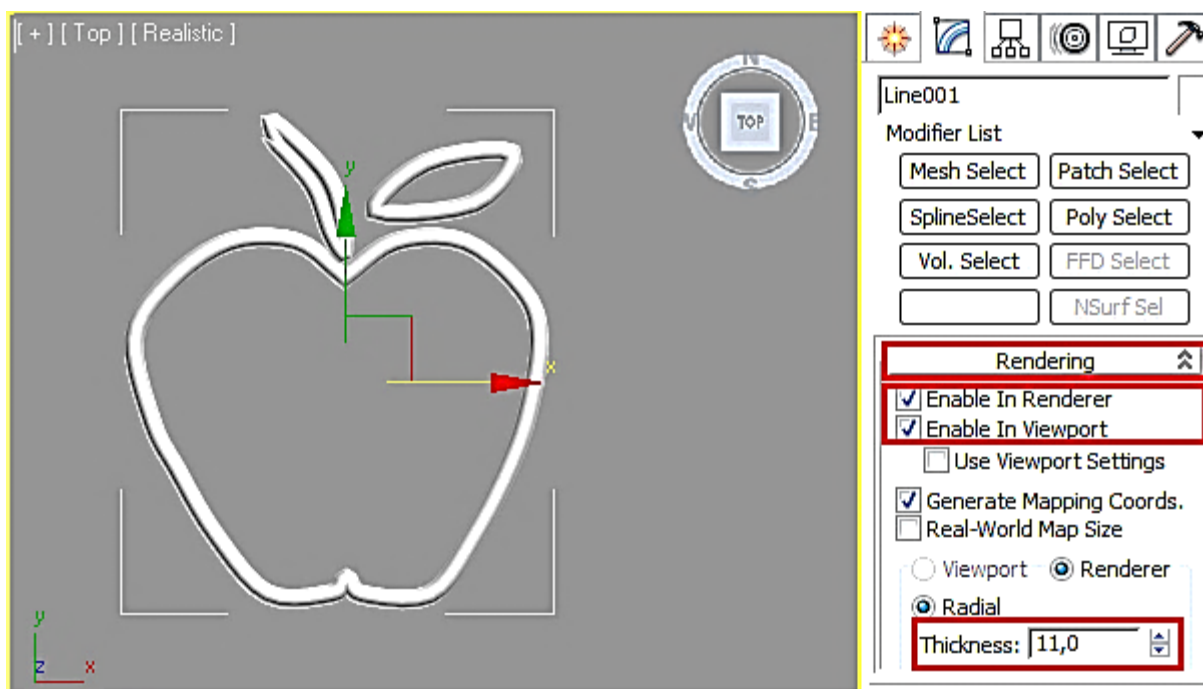


Рис. 4.21. Надання товщини лінії сплайна

4.5. Приклад моделювання штампа

Спочатку необхідно налаштувати одиниці вимірювання у сцені. Для цього заходимо у меню Customize > Units Setup (рис. 4.22).

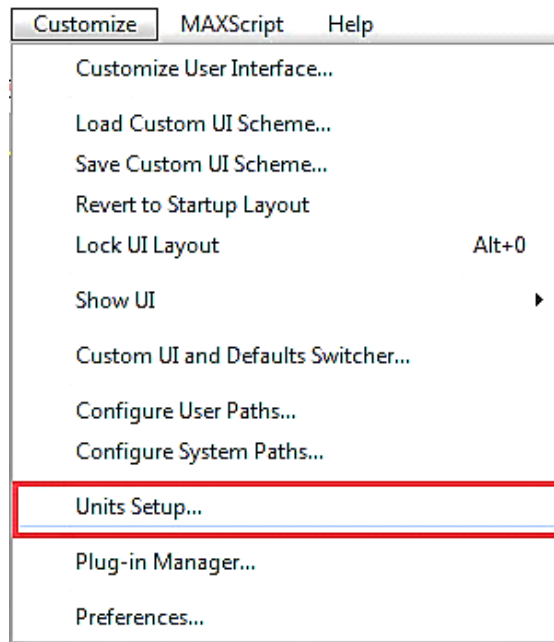


Рис. 4.22. Налаштування одиниці вимірювання у сцені

У вікні, що з'явилося, виставляємо метричну систему вимірювання (Metric) і вибираємо одиницю вимірювання – міліметри (Millimeters).

Натискаємо " System Unit Setup" і у вікні, що відкрилося, виставляємо також міліметри і значення 1,0 (рис. 4.23).

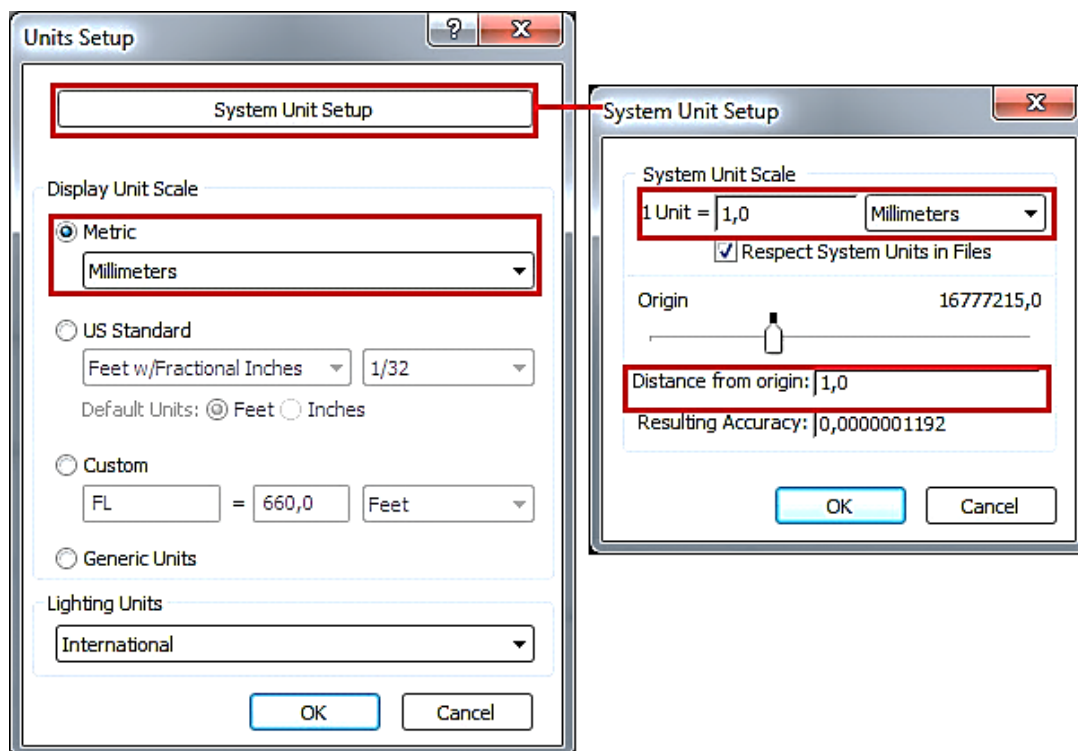


Рис. 4.23. Вибір одиниці вимірювання

Тепер сцена має натуральні звичні розміри.
Штамп повинен мати вигляд (рис. 4.24).



Рис. 4.24. Зовнішній вигляд штампа

Скористаємося малюнком штампа як підкладкою для подальшого обведення і витискування.

Створимо геометричний примітив Plane розміром 50×50 мм. Кількість сегментів по довжині (Length Segs) і ширині (Width Segs) встановимо у 1 (рис. 4.25).

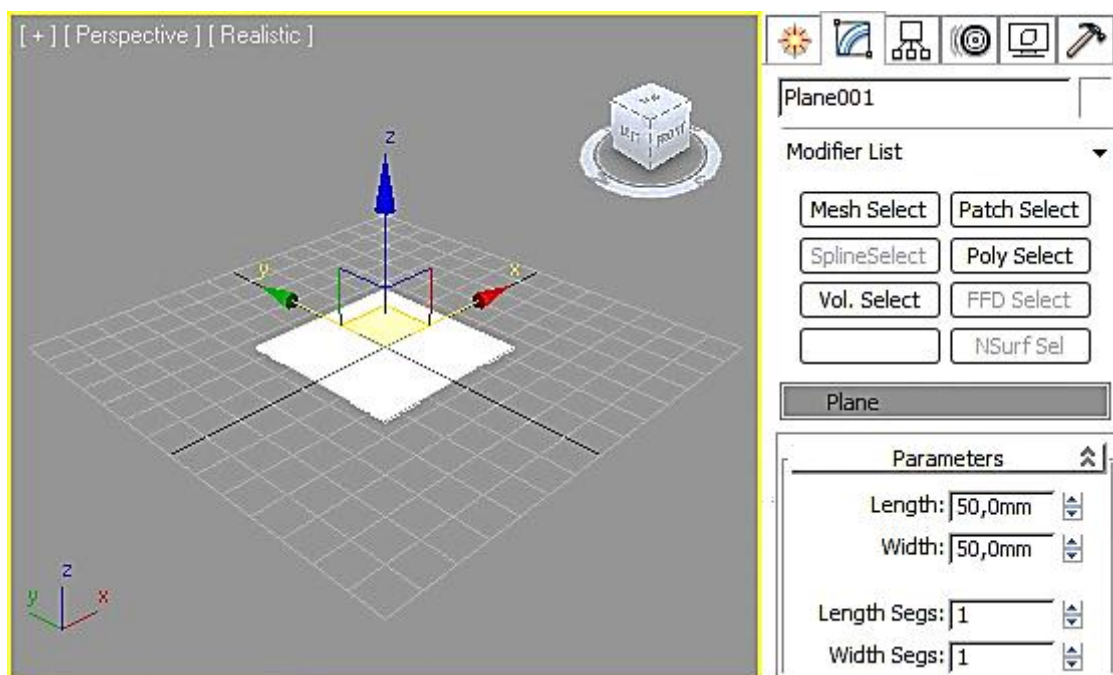


Рис. 4.25. Створення геометричного примітиву Plane

Виконаємо текстурування площини малюнком штампа шляхом перетягування файл з картинкою у вікно перспективи. Далі скористаємося модифікатором UVW Map і вкладці Parameters, що з'явилася, зніmemo галочку з пункту Real – world Map Scale (рис. 4.26).

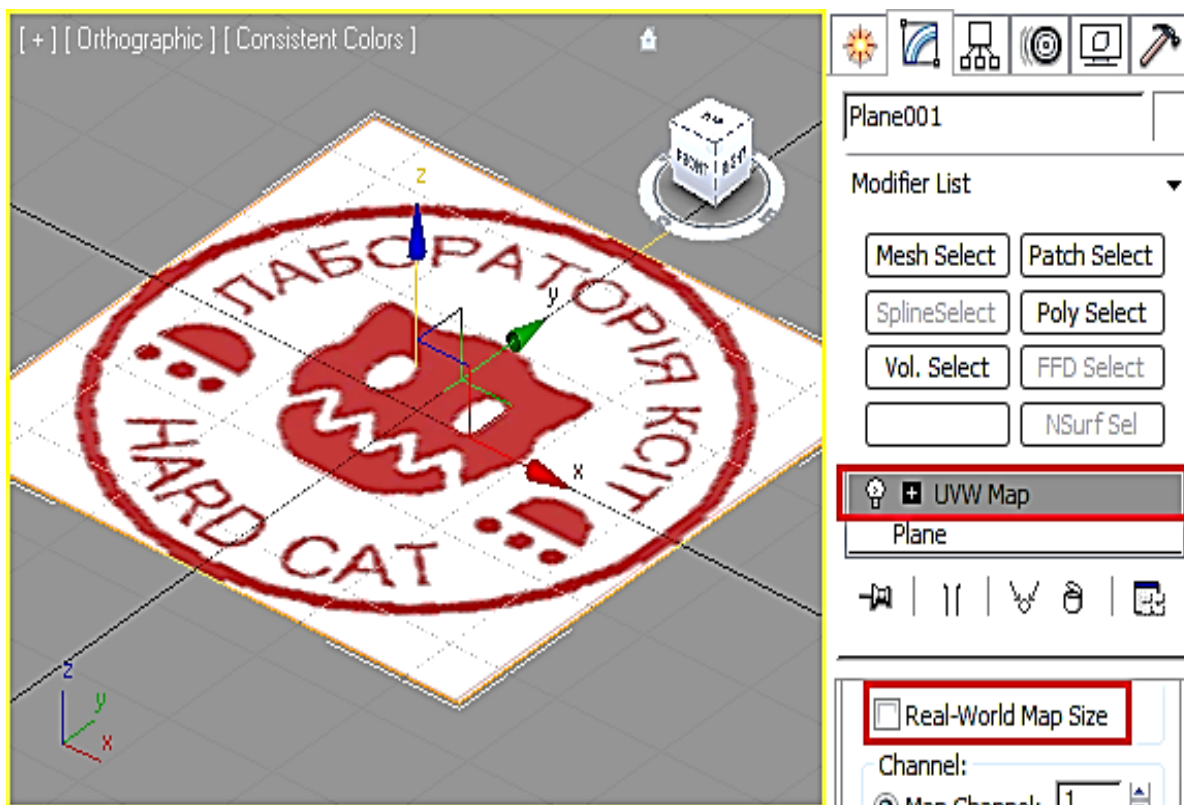


Рис. 4.26. **Текстурування площини малюнком штампа**

Для зручності й готовності підкладки відцентруємо площину ($x = 0$, $y = 0$, $z = 0$). Обведення підкладки найзручніше робити у вікні Top.

Для того щоб побачити малюнок штампа у вікні Top треба встановити режим відображення в'ю порту Shaded (F3).

Використовуємо різні примітиви сплайнів для обведення малюнка штампа. У меню створення об'єктів вибираємо Shapes > Splines і вибираємо Line (Лінію), потім обводимо голову кота. Для обведення очей скористаємося примітивом сплайна еліпс. З'єднаємо всі чотири сплайни у один командою Attach і для більшої наочності використаємо клавішу (F3).

Для обведення лапок використовують примітиви сплайнів еліпс, дугу і лінію. Після цього об'єднуємо сплайн голови зі сплайнами лапок (рис. 4.27).

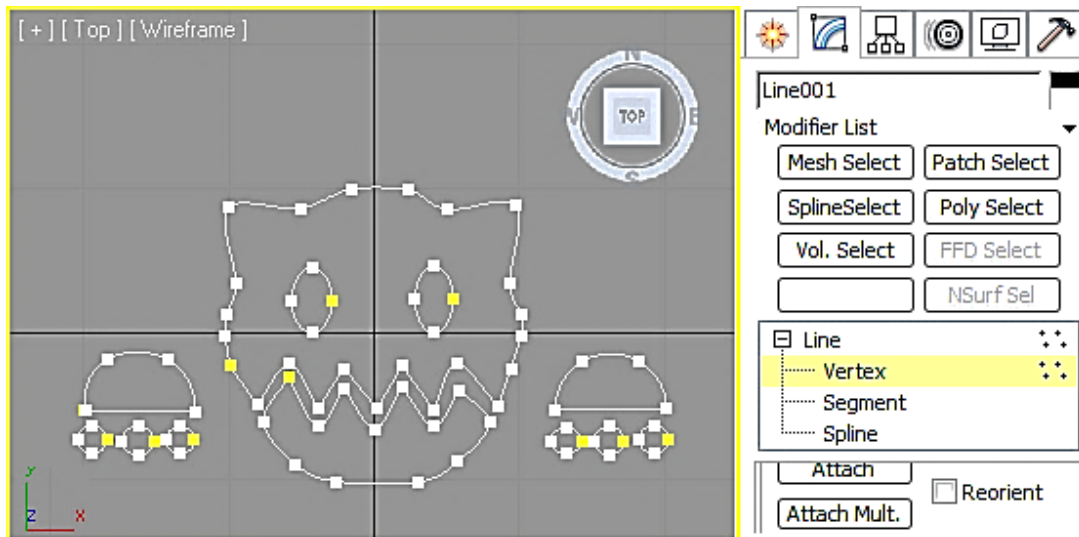


Рис. 4.27. Обведення ліній малюнка штампа

Тепер можна нанести текст. Вибираємо Text у списку примітивів сплайнів і ставимо його у центрі штампа. За замовчуванням з'явиться великий напис "Max Text". Щоб змінити текст, розмір, шрифт та інтервали, заходимо у вкладку Редагування об'єкта і міняємо відповідні параметри:

Потім текст необхідно зігнути у формі кола. Для цього скористаємося модифікатором Bend (рис. 4.28). Залежно від напрямку вигину тексту параметр налаштування Angle модифікатора Bend буде з мінусом або плюсом. Далі приєднуємо текст до основного сплайна.

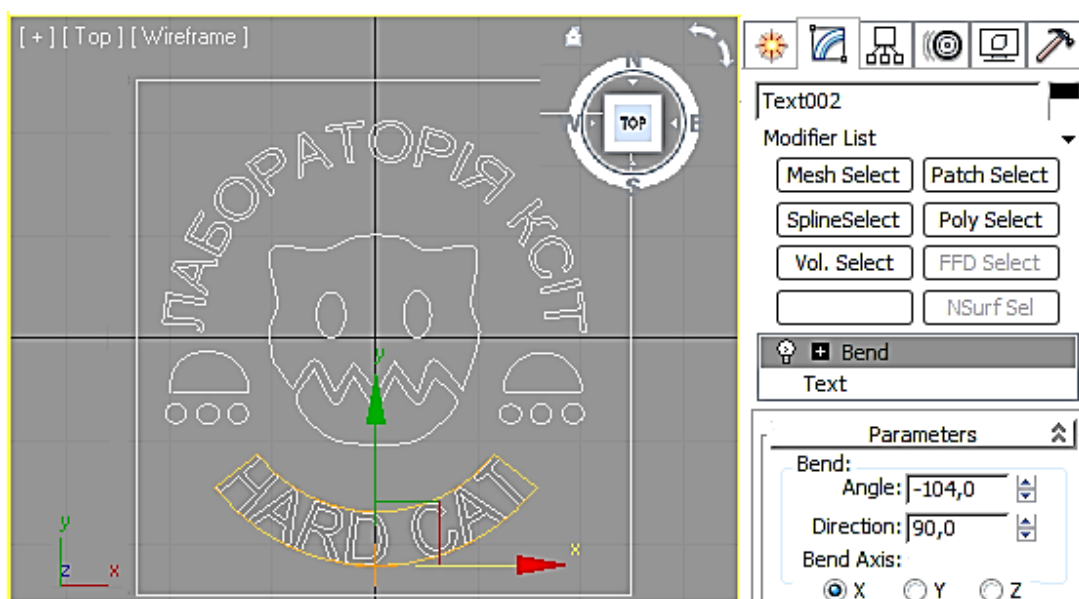


Рис. 4.28. Вигин напису на малюнку штампа

Сформуємо на малюнку штампа обідок з двох кіл і приєднаємо їх до основного сплайна (рис. 4.29).

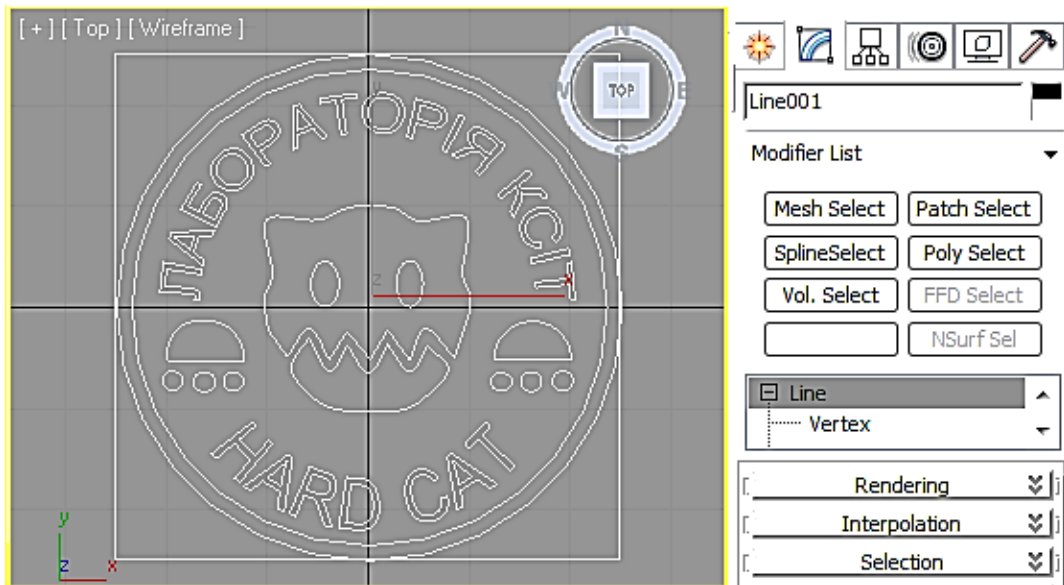


Рис. 4.29. Формування обідка штампа

Після того як всі сплайни об'єднані, застосуємо до малюнка штампа модифікатор Extrude, у налаштуваннях зазначаємо висоту витискання. Для друку можна узяти значення від 2 до 3 мм. Перевіряємо, щоб стояли галочки Cap Start і Cap End. Якщо усе зроблено правильно, то побачимо, що малюнок штампа став об'ємним (рис. 4.30).



Рис. 4.30. Застосування до малюнка штампа модифікатора Extrude

Після створення малюнка штампа можна прибрати об'єкт Plane з малюнком через непотрібність.

Тепер змоделюємо рукоятку. Інструментом Line у вікні проєкції

Front або Left малюємо половинку руків'я. Починати малювати можна з будь-якої точки. Після завершення малювання половинки контуру рукоятки перетворимо червоні точки в Smooth, що залишилися три точки залишимо Bezier (рис. 4.31).

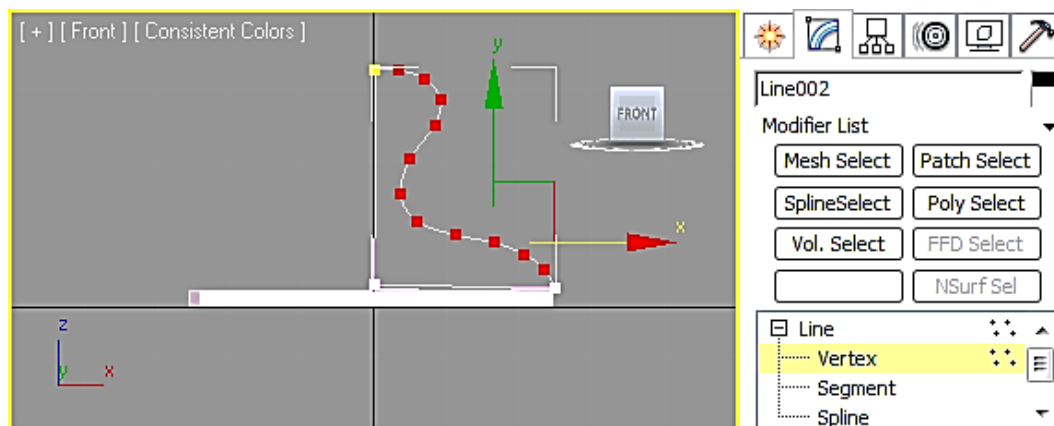


Рис. 4.31. Формування профілю рукоятки штамп

Тепер до сплайна-половинки, застосовуємо модифікатор Lathe. Для налаштування використовують елементи управління:

Segments – змінити кількість сегментів;

Direction - вибрати вісь, по якій буде відбуватиметься закручування;

Align - параметр вирівнювання.

Якщо усе зроблено правильно, то побачимо, як зі сплайна вийшло тіло обертання (рис. 4.32).

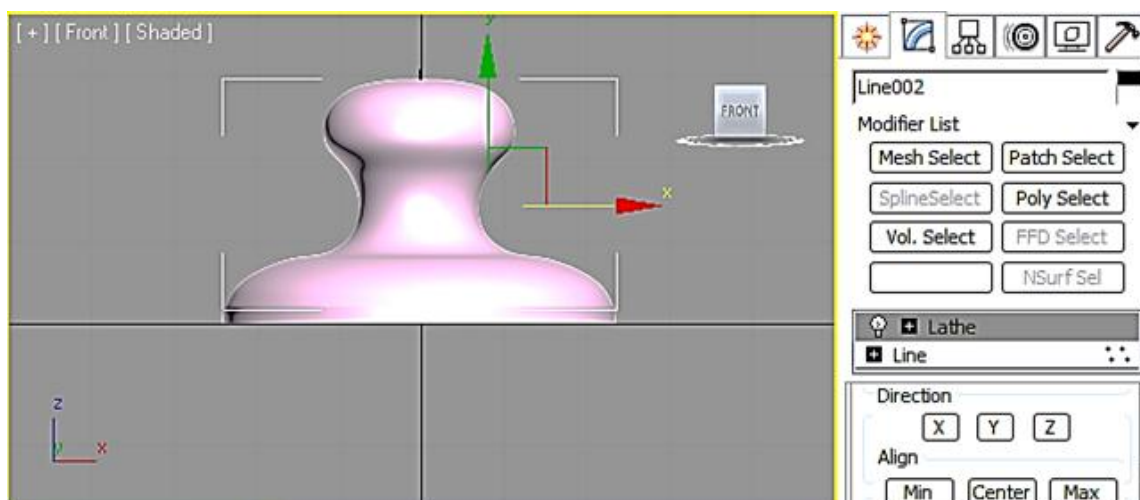


Рис. 4.32. Застосування модифікатора Lathe до профілю рукоятки

Тепер необхідно об'єднати витиснуті малюнок штампа і рукоятку. Для цього конвертуємо руків'я і малюнок штампа у Editable Poly і групуємо їх (рис. 4.33).

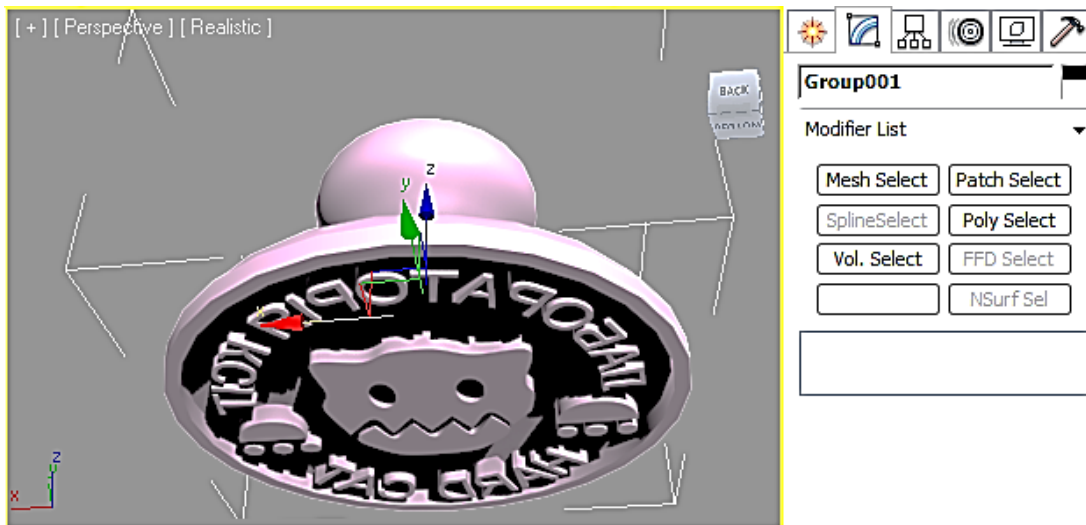


Рис. 4.33. Об'єднання малюнка штампа з руків'ям

Лінію сплайна можна зробити тривимірною без витискування. Для цього у налаштуваннях об'єкта потрібно поставити галочки у пунктах Enable In Renderer і Enable in Viewport. Потім задаємо профіль Radial (круглий) або Rectangular (прямокутний) і виставляємо необхідні розміри. Таким чином, обвівши різні фігури, можна виготовити форми.

Контрольні запитання

1. Сплайнова інтерполяція.
2. Види сплайнів.
3. Сплайнова функція.
4. Поліноміальна сплайн-інтерполяція.
5. Конструювання криволінійних поверхонь.
6. Сплайнові примітиви .
7. Додаткові сплайні об'єкти.
8. Створення складних геометричних тривимірних об'єктів на основі сплайнівих фігур.
9. Модифікатори сплайнів.
10. Створення тривимірних об'єктів на основі сплайнів
11. Основні прийом роботи з NURBS кривими і поверхнями.
12. Режим Sub-Object.

Література

Основна

1. Верстак В. А. Видео самоучитель. 3ds max (+DVD). / В. А. Верстак – Санкт-Петербург : Питер, 2008. – 336 с.
2. Голованов Н. Н. Геометричне моделювання. / Н. Н. Голованов – Москва : Изд-во физико-математической литературы, 2002. – 472 с.
3. Макаров Е. Г. Mathcad : учебный курс. / Е. Г. Макаров – Санкт-Петербург : Питер, 2009. – 384 с.
4. Маценко В. Г. Комп'ютерна графіка : навч. посібник. / В. Г. Маценко – Чернівці : Рута, 2009. – 343 с.
5. Морозов А. Д. Введение в теорию фракталов. / А. Д. Морозов – Москва – Ижевск : Институт компьютерных исследований, 2002. – 160 с.

Додаткова

6. Демин А. Ю. Основы компьютерной графики: учеб. пособ. / А. Ю. Демин – Томский политехнический университет. – Томск : Изд-во Томского политехнического университета, 2011. – 191 с.
7. Иванов В. П. Трехмерная компьютерная графика / В. П. Иванов, А.С. Батраков / Под ред. Г.М. Полищука. – Москва : Радио и связь, 1995. – 224 с.
8. Мандельброт Б. Б. Фракталы и хаос. Множество Мандельброта и другие чудеса. / Б. Б. Мандельброт Б. Б. – Москва : Ижевск : НИЦ «Регулярная и хаотическая динамика», 2009. – 392 с.
9. Методічні рекомендації до самостійної роботи «Теорія цифрових зображень» для студентів галузі знань 0515 «Видавничо-поліграфічна справа» всіх форм навчання / уклад. Гаврилов В. П. – Харків : Вид. ХНЕУ, 2012. – 93 с.
10. Никулин У. А. Компьютерная геометрия и алгоритмы машинной графики. / У. А. Никулин – Санкт-Петербург : БХВ-Петербург, 2003. – 560 с.
11. Рябцев Д. В. Дизайн помещений и интерьеров в 3ds max 2009 (+DVD). / Д. В. Рябцев – Санкт-Петербург : Питер, 2009. – 512 с.
12. Стиренко А. С. 3ds Max 2009/3ds Max Design 2009. Самоучитель. / А. С. Стиренко – Москва : ДМК Пресс, 2008. – 544 с.
13. Шредер М. Фракталы, хаос, степенные законы. Миниатюры из бесконечного рая. / М. Шредер – Ижевск : НИЦ «Регулярная и

хаотическая динамика», 2001. – 528 с.

Интернет-ресурсы

14. Школа дизайна SKIDEL-SKY [Электронный ресурс] – Режим доступа : www.skidel-sky.ru.

15. Форум Фракталус [Электронный ресурс] – Режим доступа : <https://fractalus.ru>.

16. Уроки з 3ds max [Электронный ресурс] – Режим доступа : www.3dmir.ru/s_tutor/tutorial/1.html.

17. Incendia EX. Урок. Хризантема.avi, Видео, Смотреть онлайн. [Электронный ресурс] – Режим доступа : http://salda.ws/video.php?id=_H_LSJ7U3mY.

18. Incendia EX V. Руководство. Часть первая. [Электронный ресурс] – Режим доступа : <https://fractalus.ru/lessons/item/124-incendia-ex-v-rukovodstvo-chast-pervaya.html>.

19. Incendia EX V. Руководство. Часть вторая. [Электронный ресурс] – Режим доступа : <https://fractalus.ru/lessons/item/125-incendia-ex-v-rukovodstvo-chast-vtoraya.html>.

Зміст

| | |
|--|----|
| Вступ | 3 |
| Розділ 1. Аналітичні 3D-об'єкти | 4 |
| 1. Аналітична 3D-графіка | 4 |
| 1.1. Способи побудови поверхонь | 4 |
| 1.2. 3D-графіка | 5 |
| 1.3. Математична модель поверхонь | 7 |
| 1.4. Аналітичні поверхні | 9 |
| 1.4.1. Площина | 10 |
| 1.4.2. Сфера | 11 |
| 1.4.3. Еліпсоїд | 13 |
| 1.4.4. Однопорожнинний гіперболоїд | 14 |
| 1.4.5. Двопорожнинний гіперболоїд | 15 |
| 1.4.6. Еліптичний параболоїд | 16 |
| 1.4.7. Гіперболічний параболоїд | 17 |
| 1.4.8. Циліндр | 18 |
| 1.4.9. Тороїдальна поверхня | 20 |
| 1.5. Приклади 3D-графіків у Mathcad | 21 |
| 1.5.1. Побудова поверхонь по матриці аплікат. | 21 |
| 1.5.2. Побудова за допомогою функції CreateMesh | 22 |
| 1.5.3. Побудова поверхонь обертання | 23 |
| 1.5.4. Форматування тривимірних графіків | 24 |
| 1.5.5. Приклади для самостійного вирозв'язання | 27 |
| Контрольні запитання | 28 |
| 2. Фрактальна 3D-графіка | 29 |
| 2.1. Фрактали | 29 |
| 2.2. Сфера застосування фракталів | 30 |
| 2.3. Математичний опис фракталів | 32 |
| 2.4. Задавання на самостійну роботу | 40 |
| 2.5. Інструментальні засоби для створення 3D фракталів | 43 |
| 2.5.1. XenoDream | 43 |
| 2.5.2. Incendia | 47 |
| 2.6. Експорт фрактала з Incendia у 3ds max | 55 |
| 2.7. Практичний приклад | 58 |
| 2.8. Задавання для самостійної роботи | 64 |
| Контрольні запитання | 65 |
| Розділ 2. Полігональні 3D-об'єкти | 66 |

| | |
|---|-----|
| 3. Полігональна 3D-графіка | 66 |
| 3.1. Класифікація подання тривимірних геометричних об'єктів | 66 |
| 3.1.1. Простий спосіб задавання тривимірних об'єктів | 66 |
| 3.1.2. Граничний спосіб задавання тривимірних об'єктів | 66 |
| 3.1.3. Об'ємний спосіб задавання тривимірних об'єктів | 68 |
| 3.2. Моделювання за допомогою редагованих поверхонь | 69 |
| 3.2.1. Полігональні сітки | 69 |
| 3.3. Редаговані поверхні у 3ds max | 72 |
| 3.3.1. Основи полігонального моделювання | 73 |
| 3.4. Полігональне моделювання на конкретних прикладах | 79 |
| 3.4.1. Чашка з ручкою | 79 |
| 3.4.2. Склянка | 84 |
| 3.4.3. Попільничка | 87 |
| 3.4.4. Ложка | 89 |
| 3.4.5. Плитка шоколаду | 93 |
| 3.4.6. Подушка | 97 |
| 3.4.7. Тюбик | 100 |
| 3.4.8. Колона | 105 |
| Контрольні запитання | 107 |
| 4. Сплайн 3D-графіка | 108 |
| 4.1. Інтерполяція сплайна | 108 |
| 4.2. Поліноміальна сплайн-інтерполяція | 109 |
| 4.3. NURBS | 114 |
| 4.4. Моделювання сплайна у 3ds max | 115 |
| 4.5. Приклад моделювання штампа | 123 |
| Контрольні запитання | 130 |
| Література | 131 |

НАВЧАЛЬНЕ ВИДАННЯ

Гаврилов Володимир Петрович

**3D-графіка
навчальний посібник**

Відповідальний за випуск О. І. Пушкар

Відповідальний редактор М. М. Оленич

Редактор О. С. Новицка

Коректор

План 2018 р. Поз. № 26-ЕНП

Підп. до друку Формат 60 × 90 1/16. Папір MultiCopy. Друк Riso.

Ум. – друк арк. 10 Обл. – вид. арк. Тираж 400 прим. Зам. №

*Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
ДК №481 від 13.06.2001 р.*

Видавець і виготовлювач ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А