

**СПЕЦИАЛИЗИРАН НАУЧЕН СЪВЕТ ПО ИНФОРМАТИКА И
МАТЕМАТИЧЕСКО МОДЕЛИРАНЕ ПРИ ВАК**

**ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ
“СВ. СВ. КИРИЛ И МЕТОДИЙ”**

Емилиян Георгиев Петков

**ИЗСЛЕДВАНЕ И РАЗВИТИЕ
НА МОДЕЛИТЕ НА КРИВИ И ПОВЪРХНИНИ
В КОМПЮТЪРНАТА ГРАФИКА**

ДИСЕРТАЦИЯ

за присъждане на образователната и научна степен “Доктор”
по научна специалност 01.01.12 “Информатика”

Научен консултант: доц. д-р Любен И. Цеков

Велико Търново, 2008 г.

Съдържание

УВОД	4
Глава първа	
ГЕОМЕТРИЧНОТО МОДЕЛИРАНЕ В КОМПЮТЪРНАТА ГРАФИКА.....	8
1.1. ГРАФИЧНА СИСТЕМА ЗА ГЕОМЕТРИЧНО МОДЕЛИРАНЕ	8
1.1.1. <i>Приложна програма</i>	<i>9</i>
1.1.2. <i>Базов графичен пакет</i>	<i>9</i>
1.1.3. <i>Преносимост и графични стандарти.....</i>	<i>10</i>
1.1.4. <i>Синтезиране на реалистични изображения.....</i>	<i>10</i>
1.2. АНАЛИТИЧНО ПРЕДСТАВЯНЕ НА КРИВИ ЛИНИИ В ГЕОМЕТРИЧНОТО МОДЕЛИРАНЕ	12
1.2.1. <i>Основна форма за представяне на криви</i>	<i>13</i>
1.2.2. <i>Криви на Безие.....</i>	<i>14</i>
1.2.2.1. <i>Рационални криви на Безие</i>	<i>16</i>
1.2.2.2. <i>Съставни криви на Безие</i>	<i>18</i>
1.2.3. <i>В-сплайн криви.....</i>	<i>18</i>
1.2.3.1. <i>Сплайн функции и криви.....</i>	<i>19</i>
1.2.3.2. <i>В-сплайн функции</i>	<i>20</i>
1.2.3.3. <i>В-сплайн криви</i>	<i>22</i>
1.2.3.4. <i>Параметризация на В-сплайн криви</i>	<i>23</i>
1.2.3.5. <i>Рационални В-сплайн криви.....</i>	<i>24</i>
1.2.4. <i>NURBS криви</i>	<i>25</i>
1.3. АНАЛИТИЧНО ПРЕДСТАВЯНЕ НА ПОВЪРХНИНИ В ГЕОМЕТРИЧНОТО МОДЕЛИРАНЕ	28
1.3.1. <i>Модели на повърхнини.....</i>	<i>29</i>
1.3.2. <i>Повърхнини в Безие-Бернщайн базиса</i>	<i>30</i>
1.3.3. <i>В-сплайн повърхнини</i>	<i>31</i>
1.3.4. <i>NURBS повърхнини</i>	<i>32</i>
1.3.5. <i>Методи за построяване на повърхнини.....</i>	<i>35</i>
1.3.5.1. <i>Ротационни повърхнини</i>	<i>35</i>
1.3.5.2. <i>Повърхнини на замятане.....</i>	<i>35</i>
1.3.5.3. <i>Повърхнини от парчета</i>	<i>35</i>
1.3.5.4. <i>Праволинейни повърхнини.....</i>	<i>35</i>
1.4. МЕТОДИ ЗА ПРЕДСТАВЯНЕ НА КВАДРАТИЧНИ КРИВИ И ПОВЪРХНИНИ В ГЕОМЕТРИЧНОТО МОДЕЛИРАНЕ.....	36
1.5. Изводи	41

Глава втора

ПРЕДСТАВЯНЕ НА КРИВИТЕ ОТ ВТОРА СТЕПЕН ЧРЕЗ NURBS КРИВИ.....	44
2.1. Аналитично представяне на кривите от втора степен	44
2.2 Идеята за представяне на кривите от втора степен чрез NURBS криви.....	48
2.3. Представяне на дъга от елипса чрез NURBS крива	49
2.3.1. Посока на построяване на дъгата	50
2.3.2. Основна идея за решение.....	51
2.3.3. Афинна трансформация на единична окръжност в елипса.....	51
2.3.4. Алгоритъм за построяване на NURBS крива, представяща дъга от единичната окръжност.....	52
2.3.5. Представяне на дъга от елипса чрез дъга от единичната окръжност представена чрез NURBS крива от втора степен	55
2.4. Представяне на дъга от парабола чрез NURBS крива	58
2.5. Представяне на дъга от хипербола чрез NURBS крива	62
2.6 Изводи	74

Глава трета

ПРЕДСТАВЯНЕ НА ПОВЪРХНИНИТЕ ОТ ВТОРА СТЕПЕН ЧРЕЗ NURBS	
ПОВЪРХНИНИ	76
3.1. Аналитично представяне на повърхнините от втора степен	76
3.2. Представяне на повърхнините от втора степен чрез биквадратни NURBS	
ПОВЪРХНИНИ	81
3.2.1. Представяне на елипсоид, параболоид, хиперболоид, двоен хиперболоид и конус чрез биквадратни NURBS повърхнини	82
3.2.1.1. Построяване на ротационна биквадратна NURBS повърхнина	83
3.2.1.2. Построяване на NURBS елипсоид-повърхнина (NE-повърхнина).....	88
3.2.1.3. Построяване на NURBS параболоид-повърхнина (NP-повърхнина)	91
3.2.1.4. Построяване на NURBS хиперболоид-повърхнина (NH-повърхнина)	92
3.2.1.5. Построяване на NURBS двоен-хиперболоид-повърхнина (NDH-повърхнина).....	94
3.2.1.6. Построяване на NURBS конус-повърхнина (NC-повърхнина)	96
3.2.2. Представяне на цилиндрите чрез биквадратни NURBS повърхнини	97
3.2.3. Представяне на хиперболичен параболоид чрез биквадратна NURBS повърхнина.....	102
3.2.4. Подход за подразделяне на биквадратна NURBS повърхнина без вътрешни възли в U и V	105
3.3 Изводи	107

Глава четвърта

ПРИЛАГАНЕ НА NURBS МОДЕЛИТЕ НА КРИВИТЕ И ПОВЪРХНИНТЕ ОТ ВТОРА СТЕПЕН В 3D ГРАФИЧНАТА СИСТЕМА ЗА МОДЕЛИРАНЕ AUTODESK 3DS MAX . 109

4.1. Възможности за функционално разширяване на 3D графичните системи за моделиране.....	109
4.2. Избор на 3D графична система за моделиране, притежаваща средства за функционално разширяване	110
4.3. Разработване на специализираните модули.....	113
4.3.1. Избор на софтуерен инструментариум.....	113
4.3.2. Графичен потребителски интерфейс на CCM.....	114
4.3.3. Функции създаващи новите NURBS модели в CCM.....	118
4.3.4. Интерактивно създаване на NURBS обектите.....	120
4.3.5. Създаване на NURBS обектите в различните изгледи	122
4.3.6. Допустими стойности на параметрите на CCM	123
4.4 Повдигане на степента на NURBS кривите и повърхнините	127
4.5. Сравнителен анализ и демонстрация на работата на CCM	127
4.5.1. Построяване на NURBS дъга от елипса	128
4.5.2. Построяване на NURBS дъга от парабола	130
4.5.3. Построяване на NURBS дъга от хипербола	131
4.5.4. Построяване на NURBS елипсоид	132
4.5.5. Построяване на NURBS параболоид	134
4.5.6. Построяване на NURBS хиперболоид	136
4.5.7. Построяване на NURBS двоен хиперболоид	138
4.5.8. Построяване на NURBS конус.....	139
4.5.9. Построяване на NURBS елиптичен цилиндър.....	140
4.5.10. Построяване на NURBS параболичен цилиндър	142
4.5.11. Построяване на NURBS хиперболичен цилиндър	143
4.5.12. Построяване на NURBS хиперболичен параболоид.....	144
4.6. Изводи	145
ЗАКЛЮЧЕНИЕ.....	147
ЛИТЕРАТУРА	150

УВОД

Компютърната графика (КГ) е един от дяловете на информатиката, свързани със създаването, съхранението и обработката на визуална информация, т.е. симулирането на условията на реалния свят по начин, полезен за човека [1, 5, 17, 21, 30, 46, 100]. Предметът на компютърната графика включва *синтез, анализ и обработка на изображения* [9, 12, 15, 22, 133]. Компютърните приложни програми, разработени с тази цел, се наричат *графични приложения*, а по-сложните от тях (подчинени на системния подход) – *графични системи* [126, 131, 134, 143]. Те изпълняват основните си задачи на базата на подходящи описания - *компютърни модели*, отчитащи основните параметри и характеристики на реалните графични обекти и процеси.

Един от най-важните дялове в КГ е *геометричното моделиране* (ГМ) (Computer Aided Geometric Design – CAGD) [40, 60, 75, 76, 85, 96, 124]. Основната му задача е намирането на математически методи и алгоритми за представяне и обработване на геометрични фигури с произволни криволинейни граници [8, 61].

Основен **обект на изследване** в тази дисертация са възможностите за развитие на моделите на криви и повърхнини от втора степен в КГ и методите и алгоритмите за моделиране чрез тях на графични обекти в триизмерните (3D¹) графични приложения – като част от основната задача за синтез на изображения.

Задачите на КГ в това направление са свързани с намирането на подходящи математически модели за представяне на криволинейните форми на реални обекти в равнината и пространството [7, 28, 56, 74, 97]. Съществуват много такива модели [60, 97], но за практиката те винаги не са достатъчен брой, защото най-често се търсят и избират тези от тях, които отговарят на определени условия. Например: моделите най-лесно да се представят на езика на компютърната система; чрез тях да се задават основните геометрични примитиви; над тях да могат да се прилагат всички геометрични трансформации; да са база за разработване на бързи алгоритми; да позволяват лесно интерактивно моделиране на свободни графични форми и т.н.

Този, наложил се повсеместно, потребителски подход прави задачата за създаване на все по-нови геометрични модели винаги актуална.

От горепосочената гледна точка такива математически модели на криви и повърхнини, които най-широко се използват в съвременните графични системи са Bezier, Spline, B-spline, NURBS [65, 66, 68, 71, 78, 86, 91, 101, 105, 106].

Понастоящем основните очаквания към възможностите на триизмерните графични системи все повече се насочват към разработване на специфични набори от

¹ Съкращение от английски език: 3-Dimentional – триизмерен.

модели, усъвършенстващи реалистичното представяне на различни обекти и процеси, като: движения, деформации, влияния на странични елементи и ефекти върху обектите и т.н. [111, 116, 133].

Това определи в обхвата на разработката да се разгледат и изследват една група от обекти притежаващи специфични и важни свойства и да се предложат полезни модели, по-ефективни алгоритми и техники за тяхното прилагане в графичните системи като *графични примитиви за моделиране*², разширявайки техния *графичен инструментариум*³. Това са линиите и повърхнините от втора степен (квадратични обекти), които се представят с общо алгебрично уравнение от втора степен, породено от квадратична форма [13, 27, 33]. Реалните, неразпадащи се криви от втора степен са познатите от геометрията *окръжност, елипса, парабола и хипербола*, наричани, заради геометричната им природа, още *конични сечения (коники)* [55, 120, 125, 128, 130, 136, 138, 144]. Повърхнините от втора степен са *сфера, елипсоид, елиптичен параболоид*, (прост, едно-повърхнинен) *хиперболоид, двоен хиперболоид, конуси и цилиндри с управителна квадратична крива, хиперболичен параболоид* [33, 132, 142]. Тези обекти притежават важни геометрични свойства [33, 50, 52, 59, 139]. Тяхното широко приложение може да се види в [70, 73, 79, 81, 87, 113, 114, 115, 121, 140, 141].

Методологията на изследването включва следните подходи: критичен евристичен анализ на проблемите в избраната научно-практическа област на изследване за формиране на обосновани изводи и издигане на нови идеи и хипотези; търсене, изследване и ефективно развитие на математически модели; развитие и използване на принципи на геометрията и геометричното моделиране; евристичен синтез и сравнение на известни и нови решения за намиране на по-добри алгоритми и техники. В хода на изследването са използвани средата за разработка на приложения MS Visual C++ 6.0 [14, 31, 32], в която са разработени тестови приложения, и Maple 8 [135] за някои математически пресмятания и фигури.

Дисертацията се състои от увод, четири глави, заключение и използвана литература.

В първа глава е изследвана обобщена структура на графична система за геометрично моделиране като се набляга на значението и мястото на геометричния модел в нея, както и на процеса на синтезиране на изображения в 3D приложенията. Показано е аналитичното представяне на модели на криви и повърхнини в ГМ, които се явяват база за разработване на предложените в това изследване модели. Това са

² Основни обекти за моделиране в графичните системи. Обикновено това са добре познати геометрични обекти, които се създават лесно на базата на няколко параметъра. Най-често създаването на модел на по-сложен обект става от един или няколко примитива, които се моделират за да се получи желаната форма.

³ Това е наборът от примитиви и функции подпомагащи процеса на моделиране в една графична система.

Безие, B-сплайн и NURBS моделите на криви и повърхнини. Направен е обзоре анализ на съществуващите модели и подходи за построяване на кривите и повърхнините от втора степен в ГМ. На тази основа са формулирани основната *цел* на дисертацията и произтичащите от нея основни *задачи*.

При решаването на поставените задачи по-нататък разработката е разделена на две части. В първата част е разгледано представянето на кривите, а във втората – представянето на повърхнините. В тези две части е разработен NURBS модела на всяка една от квадратичните криви и повърхнини.

Втора глава съдържа аналитичното представяне на кривите чрез параметричните им уравнения. Предложени са модели на NURBS криви, представящи: елипса, парабола и хипербола. При представянето на някои от обектите са приложени авторски подходи. Във формулираните в края на главата изводи са посочени предимствата на разработените модели.

Трета глава съдържа аналитичното представяне на повърхнините чрез параметричните им уравнения. Предложени са модели на NURBS повърхнини, представящи: сфера, елипсоид, елиптичен параболоид, прост хиперболоид, двоен хиперболоид, конус и елиптичен цилиндър, параболичен цилиндър, хиперболичен цилиндър и хиперболичен параболоид. Моделите са разработени на базата на известни подходи за построяване на повърхнини, но приложени за NURBS обекти. Във формулираните в края на главата изводи са посочени предимствата на предложените модели.

Четвърта глава отразява изследването за приложимост на предложените във втора и трета глава NURBS модели на криви и повърхнини в една от широко разпространените 3D графични системи за моделиране Autodesk 3ds Max. Това е направено чрез разработване на специализирани софтуерни модули-добавки (плъгини, от англ. plugins) за това приложение. Направен е сравнителен анализ на работа им и примитивите, които те създават, спрямо съществуващите в системата инструменти и техники за построяването на този род обекти.

В заключението е направено обобщение на резултатите, които са получени. Формулирани са основните резултати в дисертацията. Посочени са някои актуални задачи, които могат да бъдат естествено продължение на това изследване.

Разработката и апробацията на резултатите са извършени в катедра “Компютърни системи и технологии” на Великотърновския университет “Св. Св. Кирил и Методий”.

Получените резултати са публикувани в седем доклада и една научна статия в периода след 2002 г. От тях един брой е в чужбина.

Дванадесетте специализирани софтуерни модула, които бяха разработени за 3D графичната система за моделиране 3ds Max, са одобрени и приети от Internet портала за високи 3D технологии и приложения *HighEnd3D* и са предоставени за свободно изтегляне и ползване с нетърговски цели.

Изказвам голямата си благодарност и признателност на научния си консултант доц. д-р Любен Цеков за всичко, което съм научил от него, за помощта и отзивчивостта му, за препоръките и ценните съвети по темата на дисертацията, преподавателската ми дейност и извън тях. Благодарен съм на доц. д-р Маргарита Тодорова за подкрепата през цялото време на разработване на дисертацията и за възможностите, които ми предостави да посетя висши учебни заведения в страни от Европейски съюз. Благодарен съм на доц. д-р Милен Христов за времето, което ми отдели и помощта, която ми оказа по геометрия. Благодарен съм на колегите си от катедра “Компютърни системи и технологии” при ВТУ “Св. Св. Кирил и Методий” за съпричастността им към работата ми и за разбирането и помощта, която ми оказаха. Благодарен съм на всички колеги от направление “Математика и информатика” на Педагогически факултет при ВТУ “Св. Св. Кирил и Методий”, които са ми помагали и насърчавали. Благодарен съм на семейството ми за подкрепата и вярата им в мен.

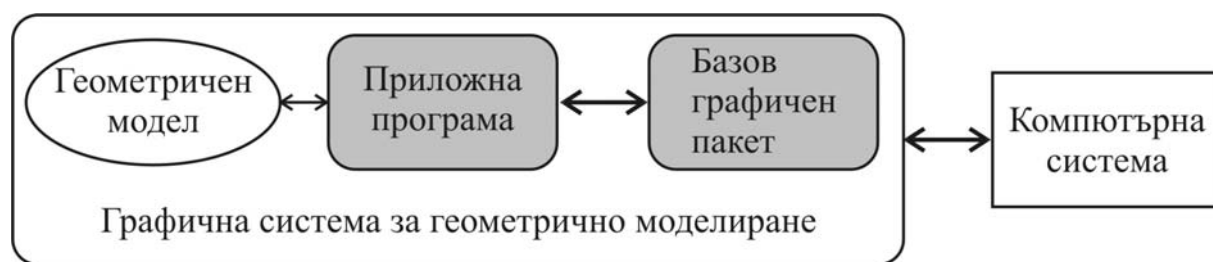
ГЕОМЕТРИЧНОТО МОДЕЛИРАНЕ В КОМПЮТЪРНАТА ГРАФИКА

Обхватът на настоящето изследване налага да се направи обзор на състоянието и проблемите на ГМ. От особено значение е да се открият ефективните модели на криви и повърхнини, както и да се направи обзорен анализ на съществуващите методи и алгоритми за построяване на кривите и повърхнини от втора степен в 3D-графичните системи.

Геометричното моделиране е в основата на математическите модели, чрез които е възможно компютърното моделиране на реални обекти и явления [8, 61]. Това включва представянето, построяването, преобразуванията и апроксимацията на криви и повърхнини. Практическите приложения на ГМ са широко разпространени в много области. Например: геометричните модели са в основата на моделирането на форми за архитектурни сгради, инженерни изделия, облекла и други практически произведения, компютърни игри, симулатори, анимация и др. Графичните модели могат да бъдат форми на предмети, живи организми и физични явления [24, 42, 47].

1.1. Графична система за геометрично моделиране

Графичната система за геометрично моделиране (ГСГМ) (фиг. 1.1) има два основни компонента [5, 6, 17]: *приложна програма* (ПП) и *базов графичен пакет* (БГП).



Фиг. 1.1 Графична система за геометрично моделиране.

Приложната програма се грижи за създаването на геометричен модел, извършването на операции върху него, както и за организирането му във вид, удобен за визуализиране.

Вторият компонент – *базов графичен пакет* – предоставя набор от технически и програмни средства за визуализация.

1.1.1. Приложна програма

Всяка графична ПП отразява спецификата на съответната приложна област, но въпреки различията, всяка от тях извършва три основни дейности: *моделиране; описание на модела за БГП; интерактивна работа с модела*. Следователно, една от основните задачи при съставянето на ПП е да се избере наборът от обекти и начинът, по който те и връзките между тях да се представят, за да бъдат обработвани от нея. За решаването на тази задача ПП построява и използва *геометричен модел* (ГМ) на обектите. Геометричният модел е компютърното представяне на *формата, разположението, ориентацията и размерите* на обектите на ПП, а *моделирането е процеса на създаване или използване на геометричния модел*. Както всеки друг модел, така и ГМ се създава, за да може върху него ПП да извърши разнообразни анализи и изследвания, вместо това да се прави с обектите от реалния свят. Геометричният модел от своя страна се състои от определен тип обекти, които имат своя метрика, топология, атрибути, неграфични данни и обработващи алгоритми. Средствата за моделиране се предлагат от БГП, а ПП е тази, която създава модела.

Описанието на геометричната форма в модела е пряко свързано с поставения приложен проблем и е напълно независимо от БГП. За да бъде визуализиран приложния геометричен модел или някаква част от него, е необходимо той да бъде представен чрез формалния език (синтаксис, семантика и лексика) и понятийната система на БГП, който извършва самото изобразяване.

Взаимодействието между графичните ПП и потребителя се осъществява с помощта на графичния потребителски интерфейс, а диалогът най-често има сценариеен характер.

1.1.2. Базов графичен пакет

Базовият графичен пакет е сравнително независим от решаваните приложни задачи. Неговите функции са да обслужва графичния вход и изход при наличните графични устройства. Една примерна структура на БГП от три нива, може да бъде видяна в [17, 21-22 с.].

Все по-голямо внимание в съвременните БГП се отделя на предоставяне на средства за по-лесно описание на моделите: визуализация на пространствени обекти, реалистична визуализация, наличие на примитиви като рационални криви и произволни повърхнини и др.

1.1.3. Преносимост и графични стандарти

За да бъде осигурена преносимост на ГСГМ от една платформа на друга се създават спецификации за БГП [17, 91]. Първата такава документация е направена през 1977 г. за системата Core. След това през 1985 г. се появява и спецификация за системата GKS (Graphics Kernel System), но за разлика от Core, GKS е само двумерна система и през 1988 г. се появява и тримерната GKS-3D. Тя бързо бива изместена от също тримерната, но с много по-големи възможности система PHIGS (Programmer's Hierarchical Interactive Graphics System). Появява се и спецификацията на PEX (PHIGS Extension to X Window System), чиято цел е да съчетае възможностите за моделиране на PHIGS и интерактивните средства на X Window System. PHIGS е първата система, която е ориентирана не само към визуализация, но и към моделиране. В тази система се включват като примитиви и рационални криви и повърхнини.

Появяват се също Glide, разработен за 3Dfx, Heidi, разработен от Autodesk за 3D ускоряване, използван от програми като AutoCAD и 3D Studio MAX, Iris GL разработен от SGI и по-късно преименувана на OpenGL (Open Graphics Library) [57, 88, 137]. OpenGL е графична библиотека създадена за да надгражда графичните възможности на операционните системи. Поради това тази библиотека се нарича и графичен приложен програмен интерфейс (Graphics Application Programming Interface - GAPI). Microsoft също разработва GAPI, наречен DirectX Graphics [127].

Друго направление в стандартизацията става обменът на графична информация. Създават се файлови формати за съхраняване на растерни изображения (GIF, TIFF, JPEG, TGA, PNG и др.), на двумерни векторни графики (EPS, WMF, EMF, CGM, SVG, DWG, PLT, CMX, WPG, SWF и др.), на тримерни модели (3DS, ASE, DWG, DXF, IGS, W3D, WRL и др.), както и за анимация (AVI, MPEG и др.) [10, 20, 36, 37, 39, 54, 58, 84, 89, 109].

1.1.4. Синтезиране на реалистични изображения

Процесът на създаване на реалистични изображения в 3D ГСГМ може да бъде представен като изпълнение на три последователни и взаимосвързани етапа, наречени: *моделиране*, *рендъринг* (от англ. *rendering*) и *изобразяване* (визуализация), както е показано на фиг. 1.2.



Фиг. 1.2 Етапи в синтезирането на реалистични изображения.

Тримерното моделиране включва подходящо представяне в ПП на тримерните обекти съставлящи сцената и друга информация необходима за синтеза на изображението, във формата на структурирани числени данни и/или алгоритми. С други думи, това е математическо описание на *3D модела* на сцената, което е подадено като входна информация към рендъринга.

Тримерният модел съдържа три типа данни: *данни за обектите*, *данни за светлинните източници* и *параметри за визуализация или виртуална камера*.

Следвайки процеса на синтеза на изображение, описания 3D модел след моделирането на сцената се подава като входна информация за рендъринга. Рендърингът като процес представлява извършване на селекция и необходими трансформации на данните в 3D модела с цел да се получи двумерно представяне на сцената в проекционна равнина. Той съдържа няколко основни стъпки симулиращи геометричните свойства на централна проекция и физичните свойства в човешкото зрение или фотографията. Главните задачи които се решават са: осъществяване на централна проекция (с краен или безкраен център), определяне на видимите повърхности и шейдинг (определяне цвета и нюанса на всеки пиксел в изображението).

Крайният етап е *изобразяване* на синтезираното изображение, който възпроизвежда на изходно графично устройство картината от двумерното представяне, получено по време на рендъринга. Стойностите на цветовете на индивидуалните пиксели са преработени и представени във вид удобен за изобразяване.

Сравнителни характеристики на някои от основните 3D ГСГМ [110] и техните области на приложение могат да бъдат намерени в [122, 129].

Възникват няколко въпроса по отношение на функционалната приложимост на всяка от 3D графичните системи за моделиране:

- Съществува ли компютърен модел на обекта, който трябва да бъде моделиран в графичния проект?

- Достатъчни ли са геометричните примитиви, които се предоставят от 3D графичните системи?
- Може ли да има по-голямо разнообразие от геометрични примитиви, които притежават специфични свойства, необходими при представянето на реални обекти и процеси?

1.2. Аналитично представяне на криви линии в геометричното моделиране

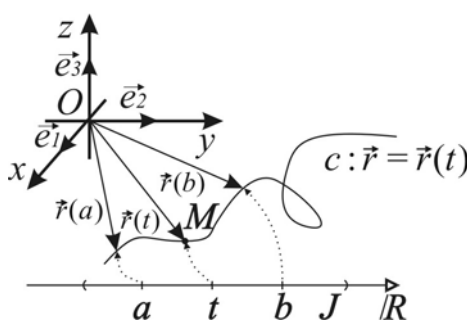
В ГМ кривите се задават основно по два начина, чрез *общо (неявно) уравнение* и чрез *параметрични уравнения* във *векторна* или в *скаларна форма* [13, 17, 27, 33].

Общото уравнение на крива от втора степен, има вида $F(x, y) = 0$, където $F(x, y)$ е полином от втора степен на две променливи:

$$F(x, y) = a_{11}x^2 + a_{22}y^2 + 2a_{12}xy + 2b_1x + 2b_2y + c. \quad (1.1)$$

Чрез теоремата за съществуване и единственост на неявна функция, уравнението $F(x, y) = 0$ задава линията $c: y = f(x)$ локално около т. $P_0(x_0, y_0)$ ако $F(x_0, y_0) = 0$ и $F'_y(x_0, y_0) \neq 0$ [18].

Нека в \mathbb{R}^3 да е фиксирана дясна декартова координатна система $K = \{O, \vec{e}_1, \vec{e}_2, \vec{e}_3\}$ (фиг. 1.3). Гладка пространствена линия c се представя параметрично във вида $c: \vec{r} = \vec{r}(t)$, където $\vec{r}(t) = (x(t), y(t), z(t)) \in C^1$ е гладка векторна функция на един реален аргумент (параметър на кривата), дефинирана в интервал $J \subseteq \mathbb{R}$ и с неанулираща се в J непрекъсната производна [34].



Фиг. 1.3 Задаване на крива линия чрез векторна функция.

В скаларна форма, всяка една от координатите на точка от кривата се представя отделно като явно изразена функция на параметъра:

$$c: \begin{cases} x = x(t) \\ y = y(t), \quad t \in [a, b]. \\ z = z(t) \end{cases} \quad (1.2)$$

Ако $\vec{r}(t)$ е непрекъсната векторна функция в $[a, b]$, тогава *крива линия* е множеството от точки M , за които вектора $\vec{p} = \overrightarrow{OM} = r(t), t \in [a, b]$ (фиг. 1.3). Условието за гладкост на линията гарантира съществуването на ненулев допирателен вектор във всяка нейна вътрешна точка. Интервалът $[a, b]$ е произволен, но обикновено (с точност до репараметризация на кривата) е нормализиран до $[0, 1]$.

Параметричното задаване се явява по-естествено за представянето и дизайна на форми чрез компютър [91, 1-4 с.]. От друга страна - коефициентите на много параметрични функции (като Bézier и B-spline), притежават важен геометричен смисъл. Това се отчита в методите за интуитивен дизайн и числено стабилни алгоритми с видимо добър геометричен резултат.

При проектирането на една ГСГМ се извършва единно съгласуване на координатните функции. Най-доброто за реализацията е да се ограничи представянето на кривите до клас от функции [91, 5 с.], които отговарят на следните критерии:

- имат не сложно за реализация аналитично задаване и са математически добре дефинирани;
- предоставят възможност за представяне на всички криви, от които се нуждае потребителят на системата;
- лесно, ефикасно и точно се представят чрез програмен език.

1.2.1. Основна форма за представяне на криви

Широко използван клас от функции са *полиномите*. Те са в основата на методите за интерполиране на функции [4, 5, 23, 26]. Полиномиална крива от n -та степен във нормална форма се задава във вида:

$$\vec{r}(t) = (x(t), y(t), z(t)) = \sum_{i=0}^n a_i t^i, \quad t \in [0, 1], \quad (1.3)$$

където $a_i = (x_i, y_i, z_i)$ са константни реални вектори, т.е. координатните функции са полиноми от n -та степен:

$$x(t) = \sum_{i=0}^n x_i t^i, \quad y(t) = \sum_{i=0}^n y_i t^i, \quad z(t) = \sum_{i=0}^n z_i t^i. \quad (1.4)$$

Последователното диференциране на уравнение (1.3) води до $a_i = \frac{r^{(i)}(t)|_{t=0}}{i!}$, където $r^{(i)}(t)|_{t=0}$ е i -тата производна на $\vec{r}(t)$ в $t=0$, $i=0,\dots,n$. Множеството от $n+1$ функции $\{1, t, t^2, \dots, t^n\}$ се наричат *базисни функции* (те са стандартния базис за \mathbb{R} -линейното пространство $P_n[x]$ от полиномите от n -та степен с реален аргумент x), а фамилията константни вектори $\{a_i\}$ - *коэффициенти*.

За дадено t_0 , точката $\vec{r}(t_0)$ на полиномиална крива от степен n , записана в нормален вид, се изчислява най-ефективно по метода на Хорнер:

$$\vec{r}(t_0) = \left(\left(\dots (a_n t_0 + a_{n-1}) t_0 + a_{n-2} \right) t_0 + \dots + a_0 \right).$$

Въпреки, че полиномите като базис удовлетворяват последните два от посочените по-горе критерии към координатните функции, съществува голямо множество от важни типове криви, които не могат да бъдат точно представени чрез тях. Този базис има следните недостатъци [91, 9 с.]:

- Редицата от функции $1, t, t^2, \dots, t^n$ не е равномерно сходяща върху интервала $[0,1]$.
- Той е неестествен за интерактивен дизайн. Коэффициентите $\{a_i\}$ носят много малка геометрична информация за формата на кривата. Дизайнерите обикновено искат да специфицират крайните състояния в двата края на линията, не само в началната точка.
- Изчислителните алгоритми имат повече алгебричен отколкото геометричен смисъл (метода на Хорнер).
- Числено, е слаба форма. Методът на Хорнер предразполага към грешки при закръгляне ако коэффициенти варират в голям диапазон.

1.2.2. Криви на Безие

Формата на Безие (*Bézier*) за полиномиална крива [34, 41, 56, 91] е частен случай на един по-общ метод, основаващ се на разлагането на полиноми по базиса от полиномите на Бернщайн. Този базис за $P_n[t]$ се нарича *Безие-Бернщайн базис*. При известни радиус-вектори на $n+1$ характеристични точки $M_i(x_i, y_i, z_i)$ в пространството, векторно-параметричното уравнение на кривата на Безие (с характеристична начупена линия $M_0 M_1 M_2 \dots M_n$) е:

$$c: \vec{r} = \vec{r}(t) = \sum_{i=0}^n M_i B_{i,n}(t), \quad t \in [0,1] \quad (1.5)$$

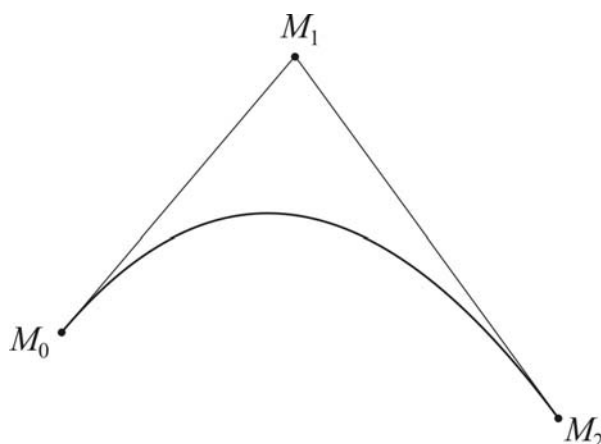
където $B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$ е i -ят полином на Бернщайн от n -та степен и където, с цел техническо удобство и следвайки означенията в [17], тук и по-нататък в работата радиус-векторът на точка M_i е означен отново с M_i , което ще се подразбира от контекста на векторните равенства.

Пример 1.1 За да се построи крива на Безие от степен $n=2$ се изчисляват полиномите на Бернщайн от втора степен:

$$B_{0,2}(t) = (1-t)^2, \quad B_{1,2}(t) = 2t(1-t), \quad B_{2,2}(t) = t^2.$$

Тогава за векторното уравнение на кривата (фиг. 1.4) получаваме:

$$c: \vec{r} = \vec{r}(t) = (1-t)^2 M_0 + 2t(1-t) M_1 + t^2 M_2.$$



Фиг. 1.4 Крива на Безие от втора степен.

Кривата на Безие от n -та степен минава през началото M_0 и края M_n на *характеристичната си начупена* линия. Двете двойки крайни точки M_0M_1 и $M_{n-1}M_n$ задават допирателните прави за c в двата края:

$$\begin{aligned} \vec{r}(0) &= M_0 & \vec{r}'(0) &= n(M_1 - M_0) \\ \vec{r}(1) &= M_n & \vec{r}'(1) &= n(M_n - M_{n-1}), \quad \text{защото:} \end{aligned}$$

$$\frac{d}{dt} r(t) = n \left[\sum_{i=1}^n M_i B_{i-1,n-1}(t) - \sum_{i=0}^{n-1} M_i B_{i,n-1}(t) \right] = n \sum_{i=0}^{n-1} (M_{i+1} - M_i) B_{i,n-1}(t). \quad (1.6)$$

Намирането на линия от произволна степен n става чрез рекурентната връзка:

$$B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t),$$

$$\vec{r}(t) = \sum_{i=0}^{n-1} M_i B_{i,n-1}(t)(1-t) + \sum_{i=1}^n M_i B_{i-1,n-1}(t)(t) = \sum_{i=0}^{n-1} \{M_i(1-t) + M_{i+1}t\} B_{i,n-1}(t). \quad (1.7)$$

1.2.2.1. Рационални криви на Безие

Рационална крива на Безие от n -та степен, зададена с $n+1$ характеристични точки M_i , в нехомогенни координати се задава чрез векторната функция:

$$c: \vec{r} = \beta(t) = \frac{\sum_{i=0}^n M_i w_i B_{i,n}(t)}{\sum_{j=0}^n w_j B_{j,n}(t)} = \sum_{i=0}^n M_i \left\{ \frac{w_i B_{i,n}(t)}{\sum_{j=0}^n w_j B_{j,n}(t)} \right\}, \quad t \in [0,1]. \quad (1.8)$$

Тук w_i са реални числа, наричани *теглови коефициенти (тегла)* за съответния полином на Бернщайн (1.8). Коефициентите пред M_i в хомогенното представяне, спрямо фиксираните точки $\{M_i\}$, имат от гледна точка на статиката смисъл на *маси* приложени в тези точки и са наречени от Мьобиус барицентрични координати. Чрез промяната на теглата може да се модифицира линията c при фиксирани характеристични точки (при промяна на теглото на съответната точка в линейната комбинация (1.8) се мени центърът на тежестта на системата материални точки).

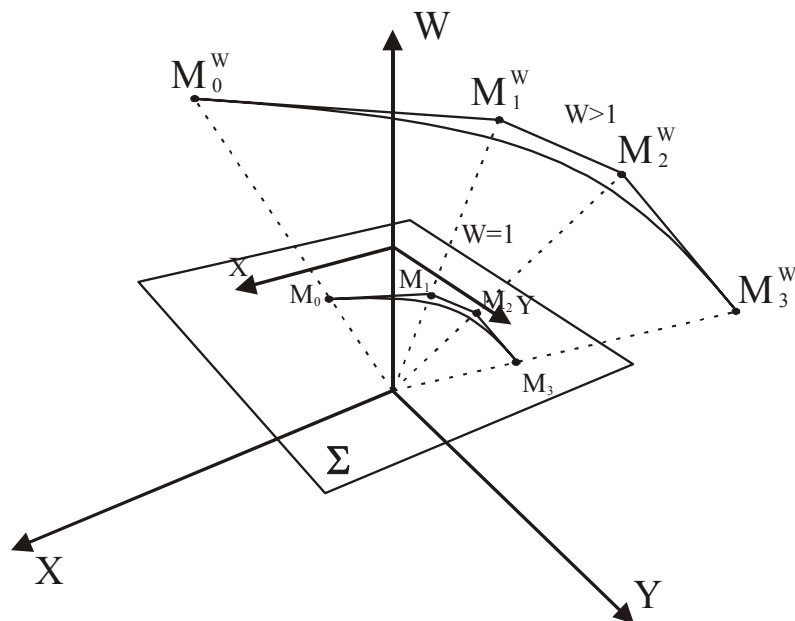
След като се положи:

$$C = \sum_{i=0}^n M_i w_i B_{i,n}(t) \text{ и } W = \sum_{j=0}^n w_j B_{j,n}(t)$$

за първата производна на $\beta(t)$, а оттам и за допирателния вектор към c в произволна точка, се получава:

$$\beta'(t) = \frac{d}{dt} \beta(t) = \frac{1}{W^2} \cdot \begin{vmatrix} C' & W' \\ C & W \end{vmatrix} = \sum_{i=0}^n M_i \cdot \frac{w_i}{w^2} \cdot \begin{vmatrix} B'_{i,n}(t) & W' \\ B_{i,n}(t) & W \end{vmatrix}. \quad (1.9)$$

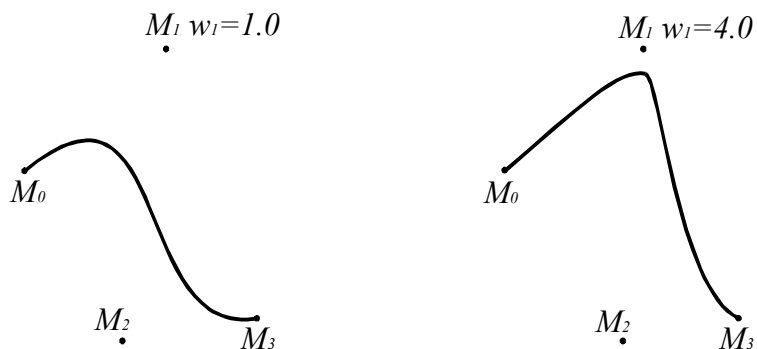
Оттук не е трудно да се провери, че $\beta'(0) \parallel \overrightarrow{M_0 M_1}$, $\beta'(1) \parallel \overrightarrow{M_n M_{n-1}}$. Геометричният смисъл на (1.8) и (1.9) е показан на фиг. 1.5, където е изобразена в перспектива равнинна рационална крива.



Фиг. 1.5 Геометрично представяне на равнинна рационална крива на Безие (от трета степен).

Рационалната крива от n -та степен минава през началото и края на характеристичната си начупена, а двете крайни отсечки са колинеарни с производните на векторната функция в двата края (фиг. 1.5). Ако кривата е от втора степен, тогава характеристичните точки са три M_0, M_1, M_2 като отсечките (от характеристичната начупена) M_0M_1 и M_1M_2 са допирателните в двата края на кривата.

“Притеглянето” на кривата към характеристичните точки зависи от приложените в тях тегла. На фигура 1.6 е показана една и съща крива при две различни стойности на последната координата на точката M_1 , а именно $w_1 = 1.0$ и $w_1 = 4.0$. Най-често теглата на крайните две точки се фиксират на 1.



Фиг. 1.6 Равнинна рационална крива на Безие (от трета степен) при две различни стойности на параметъра w_1 .

Кривите състоящи се от само един полиномиален или рационален сегмент са в повечето случаи са незадоволителни за целите на моделирането. Основни техни недостатъци са:

- Работи се с много висока степен на базисната функция при по-голям брой на характеристични точки, които се подават на алгоритъма за изчисляване на кривата (за $n+1$ точки базисната функция е от степен n). Кривите от по-висока степен са неефективни и изчислително нестабилни.
- Степента на базисната функция многократно се умножава при сглобяване на няколко геометрични форми.
- Едно-сегментните криви не са добре пригодени за интерактивен дизайн (моделиране) на геометрични форми. Въпреки, че кривите на Безие позволяват промяна на формата чрез характеристичните точки и теглата им (ако са рационални), контролът не е достатъчно локален, т.е. при промяна на позицията или теглото на една от точките е нужно преизчисляване (и преизчертаване) на цялата крива.

1.2.2.2. Съставни криви на Безие

Става ясно, че проблем, който съществува в моделирането с криви на Безие, е увеличаването на степента на полинома при увеличаването на характеристичните точки, което прави изчислението й по-сложно [17]. Един метод за намаляването на степента на линията е като тя се раздели на множество от криви на Безие от по-малка степен (например втора или трета). Отделните части на линията се наричат *сегменти*.

Съставната линия трябва да е непрекъсната [17] от преминаването от един сегмент към друг, или от клас C^0 . Когато кривата трябва да има непрекъснат наклон в точката на съединяване, т.е. посоката на допирателните в тези точки съвпадат (без непременно големините им да са равни), се казва, че кривата е от клас G^1 (G - геометрична непрекъснатост). Когато производните в точките на свързване са равни и като вектори, кривата е от клас C^1 (или гладка), т.е. е с непрекъсната и неанулираща се производна.

Тези криви са подходящи, когато гладкостта в точките на свързване на сегментите трябва да може да се контролира и това е от особено значение за процеса на моделиране.

1.2.3. B-сплайн криви

В-сплайн кривите се дефинират на базата на така наречените В-сплайн функции, които се явяват частен случай на сплайн функциите.

1.2.3.1. Сплайн функции и криви

Дефиниция на сплайн функция е дадена в [4, 47-54 с.].

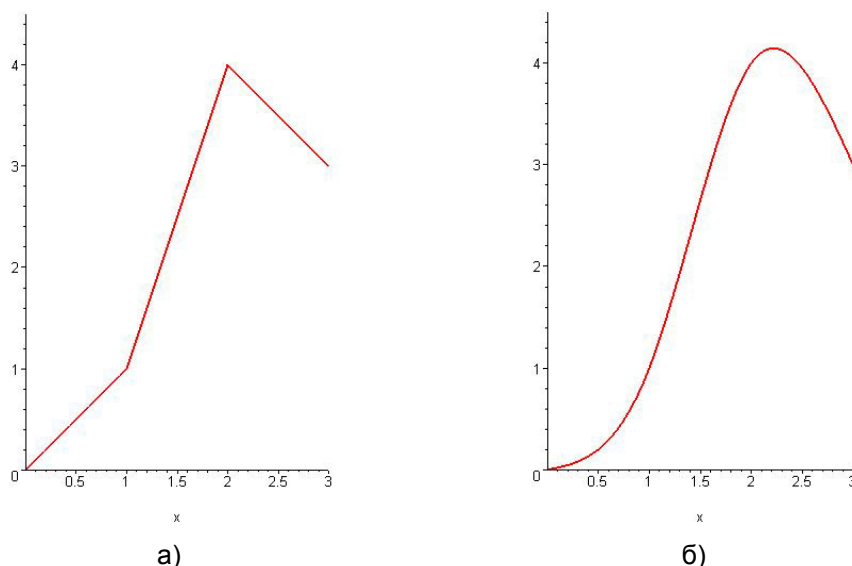
Дефиниция 1.1 Функцията $s(x)$, се нарича *сплайн-функция* от степен p с възли $x_1 < \dots < x_n$, ако:

- $s(x)$ е полином от степен най-много p във всеки подинтервал $[x_i, x_{i+1}]$, $i = 0, \dots, n$, ($x_0, x_{n+1} : x_0 \rightarrow -\infty; x_{n+1} \rightarrow +\infty$);
- $s(x), s'(x), \dots, s^{(p-1)}(x)$ са непрекъснати функции в $(-\infty, +\infty)$.

Съществуват няколко начина за построяване на сплайн крива, които могат да се видят в [17, 266-271 с.] като най-често се работи със сплайн от трета степен. Сплайните са удобно средство за конструиране на криволинейни форми поради:

- непрекъснатостта на производните им, което в случая на непрекъснатата втора производна води до непрекъснатост на кривината им;
- неголямата информация за производните, необходима за тяхното построяване.

На фигура 1.7 са показани сплайн криви, интерполиращи точките $(0,0)$, $(1,1)$, $(2,4)$ и $(3,3)$, съответно а) линеен и б) кубичен сплайн.



Фиг. 1.7 Сплайн криви: а) линеен сплайн и б) кубичен сплайн.

Сплайн кривите имат някои ограничения, най-важните от които са:

- Не е възможна локална модификация: при промяна на една от интерполираните точки целият сплайн трябва да се преизчисли.

- Неприложимост при наличие на вертикални тангенти – това би предизвикало деление на 0.
- Възможна е осцилация при интерполация на точки, принадлежащи на крива с прекъснати втори производни.

Последните два недостатъка се преодоляват донякъде с използването на т.нар. праметрични сплайни.

1.2.3.2. В-сплайн функции

Дефиниция на В-сплайн функция има в [4, 55-62 с.], но тук ще бъде дадена дефиниция за *нормирана* В-сплайн функция по Кокс Де Бор (Cox de Boor) [11].

Дефиниция 1.2 Нека $T = \{t_0, \dots, t_m\}$ е множество от реални числа, за което е в сила $t_i \leq t_{i+1}, i = 0, \dots, m-1$. Числата t_i се наричат *възли*, а множеството T - *възлов вектор*. Ако са известни възлите

$$t_0 < \dots < t_p < t_{p+1} < \dots < t_{n-1} < t_n < \dots < t_{n+p+1}, \text{ тогава}$$

В-сплайн функция от p -та степен (или ред $p+1$) се нарича функцията

$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \quad (1.10)$$

като
$$N_{i,0}(t) = \begin{cases} 1 & t_i \leq t < t_{i+1} \\ 0 & t \notin [t_i, t_{i+1}) \end{cases}.$$

В-сплайн функциите имат следните характеристики:

- $N_{i,p}(t) = 0$ за стойностите на t извън интервала $[t_i, t_{i+1})$.
- За $p > 0$, $N_{i,p}(t)$ е линейна комбинация на две В-сплайн функции от степен $p-1$ (следва от уравнение 1.10).
- За да се изчисли множество от В-сплайн функции е необходимо да се специфицират възлов вектор T и степен p .
- Ако в уравнение (1.10) се появи коефициент $\frac{0}{0}$, то той се игнорира, т.е. делението се дефинира да бъде равно на нула.
- $N_{i,p}(t)$ са полиноми, дефинирани по цялата реална права, но само в интервала $[t_0, t_m]$ са от интерес.
- Интервалът $[t_i, t_{i+1})$ се нарича *i -тото възлово пространство*.

В-сплайн базисните функции притежават следните свойства:

- Възловият вектор $T = \left\{ \underset{p+1}{0}, \dots, \underset{p+1}{0}, \underset{p+1}{1}, \dots, \underset{p+1}{1} \right\}$ поражда Безие-Бернщайн базиса:

$$N_{i,p}(t) = B_{i,p}(t) = \binom{p}{i} t^i (1-t)^{p-i}, \quad i = 0, 1, \dots, p-1.$$

- Неотрицателни са, за всяка стойност на i, p, t .

- Нормирани са: $\sum_{i=0}^n N_{i,p}(t) = 1$.

Често, за краткост, се записва $N_{i,p}$ вместо $N_{i,p}(t)$.

Изчислението на функциите от p -та степен поражда следната триъгълна таблица:

$$\begin{array}{ccccccc}
 & & & & & & N_{0,0} \\
 & & & & & & N_{0,1} \\
 & & & & & & N_{0,2} \\
 & & & & & & N_{0,3} \\
 & & & & & & \ddots \\
 & & & & & & N_{1,2} \\
 & & & & & & N_{1,3} \\
 & & & & & & N_{1,p} \\
 & & & & & & \vdots \\
 & & & & & & N_{2,2} \\
 & & & & & & \vdots \\
 & & & & & & N_{3,1} \\
 & & & & & & \vdots \\
 & & & & & & N_{4,0} \\
 & & & & & & \vdots
 \end{array}$$

Пример 1.2 Нека $T = \{t_0 = t_1 = t_2 = 0, t_3 = t_4 = t_5 = 1\}$ и $p = 2$. За да се изчислят В-сплайн базисните функции от втора степен, трябва да се изчислят и от нулева и първа степен. Тогава:

$$\begin{aligned}
 N_{0,0} &= N_{1,0} = 0, \quad -\infty < t < \infty \\
 N_{2,0} &= \begin{cases} 1, & t_2 = 0 \leq t < t_3 = 1 \\ 0, & t \notin [0,1) \end{cases} \\
 N_{3,0} &= N_{4,0} = 0, \quad -\infty < t < \infty \\
 N_{0,1} &= \frac{t-0}{0-0} N_{0,0} + \frac{0-t}{0-0} N_{1,0} = 0, \quad -\infty < t < \infty \\
 N_{1,1} &= \frac{t-0}{0-0} N_{1,0} + \frac{1-t}{1-0} N_{2,0} = \begin{cases} 1-t, & 0 \leq t < 1 \\ 0, & t \notin [0,1) \end{cases} \\
 N_{2,1} &= \frac{t-0}{1-0} N_{2,0} + \frac{1-t}{1-1} N_{3,0} = \begin{cases} t, & 0 \leq t < 1 \\ 0, & t \notin [0,1) \end{cases}
 \end{aligned}$$

$$\begin{aligned}
N_{3,1} &= \frac{t-1}{1-1} N_{3,0} + \frac{1-t}{1-1} N_{4,0} = 0, \quad -\infty < t < \infty \\
N_{0,2} &= \frac{t-0}{0-0} N_{0,1} + \frac{1-t}{1-0} N_{1,1} = \begin{cases} (1-t)^2, & 0 \leq t < 1 \\ 0, & t \notin [0,1] \end{cases} \\
N_{1,2} &= \frac{t-0}{1-0} N_{1,1} + \frac{1-t}{1-0} N_{2,1} = \begin{cases} 2t(1-t), & 0 \leq t < 1 \\ 0, & t \notin [0,1] \end{cases} \\
N_{2,2} &= \frac{t-0}{1-0} N_{2,1} + \frac{1-t}{1-1} N_{3,1} = \begin{cases} t^2, & 0 \leq t < 1 \\ 0, & t \notin [0,1] \end{cases}
\end{aligned}$$

От примера се вижда, че когато $t \in [0,1]$ функциите $N_{i,2}$, $i = 0,1,2$, са полиномите от втора степен на Бернщайн. По тази причина, представянето на В-сплайн крива с възлов вектор $T = \{0,0,0,1,1,1\}$ е обобщение на представянето на Безие.

Намирането на първа производна на В-сплайн базисна функция $N_{i,p}$ се извършва чрез следното рекурентно уравнение:

$$N'_{i,p}(t) = N_{i,p}^{(1)} = \frac{p}{t_{i+p} - t_i} N_{i,p-1} - \frac{p}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}. \quad (1.11)$$

Ако с $N_{i,p}^{(k)}(t)$ е обозначена k -тата производна на $N_{i,p}$, тогава тя се изчислява по основната рекурентна формула (1.12) (по Лес Пигъл (Les Piegl) [91]):

$$N_{i,p}^{(k)} = p \left(\frac{N_{i,p-1}^{(k-1)}}{t_{i+p} - t_i} - \frac{N_{i+1,p-1}^{(k-1)}}{t_{i+p+1} - t_{i+1}} \right). \quad (1.12)$$

Производната на $N_{i,p}$ също може да се изрази и чрез функциите $N_{i,p-k}, \dots, N_{i+k,p-k}$ [91]. Рекурентно-итерационна формула за намиране на $N_{i,p}^{(k)}$ е посочена в [97].

1.2.3.3. В-сплайн криви

При изследването на В-сплайн кривите се позоваваме главно на Кокс Де Бор (Cox-de Boor) [11], Бабишков Д. Л. [2], Кокс М. Г. (Cox M. G.) [49], както и на литературни източници съдържащи по-обобщена информация: [6, 7, 22, 23, 26, 43].

Дефиниция 1.3 При известни възли

$t_0 < \dots < t_p < t_{p+1} < \dots < t_{n-1} < t_n < \dots < t_{n+p+1}$ и радиус вектори на $n+1$ характеристични точки M_i , $i=0, \dots, n$, параметричното уравнение на В-сплайн крива c от p -та степен се задава с уравнението (1.13):

$$c: \vec{r} = \gamma(t) = \sum_{i=0}^n M_i N_{i,p}(t), \quad t \in [t_p, t_{n+1}]. \quad (1.13)$$

Степента p , броя на контролните върхове $n+1$ и броя на възлите $m+1$, са свързани със следното отношение:

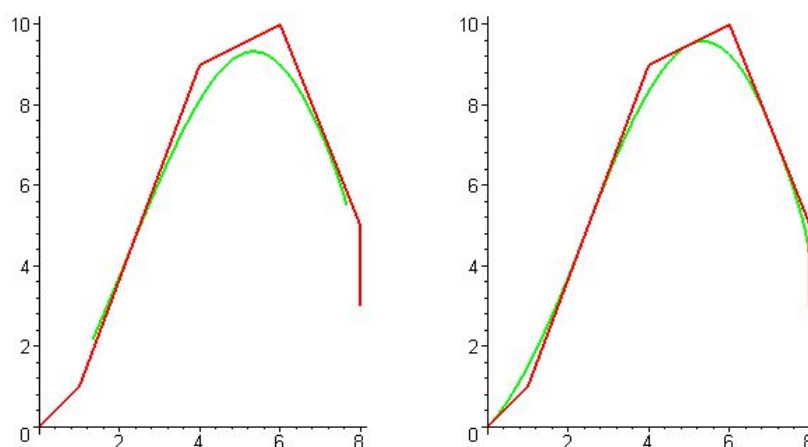
$$m = n + p + 1.$$

1.2.3.4. Параметризация на В-сплайн криви

Съществуват няколко вида В-сплайн криви [17], които се различават по начина на задаване на параметризацията. Тя се определя, както от границите, в които се мени параметърът, така и от стойностите на неговите възли. Възловите вектори биват три вида и във връзка с това сплайните, които се получават, при апроксимацията, могат да се класифицират като: *равномерни периодични*, *равномерни неперидични* и *неравномерни*.

Равномерното периодично множество от възли (например: $\{0,1,2,3,4,5\}$) поражда периодична равномерна нерационална В-сплайн крива (Periodic Uniform Non-Rational B-Spline – PUNRBS). PUNRBS кривите са неудобни поради това, че не започват и не завършват в крайните си характеристични точки.

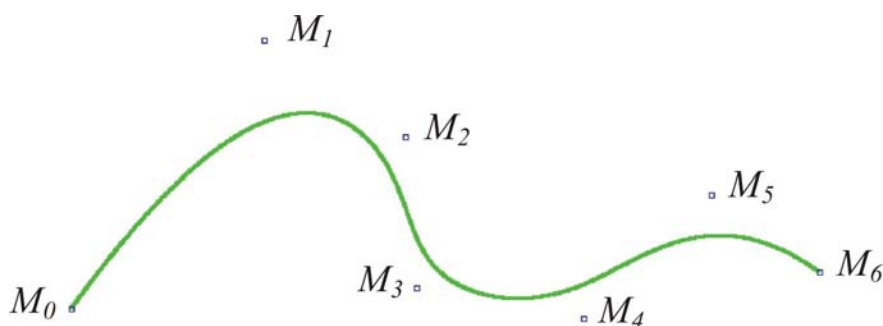
Когато се изберат първите $p+1$ и/или последните $p+1$ възела да са равни ($t_0=t_1=\dots=t_p$ и/или $t_{n+1}=t_{n+2}=\dots=t_{n+p+1}$), то кривата започва от първата и/или свършва в последната характеристична точка. Тогава В-сплайн кривата се нарича *равномерна неперидична* или NPUNRBS (Non-Periodic Uniform Non-Rational B-Spline). Кривата минава през първата и последната точка, а допирателните ѝ в двата ѝ края имат посоките на крайните отсечки от характеристичната начупена.



Фиг. 1.8 а) PUNRBS крива б) NPUNRBS крива.

На фигура 1.8 са показани а) PUNRBS крива и б) NPUNRBS крива построени за контролните върхове (0,0), (1,1), (4,9), (6,10), (8,5), (8,3).

Когато върховете на характеристичната начупена са разположени неравномерно по-добре е да се използва *неравномерна параметризация* (например: {0,0,0,1,4,7,7,7}). В-сплайн кривата се нарича *неравномерна нерационална В-сплайн крива* или NUNRBS (Non-Uniform Non-Rational B-Spline) (фиг. 1.9).



Фиг. 1.9 Равнинна NUNRBS крива определена от седем характеристични точки.⁴

1.2.3.5. Рационални В-сплайн криви

Една рационална В-сплайн крива c се задава в нехомогенни координати чрез векторното уравнение:

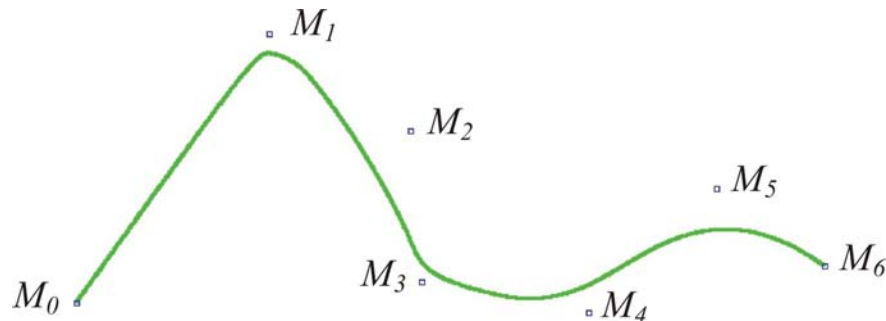
$$c: \vec{r} = \gamma(t) = \frac{\sum_{i=0}^n M_i w_i N_{i,p}(t)}{\sum_{j=0}^n w_j N_{j,p}(t)} \quad (1.14)$$

при известни $n+1$ характеристични точки M_i с тегла w_i , $i=0, \dots, n$ и $n+p+2$ възела t_j , $j=0, \dots, n+p+1$.

Предимство при използването на рационални криви е, че те включват всички частично-полиномиални криви. В частност, рационалните криви от втора степен интерполират точно конични сечения, т.е. с рационални криви е възможно точното представяне на окръжности, елипси, хиперболи и параболи. За целта при задаването на една рационална крива трябва да се определи не само положението на характеристичните ѝ точки, но и техните тегла.

⁴ Изображение генерирано от програмата Draw, която е разработена за целите на това изследване в средата Visual C++ 6.0.

Най-често се използват рационалните В-сплайн криви върху (периодично или неперидично) неравномерно множество от възли или NURBS (Non-Uniform Rational B-Spline) (фиг. 1.10).



Фиг. 1.10 Равнинна неравномерна рационална В-сплайн (NURBS) крива от трета степен определена от седем характеристични точки M_i с тегла $w_i = 1$, $i = 0, 2, \dots, 6$ и $w_1 = 3$.

1.2.4. NURBS криви

Тук ще бъде дадена дефиниция на NURBS крива и свойствата, които тези криви притежават. Основните литературни източници на които се базираме са [2, 11, 17, 48, 80, 91, 92, 97].

Дефиниция 1.4 NURBS крива c от степен p се дефинира с векторното уравнение:

$$c: \vec{r} = \gamma(t) = \frac{\sum_{i=0}^n N_{i,p}(t) w_i P_i}{\sum_{j=0}^n N_{j,p}(t) w_j}, \quad a \leq t \leq b, \quad (1.15)$$

където $\{P_i\}$ са радиус-вектори на характеристичните точки (оформящи *характеристична начупена линия* и *характеристичен полигон*), $\{w_i\}$ са *теглата*, и $\{N_{i,p}(t)\}$ са В-сплайн базисните функции от степен p (ред $p+1$), дефинирани върху

неравномерен възлов вектор $T = \left\{ \underbrace{a, \dots, a}_{p+1}, t_{p+1}, \dots, t_{m-p-1}, \underbrace{b, \dots, b}_{p+1} \right\}$.

Обикновено $a = 0$, $b = 1$, и $w_i > 0$ за всички i . Ако се положи

$$R_{i,p}(t) = \frac{N_{i,p}(t)w_i}{\sum_{j=0}^n N_{j,p}(t)w_j} \quad (1.16)$$

това дава възможност уравнение (1.15) да се запише така

$$c: \vec{r} = \gamma(t) = \sum_{i=0}^n R_{i,p}(t)P_i, \quad (1.17)$$

където $\{R_{i,p}(t)\}$ са рационалните базисни функции.

$R_{i,p}(t)$ притежават следните свойства произлизащи от (1.16) :

- Неотрицателност: $R_{i,p}(t) \geq 0$ за всички $i, p, t \in [0,1]$.
- Нормираност: $\sum_{i=0}^n R_{i,p}(t) = 1$ за всяко $t \in [0,1]$.
- $R_{0,p}(0) = R_{n,p}(1) = 1$.
- За $p > 0$, всички $R_{i,p}(t)$ достигат един максимум в интервала $[0,1]$.
- Локална поддръжка: $R_{i,p}(t) = 0$ за $t \notin [t_i, t_{i+p+1})$. Във всяко дадено възлово пространство, най-много $p+1$ от $R_{i,p}(t)$ са ненулеви (такива са $R_{i-p,p}(t), \dots, R_{i,p}(t)$ в $[t_i, t_{i+1})$).
- Всички производни на $R_{i,p}(t)$ съществуват във вътрешността на възлово пространство и са рационални функции с ненулев знаменател. В един възел, $R_{i,p}(t)$ е $p-k$ пъти непрекъснато диференцируема, където k е периодичността на възела.
- Ако $w_i = 1$ за всички i , тогава $R_{i,p}(t) = N_{i,p}(t)$ за всички i - $N_{i,p}(t)$ са специални случаи на $R_{i,p}(t)$. По-точно, ако всички теглови коефициенти $w_i = a \neq 0$, то тогава $R_{i,p}(t) = N_{i,p}(t)$ за всяко i .

Тези свойства водят до следните геометрични характеристики на NURBS кривите:

- $\gamma(0) = P_0$ и $\gamma(1) = P_n$.
- Афинна инвариантност: афинен образ на NURBS крива е NURBS крива през афинните образи на характеристичните ѝ точки. В частност, NURBS кривите са инвариантни по отношение на перспективната проекция.

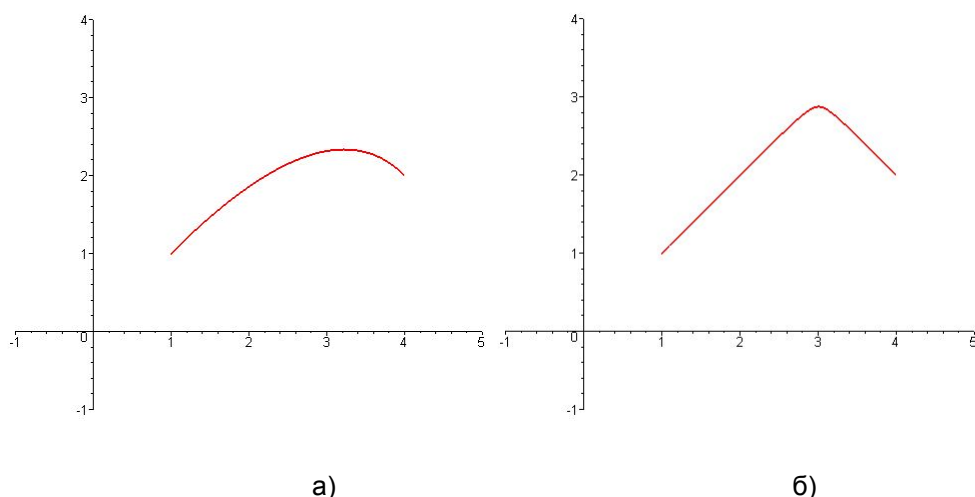
- Строго изпъкнала обвивка: ако $t \in [t_i, t_{i+1})$, тогава $\gamma(t)$ лежи в изпъкналата обвивка на характеристичните точки P_{i-p}, \dots, P_i .
- $\gamma(t)$ е безкрайно диференцируема във вътрешността на възлово пространство и p - k пъти диференцируема във възел с период k .
- Не съществува права, която да пресича повече пъти кривата отколкото правите на характеристичния ѝ полигон (в 2D). Не съществува равнина, която да пресича повече пъти кривата отколкото равнините на характеристичния ѝ полигон (в 3D).
- NURBS крива без вътрешни възли е рационална крива на Безие.
- Локална апроксимация: ако характеристична точка P_i е преместена или стойността на теглото w_i е променена, това оказва ефект само върху тази част на кривата в интервала $t \in [t_i, t_{i+p+1})$.

Алгоритъм за построяване на NURBS крива може да се види в [91], а техники за модификация на формата ѝ в [45, 91, 98, 99].

Пример 1.3 (продължение на Пример 1.2) За уравнението на равнинна NURBS крива c от втора степен определена от характеристичните точки $P_0(1,1)$, $P_1(3,3)$, $P_2(4,2)$ и от възловия вектор $T = \{0,0,0,1,1,1\}$ получаваме:

$$c: \vec{r} = \gamma(t) = \begin{cases} x(t) = \frac{(1-t)^2 \cdot w_0}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 1 + \frac{2t(1-t) \cdot w_1}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 3 + \\ \quad + \frac{t^2 \cdot w_2}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 4 \\ y(t) = \frac{(1-t)^2 \cdot w_0}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 1 + \frac{2t(1-t) \cdot w_1}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 3 + \\ \quad + \frac{t^2 \cdot w_2}{(1-t)^2 \cdot w_0 + 2t(1-t) \cdot w_1 + t^2 \cdot w_2} \cdot 2 \end{cases}$$

За $w_0 = w_1 = w_2 = 1$ получаваме кривата от фигура 1.11а. За $w_0 = w_2 = 1$ и $w_1 = 10$ получаваме кривата от фигура 1.11б.



Фиг. 1.11 Равнинна NURBS крива от втора степен определена с три характеристични точки и построена за различни стойности (а) $w_1 = 1$, б) $w_1 = 10$) на теглото на средната характеристична точка.

1.3. Аналитично представяне на повърхнини в геометричното моделиране

Както кривите така и повърхнините могат да бъдат задавани с *общо уравнение* и чрез *параметрични уравнения* във *векторна* или в *скаларна форма* [13, 27, 33, 34].

Общото уравнение на повърхнина от втора степен е:

$$S: a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{23}yz + 2a_{13}xz + 2b_1x + 2b_2y + 2b_3z + c = 0, \\ a_{ij}, b_i, c \in \mathbb{R}. \quad (1.18)$$

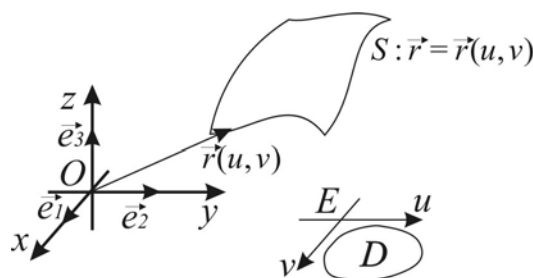
Една повърхнина S , зададена с общото си уравнение $S: F(x, y, z) = 0$, локално около т. $P(x_0, y_0, z_0)$ може да се запише с явно уравнение $S: z = f(x, y)$, ако (x_0, y_0, z_0) е решение на уравнението $F(x, y, z) = 0$ и $F'_z(x_0, y_0, z_0) \neq 0$ (теорема за съществуване и единственост на неявна функция).

Нека в \mathbb{R}^3 е фиксирана дясна декартова координатна система $K = \{O, \vec{e}_1, \vec{e}_2, \vec{e}_3\}$. Нека е дадена гладката векторна функция на два скаларни аргумента $\vec{r}: D \rightarrow \mathbb{R}^3$ и $(u, v) \rightarrow \vec{r}(u, v) = (x(u, v), y(u, v), z(u, v))$, дефинирана за $(u, v) \in D$, където D е област в равнината \mathbb{R}^2 с фиксирана декартова координатна система $k = Euv$. Чрез $\vec{r}(u, v)$ се задава с векторно-параметричното си уравнение повърхнина $S: \vec{r} = \vec{r}(u, v), (u, v) \in D$ (фиг. 1.12), като u и v се наричат нейни параметри. При това повърхнината S е гладка, ако във всяка нейна точка векторите $\vec{r}'_u = (x'_u, y'_u, z'_u)$ и $\vec{r}'_v = (x'_v, y'_v, z'_v)$ са линейно

незавасими, т.е. съществува допирателна равнина, перпендикулярна на вектора $\vec{N} = \vec{r}_u \times \vec{r}_v$ (нормален вектор за S). Често, S се задава и чрез скаларно-параметричните си уравнения:

$$S: \begin{cases} x = x(u, v) \\ y = y(u, v) \\ z = z(u, v) \end{cases} \quad (u, v) \in D. \quad (1.19)$$

С точност до репараметризация на S , когато такава съществува, D се счита за единичния квадрат от параметричната равнинина.



Фиг. 1.12 Параметрично задаване на равнина в \mathbb{R}^3 .

1.3.1. Модели на повърхнини

Параметричното представяне на повърхнини е много популярен и мощен подход за представяне на криволинейни обекти в 3D графичните приложения. От параметричното задаване на повърхнина става ясно, че за нея са нужни два параметъра, които се менят в указана равнинна област D . Областта D може да се покрие с достатъчно гъста координатна мрежа. За всеки възел от това покритие, може да се използва съчетание от две функции, като всяка от тях е дефинирана в една от посоките на параметричното пространство. Така повърхнината S може да се представи чрез оператор $F_{u,v}$ приложен върху множеството от входни данни $\{P(u, v)\}$:

$$S: \vec{r} = S(u, v) = F_{u,v}.P(u, v).$$

Съществуват три вида представяния [2] според вида на апроксимация на оператора $F_{u,v}$, но най-популярно в тримерните графични системи е представянето чрез произведение на два оператора F_u и F_v :

$$F_{u,v} = F_u.F_v$$

$$S: \vec{r} = S(u, v) = F_u.F_v.P(u, v).$$

От този вид са представени всички модели на повърхнини, включени в това изследване.

Съществуват основно две техники за представяне на повърхнини в компютърната графика [97, 379-380 с.]. Първата, най-често асоциирана с името на Кунс (Coons) [97, 422-434 с.], създава математическа повърхнина от предварително зададени данни. Неудобство в този случай е, че тези повърхнини не могат интуитивно и интерактивно да се модифицират по отношение на формата им. Втората, най-често свързвана с името на Безие [97, 435-445 с.], създава математическа повърхнина, която свободно може да бъде моделирана. Тъй като това изследване е в сферата на ГМ, обзора на моделите на повърхнини ще бъде насочен към втората техника.

1.3.2. Повърхнини в Безие-Бернщайн базиса

Повърхнините в Безие-Бернщайн базиса се представят [51, 97] като произведение на две функции приложени върху множество от входни данни $\{P_{i,j}\}$:

$$S: \vec{r} = S(u, v) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) P_{i,j}, \quad u, v \in [0, 1]. \quad (1.20)$$

Функциите $B_{i,p}(u)$ и $B_{j,q}(v)$ са полиноми на Бернщайн съответно от степен p и q . Множеството от входни данни са характеристични точки в пространството $P_{i,j}$ за $i = 0, \dots, p$ и $j = 0, \dots, q$. Тези характеристични точки много често се наричат *управляващи (контролни) точки* или *контролни върхове*⁵ и тъй като в 3D ГСГМ е възприето последното наименование, занапред в работата ще използваме него.

Повърхнините в Безие-Бернщайн базиса много често се наричат *повърхнини на Безие*. Те се дефинират върху квадратна мрежа от контролни върхове. Граничните криви на повърхнината са криви на Безие, дефинирани, посредством контролните върхове по ръбовете. Повърхнината между ръбовете е оформена пропорционално от близките контролни върхове. Използването на тези повърхнини има много предимства. Те са балансирани между простота и сложност, предоставяйки свобода с минимално количество от сложност при програмиране. Повърхнината е лесна за модификация на формата чрез промяна на позициите на контролните ѝ върхове в пространството.

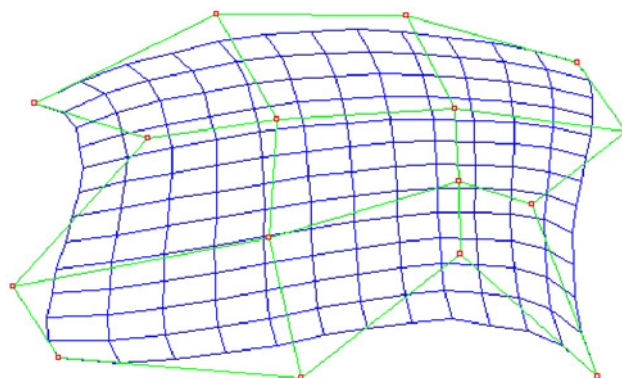
В ГМ най-често се използват повърхнините на Безие от 2-ра и 3-та степен дадени съответно с уравнения (1.21) и (1.22).

$$S: \vec{r} = S(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 B_{i,2}(u) B_{j,2}(v) P_{i,j}, \quad u, v \in [0, 1]. \quad (1.21)$$

⁵ Превод от английски език на control vertex (CV).

$$S: \vec{r} = S(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_{i,3}(u) B_{j,3}(v) P_{i,j}, \quad u, v \in [0, 1]. \quad (1.22)$$

При реализация в графична програма се избират стойности за u и v в интервала $[0, 1]$. Точките, които се получават от равнината се свързват с прави линии и така се реализира геометричната апроксимация (фиг. 1.13).

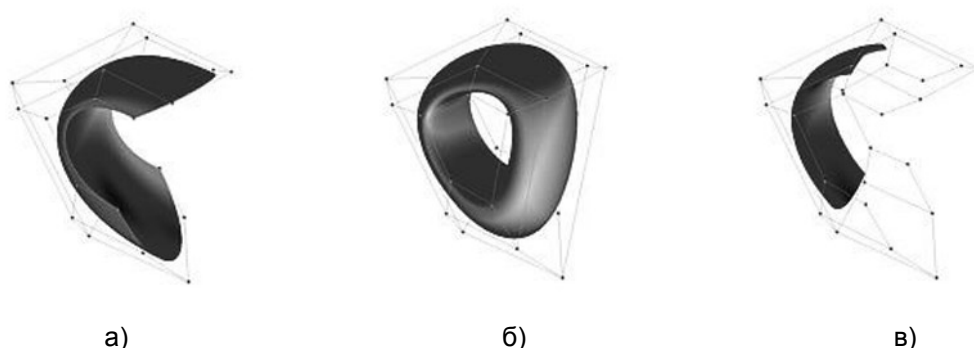


Фиг. 1.13 Повърхнина на Безие от трета степен.⁶

1.3.3. В-сплайн повърхнини

При зададени множество P от $m.n$ на брой контролни върхове $P_{i,j}$, $i=0, \dots, m-1$ и $j=0, \dots, n-1$, възлов вектор $U=\{u_0, u_1, \dots, u_{h-1}\}$, възлов вектор $V=\{v_0, v_1, \dots, v_{k-1}\}$, В-сплайн базисните функции $N_{i,p}(u)$ и $N_{j,q}(v)$ от степен съответно p (ред $p+1$) и q (ред $q+1$), като $h=m+p+1$ и $k=n+q+1$, В-сплайн повърхнината се дефинира по следния начин [2, 11, 44, 53, 107]:

$$S: \vec{r} = S(u, v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} N_{i,p}(u) N_{j,q}(v) P_{i,j}. \quad (1.23)$$



Фиг. 1.14 В-сплайн повърхнини – а) скачена, б) затворена и в) отворена.

⁶ Изображение генерирано от програмата Draw3D, която е написана за целите на това изследване в средата Visual C++ 6.0.

Коефициентът на контролен връх P_{ij} е произведение на двете В-сплайн базисни функции $N_{i,p}(u)$ и $N_{j,q}(v)$ като $N_{i,p}(u)$ може да се фиксира, докато $N_{j,q}(v)$ се променя. В [11] е посочен алгоритъмът на Де Бор (de Boor) за построяване на повърхнината. В-сплайн повърхнините могат да бъдат скачени, затворени и отворени [118] по всяка от посоките на параметрите u и v в зависимост от избора на параметризацията (фиг. 1.14).

1.3.4. NURBS повърхнини

Тук ще бъде дадена дефиниция на NURBS повърхнина и свойствата, които тези повърхнини притежават. Основните литературни източници на които се базираме са [2, 11, 17, 91, 92, 97].

Дефиниция 1.5 NURBS повърхнина S от степен p в u направлението и степен q във v направлението се дефинира с уравнението:

$$S: \vec{r} = S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{k,l}} \cdot P_{i,j}, \quad 0 \leq u, v \leq 1, \quad (1.24)$$

където $\{P_{i,j}\}$ са радиус-векторите на двупосочна *характеристична мрежа*, състояща се от характеристичните точки $P_{i,j}$, наричани още *контролни върхове*, $\{w_{i,j}\}$ са *теглата* на тези контролни върхове, а $\{N_{i,p}(u)\}, \{N_{j,q}(v)\}$ са нерационалните В-сплайн базисни функции дефинирани върху възловите вектори

$$U = \left\{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\} \text{ и } V = \left\{ \underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\},$$

където $r = n + p + 1$ и $s = m + q + 1$.

Ако се дефинира

$$R_{i,p,j,q}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u) N_{l,q}(v) w_{k,l}}, \quad (1.25)$$

тогава $\{R_{i,j}(u, v)\}$ се явяват рационални базисни функции и повърхнината може да бъде дадена и с уравнението:

$$S: \vec{r} = S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,p,j,q}(u, v) P_{i,j}. \quad (1.26)$$

Функциите $\{R_{i,p,j,q}(u, v)\}$ притежават следните свойства:

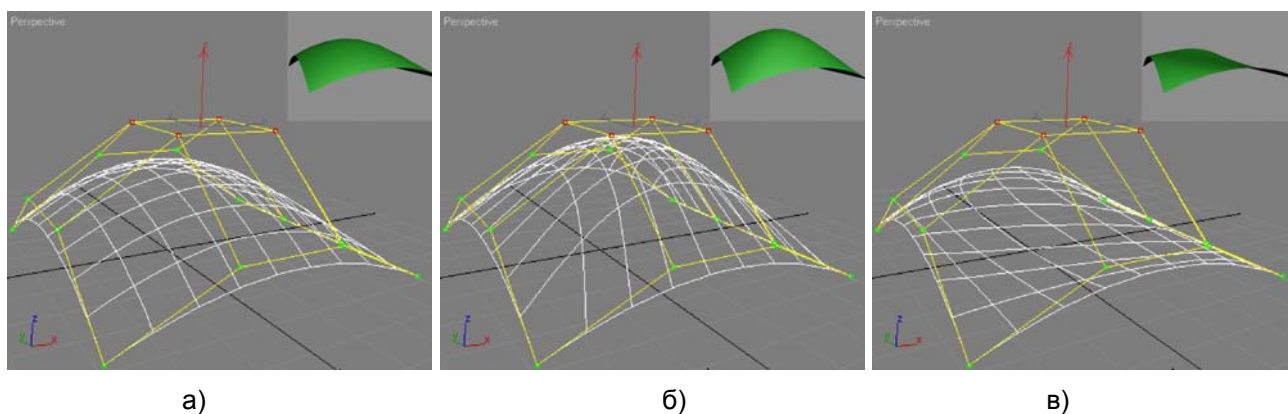
- Неотрицателност: $R_{i,p,j,q}(u, v) \geq 0$ за всяко i, j, u и v ;
- Нормираност: $\sum_{i=0}^n \sum_{j=0}^m R_{i,p,j,q}(u, v) = 1$ за всички $(u, v) \in [0, 1] \times [0, 1]$.
- Локална поддръжка: $R_{i,p,j,q}(u, v) = 0$ ако (u, v) е извън правоъгълника даден чрез $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.
- Във всеки даден правоъгълник във формата $[u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, най-много $(p+1)(q+1)$ базисни функции са ненулеви, по-точно $R_{i,p,j,q}(u, v)$ за $i_0 - p \leq i \leq i_0$ и $j_0 - q \leq j \leq j_0$ са ненулеви.
- Екстремум: ако $p > 0$ и $q > 0$, тогава $R_{i,j}(u, v)$ достига точно една максимална стойност.
- $R_{0,p,0,q}(0, 0) = R_{n,p,0,q}(1, 0) = R_{0,p,m,q}(0, 1) = R_{n,p,m,q}(1, 1) = 1$.
- Диференцируемост: вътре в правоъгълниците формирани чрез u - и v -линиите, всички частни производни на функцията $R_{i,p,j,q}(u, v)$ съществуват. Във възел u (възел v) тя е $p - k$ ($q - k$) пъти диференцируема в u (v) направлението, където k е периодичността на възела.
- Ако всички $w_{i,j} = a$ за $0 \leq i \leq n, 0 \leq j \leq m, a \neq 0$, тогава $R_{i,p,j,q}(u, v) = N_{i,p}(u)N_{j,q}(v)$ за всички i, j .

Тези свойства водят до следните геометрични характеристики на NURBS повърхнините:

- $S(0, 0) = P_{0,0}, S(1, 0) = P_{n,0}, S(0, 1) = P_{0,m}, S(1, 1) = P_{n,m}$.
- Афинна инвариантност: образът при афинна трансформация на NURBS повърхнина е NURBS повърхнината по афинните образи на характеристичните точки.

- Строго изпъкнала обвивка: ако $w_{i,j} \geq 0$ за всяко i, j и $(u, v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, тогава точките от $S(u, v)$ са в изпъкналата обвивка на контролните върхове $P_{i,j}$, $i_0 - p \leq i \leq i_0$ и $j_0 - q \leq j \leq j_0$.
- Локална модификация: ако $P_{i,j}$ е преместена или стойността $w_{i,j}$ е променена, това оказва ефект върху формата на повърхнината само в правоъгълника $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.
- Нерационални B -сплайн повърхнини, нерационални повърхнини на Безие, и рационални повърхнини на Безие са специални случаи на NURBS повърхнини.
- Диференцируемост: векторната функция, задаваща повърхнината S е $p - k$ ($q - k$) пъти диференцируема по отношение на u (v) във възел u (v) с периодичност k .

Алгоритъм за построяване на NURBS повърхнина може да се види в [91], а техники за модификация на формата ѝ в [99]. На фигура 1.15 е показана бикубична NURBS повърхнина определена от шестнадесет контролни върхове $P_{i,j}$, $i, j = 0, \dots, 3$ с тегла $w_{i,j} = 1$ на изображение а). На изображение б) теглата $w_{1,1}$, $w_{1,2}$, $w_{2,1}$ и $w_{2,2}$ (на контролните върхове с червен цвят) са със стойност 5, а на изображение в) със стойност 0.



Фиг. 1.15 Бикубична NURBS повърхнина.⁷

⁷ Изображенията са получени чрез програмата 3ds Max.

1.3.5. Методи за построяване на повърхнини

Съществуват няколко основни метода за построяване (генериране) на повърхнини в ГМ, което най-често се извършва на базата на една или няколко (криви) линии. Ще бъдат дадени тези методи, които са в тясна връзка с това изследване.

1.3.5.1. Ротационни повърхнини

Един метод за построяване на тримерна повърхнина е да се завърти двумерен обект, например права или крива, около ос в пространството [97]. Такава повърхнина се нарича *ротационна повърхнина (revolved surface)*. Обект от този клас повърхнини може да бъде завъртането на права, колинеарна на оста Ox и отстояща от нея на разстояние d , на 360° около Ox . Получава се повърхнината на ротационен цилиндър (кръгов цилиндър).

1.3.5.2. Повърхнини на замитане

Тримерна повърхнина може да бъде получена чрез *замитане (прокарване)* на даден графичен обект (с определена стъпка) по *път* (права, крива или начупена линия), който обикновено се поставя ортогонално на пътя [97]. Резултатът е повърхнина, която се нарича *повърхнина на замитане (swept surface)*.

1.3.5.3. Повърхнини от парчета

За предходните видове повърхнини съществуват аналитични описания чрез една векторна функция, но има и много повърхнини, за които такова аналитично представяне не съществува. Типични примери са автомобилните, корабните и самолетните корпуси, скулптури, бутилки, обувки и др.

Тук идеята е, отделни *повърхнини-парчета (patches)* да бъдат съединени заедно по техните контури в една завършена повърхнина [97]. Най-често се построяват парчета като части от реалните неразпадащи се повърхнини от втора степен зададени със скаларно-параметричните си уравнения. Задачата е свързана с намиране на диапазоните на вариране на параметрите на векторните функции и след това определяне и уеднаквяване на нормалите на двете повърхнини, които се съединяват в краищата си.

1.3.5.4. Праволинейни повърхнини

Една праволинейна повърхнина се генерира чрез права, която се движи по път (права или крива). В точка от повърхнина се взема равнина, съдържаща нормалата на повърхнината в тази точка, извършва се ротация около тази нормала и ако в поне една

посока всяка точка от права на равнината допира повърхнината, повърхнината е *праволинейна (ruled)* в тази посока [97].

1.4. Методи за представяне на квадратични криви и повърхнини в геометричното моделиране

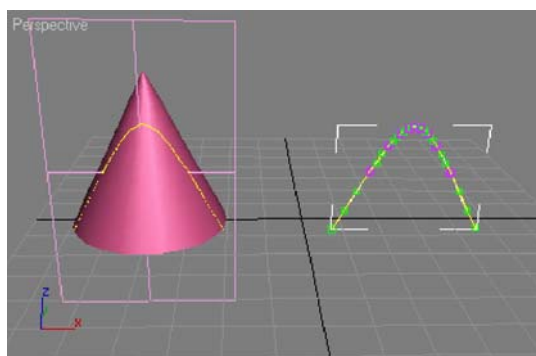
Изследването по-нататък ще е насочено към това - квадратичните криви и повърхнини, които се представят с по едно общо уравнение (за кривите с уравнение (1.1), за повърхнините с уравнение (1.18)), да присъстват с подходящи модели за моделиране в 3D графичните приложения. Много често траектории на движение, граници на реални обекти, форми и др. са квадратични криви или повърхнини [70, 73, 79, 81, 87, 113, 114, 115, 121, 140, 141]. Оптичните свойства както на кривите от втора степен така и на повърхнините от втора степен също са добре познати [33, 50, 52, 59, 139]. Затова тук ще бъде даден резултата от обзора, който беше направен на съществуващите техники за построяване на тези форми. Основните характеристики, които се следят са: брой на стъпките при построяване на кривата или повърхнината; време за построяване; възможност за локална модификация на получената крива или повърхнина; брой на контролните върхове (при NURBS повърхнина); визуализация.

Идеята за представяне на тези криви и повърхнини в графичните приложения съществува [63, 90, 93, 94, 97]. Един от първите методи за компютърно представяне на конично сечение е предложението от Лайминг (Liming) [17]. Проузър (Prosser) в [94] прави линейна апроксимация на кониките с 16-битова целочислена аритметика. Роджерс (David Rogers) в [97, 400-408 с.] анализира два метода за представяне на квадратични повърхнини, алгебричен и геометричен. Голдман (Goldman) в [67] търси пресечната крива на две повърхнини и когато последната е конично сечение, се работи на базата на геометричните дефиниции и алгебричното му представяне. Тези методи за представяне на коничните сечения обаче, не са подходящи за интерактивния дизайн или за автоматично сглобяване на криви. Целта е да бъдат намерени по-добри модели и методи, които решават успешно задачите стоящи пред ГМ.

Тео Павлидис [90] дава дефиниция за коничен сплайн. Той формира коничните сплайни чрез дъги на коники, всяка една дефинирана посредством петте си елемента: крайните си точки, допирателните в тях плюс средна точка. Освен това посочва репараметризация като използва *форма-фактор* на коничен сплайн-сегмент и класифицира върховете на характеристичния полигон. Такива сплайн криви се използват успешно както за интерактивен дизайн, така и за автоматично сглобяване на криви [72, 93].

Воуган (Vaughan) в [102] разширява изследването на Павлидис като разглежда коничните сечения като криви получени при пресичането на кръгов конус с подходяща равнина и намира представянето им чрез сплайн крива.

Обикновено графичните приложения притежават като примитив за моделиране окръжност, която е афинно-еквивалентна на елипса. Чрез подходяща афинна трансформация на окръжността се получава елипса с предварително зададени полуоси. За получаването на парабола и хипербола обаче, основната техника за построяването им е намирането на сплайн крива от пресичането на кръгов конус с подходяща равнина (фиг. 1.16). По този метод, обаче, е трудно определянето на желаната квадратична крива с точните ѝ параметри. Освен това сплайн кривите притежават някои съществени недостатъци, които бяха споменати в точка 1.2.3.1. Най-доброто решение се явява преобразуването на сплайн кривата до NURBS крива. Но това е твърде сложен метод, а и NURBS кривата е с голям брой контролни върхове (на фиг. 1.16 те са 34).



Фиг. 1.16 Получаване на конично сечение като сплайн крива (в ляво) и преобразуваната сплайн крива до NURBS крива (в дясно).

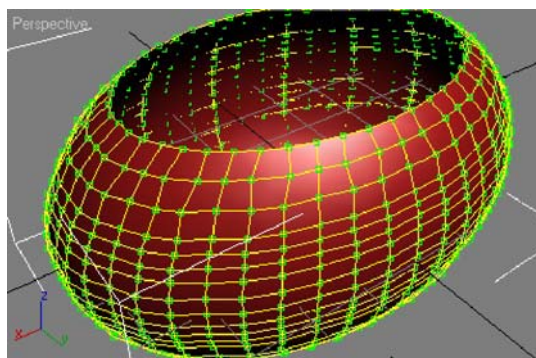
3D графичните системи, които бяха изследвани, притежават инструменти за създаване на следните квадратични повърхнини: сфера, конус и кръгов цилиндър. В повечето приложения, тези обекти се представят параметрично и не могат да бъдат модифицирани, докато в други (например Autodesk Maya), съществуват техни NURBS примитиви. В приложенията, където не съществуват NURBS примитиви, получените повърхнини трябва да се преобразуват (ако системата разполага с такава възможност) до NURBS повърхнина. Този процес, обаче, е “скъп” по отношение на ресурсите на компютърната система.

Построяването на всяка от квадратичните повърхнини в 3D графичните системи е възможно на базата на сплайн управителна крива, която е получена като конично сечение. От сплайн кривата се построява повърхнина, която в повечето случаи е

подчинена от управителната крива (сплайн кривата). Това означава, че редактирането само на части от повърхнината е невъзможно. За да бъде редактирането възможно, повърхнината трябва да се преобразува до пач (patch) или меш (mesh) повърхнина. Но моделирането с пач или меш не е достатъчно прецизно и гъвкаво. На лице е също и не добрия визуален резултат при рендериране на така получените обекти, което е от основно значение в разглежданите 3D графични системи. При преобразуване на получения меш или пач в NURBS повърхнина вече е възможно редактиране на части от повърхнината. Основните недостатъци са:

- Броят на стъпките при построяване на повърхнините е голям: за различните повърхнини той е между 3 и 6.
- Времето за построяване при различните повърхнини е между 1 и 5 минути, а когато се желае отлично качество на визуализация, това време може да надхвърли и един час. Причината за това, главно е процесът на преобразуване на повърхнините до NURBS.
- Броят на контролните върхове на преобразуваната повърхнина до NURBS е от 256 до 4864 (и повече), което е много за интерактивното моделиране.
- Локалната модификация е възможна само при преобразуване на повърхнината до NURBS, но ако контролните ѝ върхове са голям брой модификацията е почти невъзможна.
- Визуалният резултат в голяма част от методите е незадоволителен, а само в няколко частни случая е с отлично качество.

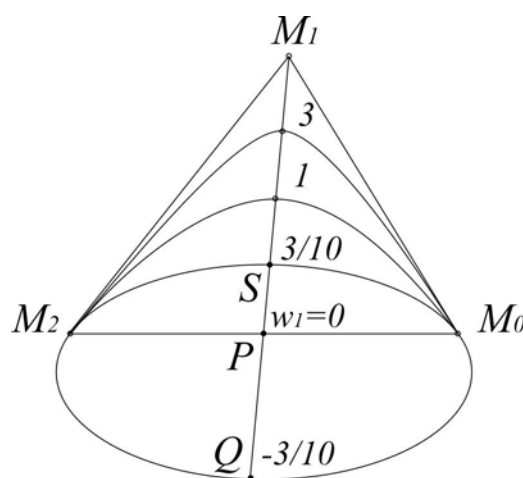
На фигура 1.17 може да се види повърхнината на част от елипсоид, преобразувана до NURBS, която се получава при един от най-добрите методи и с най-добро качество спрямо другите повърхнини.



Фиг. 1.17 NURBS елиптична повърхнина с 1280 контролни върхове.

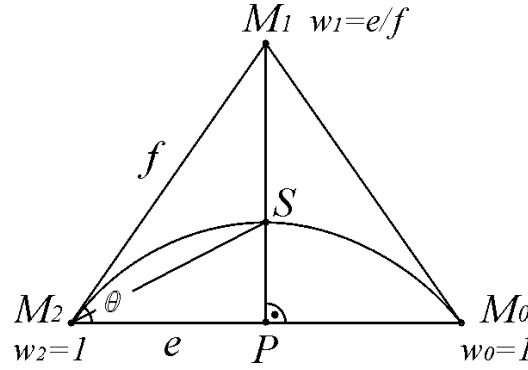
Коничните сечения са реални криви от втора степен. Лес Пигъл (Les Piegl) в [91] посочва Лий (Lee) за доказателството, че части от тях се представят точно с

рационални криви на Безие от втора степен [91]. Тези части могат да бъдат дъги от: окръжност, елипса, хипербола, парабола и отсечка. Уравнението на тази крива е уравнение (1.8) при $n=2$. В това уравнение, параметрите, които оказват влияние на формата на кривата са $w_i, i=0,1,2$. Обикновено се фиксират $w_0 = w_2 = 1$ - *нормална параметризация*. Тогава формата на кривата зависи от стойността на $w_1 \in (-\infty; +\infty)$ и от знаменателя на уравнението. За формата се получава, че ако: $-1.0 < w_1 < 1$ тя е дъга от елипса, $w_1 = -1$ или $w_1 = 1$ тя е дъга от парабола, $w_1 < -1$ или $w_1 > 1$ тя е дъга от хипербола (фиг. 1.18).



Фиг. 1.18 Четири дъги на конични сечения за $w_1 = \{-0.3, 0.3, 1, 3\}$.

Трябва да се отбележи, че w_1 може да бъде и 0, което задава отсечката M_0M_2 (фиг. 1.18). Също така, при $w_1 < 0$ се получава допълващата от другата страна на M_0M_2 дъга ($\widehat{M_0QM_2}$), която може да бъде допълващата част за дъгата $\widehat{M_0SM_2}$ от елипса за $w_1 < 1$. Това, обаче, не е така когато дъгата е част от хипербола или парабола. Получава се несиметрична и изродена крива. По тази причина по-нататък в това изследване ще работим само с положителни стойности за w_1 .



Фиг. 1.19 Представяне на дъга от окръжност по-малка от 180° чрез рационална крива на Безие от втора степен.

Изменението на w_1 води до фамилия от конични дъги и един удобен начин да се избере конична форма е да се зададе и трета точка лежаща на кривата (т. S (фиг. 1.19)), която се получава за стойност на параметъра $t = 0.5$ в уравнение (1.8) за $n=2$. Тази точка се нарича *раменна точка* на кривата. Заместването с $t = 0.5$ в уравнение (1.8) за $n=2$ води до

$$S = \frac{1}{1+w_1}P + \frac{w_1}{1+w_1}M_1, \quad (1.27)$$

където P е средата на M_0M_2 . Ако s се избере да бъде нов параметър, който задава линейна интерполация между P и M_1 , тогава съществува стойност на $s \in (0,1)$ при която

$$S = (1-s)P + sM_1 \Rightarrow s = \frac{w_1}{1+w_1} \Rightarrow w_1 = \frac{s}{1-s}. \quad (1.28)$$

Чрез уравнение (1.8) за $n=2$ е възможно също представянето на дъга от окръжност по-малка от 180° . За да се получи симетрия, $M_0M_1M_2$ (Фиг. 1.13) трябва да бъде равнобедрен триъгълник като $|M_0M_1| = |M_1M_2|$. Ако $\theta = \angle M_1M_2P$, то от симетрията следва, че дъгата $\widehat{M_2S}$ е равна на $\widehat{SM_0}$, следователно M_2S е ъглополовящата за $\angle \theta$. От уравнение (1.28) и свойствата на ъглополовящите се вижда геометричния смисъл на параметъра w_1 :

$$w_1 = \frac{s}{1-s} = \frac{|PS|}{|SM_1|} = \frac{|PM_2|}{|M_1M_2|} = \frac{e}{f} = \cos \theta. \quad (1.29)$$

В [63] Фарин Джералд (Farin Gerald) дава също доказателство на това, че всяка рационална крива на Bezier от втора степен представлява дъга от конично сечение.

Той изказва твърдението, че коничните сечения могат да бъдат представени и чрез NURBS криви.

Лес Пигъл в [91] представя алгоритъм за построяване на дъга от коника чрез крива на Безие от втора степен. Предварително се изчисляват двете крайни точки и техните допирателни, плюс една допълнителна точка от дъгата. След като тези необходими данни бъдат получени, се подават на алгоритъма, който намира средната характеристична точка на кривата и нейното тегло. Това се прави чрез двукратно извикване на функция, която намира пресечна точка на две прави в пространството. Така значително се повишава сложността на алгоритъма, което пък води до увеличаване на времето за изпълнение.

При интерактивното построяване (в реално време) на дъга от коника е необходимо в много кратък период от време да се построят и после премахнат няколко десетки криви и накрая да остане една. Следователно, прилагането на изчисления от вида на посочените в метода на Пигъл ще забавят значително процеса за моделиране.

Освен това, алгоритъмът на Пигъл не позволява чертането на по-голяма дъга (теоритично не може да бъде 180° и повече, а на практика не по-голяма от 175°), на пълна елипса или пък на дъга от елипса, която е по-голяма от 360° . За представянето на пълна елипса той предлага отделно решение, а именно да се търсят центъра на елипсата, малката и голямата полуос, както и осите на локалната координатна система, да се изчисли само една дъга от елипсата и на базата на тези данни елипсата да се допълни. Това обаче повишава изчислителната сложност на алгоритъма.

Пигъл дава и метод, в който използва безкрайни точки, които да участват в характеристичния полигон при определени стойности на входните данни (задаващи 180 градусова дъга от елипса). Но това не е много подходящо за интерактивния дизайн.

Когато бъдат получени характеристичните точки на рационалната крива на Безие, тя може да бъде преобразувана до NURBS крива от втора степен.

1.5. Изводи

От направения обзор в Глава I се налагат следните изводи:

1. Компютърните графични системи за геометрично моделиране се разработват с цел доставяне на набори от софтуерни инструменти предимно за дизайн и визуализация. Развитието на 3D ГСГМ налага моделите на криви и повърхнини в ГМ да се развият в посока на създаване на нови графични примитиви с цел разширяване на функционалността на този вид приложения.

2. Най-широко използваните модели на криви и повърхнини в ГМ притежават важни аналитични свойства и геометрични характеристики, които дават възможност за представянето на изключително голямо разнообразие от форми. В-сплайн базисът отговаря напълно на критериите, поставени в раздел 1.2, за клас от координатни функции нужни при представянето на кривите и повърхнините. NURBS кривите и повърхнините позволяват моделиране, включващо и надминаващо възможностите, които предоставят Безие и сплайн моделите. При необходимост обаче, NURBS кривите и повърхнините могат да бъдат еквивалентно преобразувани до Безие и сплайн криви и повърхнини [98, 108].

3. Обзорът показва, че 3D ГСГМ не притежават инструменти за създаване на NURBS модели на всички криви и повърхнини от втора степен като графични примитиви, а само на окръжност – от кривите и на сфера, конус и кръгов цилиндър – от повърхнините.

4. Присъствието на квадратичните криви и повърхнини в 3D графичните приложения като примитиви за моделиране, би помогнало много по-лесно да бъдат създавани графични модели на обекти, чиито свойства при попадане на светлина, радио вълни, твърди тела и др., са вече известни и изследвани. Поведението на телата след попадането им върху тези повърхнини е предсказуемо или известно. Следователно: 1) при необходимост да се моделират обекти със споменатите свойства на кривите и повърхнините, това ще е възможно и 2) изчислителния процес при симулирането на поведението на обектите, преминаващи или отблъскващи се от тях ще бъде облекчен значително.

5. Моделите на кривите и повърхнините от втора степен трябва да бъдат подходящи за реализиране чрез ефективни алгоритми и създаване на специализирани софтуерни модули-добавки (плъгини) или чрез приложен програмен интерфейс или чрез средствата, предоставени от 3D графичните системи.

6. Представянето на кривите и повърхнините от втора степен е възможно и чрез криви на Безие. Изискването, обаче, кривите и повърхнините да позволяват локална модификация в отделни техни части, налага представянето им чрез съставни криви на Безие. Но моделирането с криви, които се състоят от множество криви, е по-трудно отколкото моделирането само с една крива, а при повърхнините сложността още повече нараства. Направеният обзор показва, че NURBS моделът е изключително подходящ при представянето на криви и повърхнини за моделиране в 3D графичните системи с тази цел (за които се изисква да позволяват локална модификация).

На база на формулираните изводи от направените аналитични изследвания и проучвания за състоянието на ГМ в КГ и моделите на кривите и повърхнините от втора степен за дизайн са формулирани *целта* и *задачите* на дисертационния труд.

Цел: Изследване и развитие на моделите на криви и повърхнини от втора степен в геометричното моделиране, които да служат за разработване на графични инструменти за интерактивен дизайн.

За успешното реализиране на така формулираната цел са поставени следните **задачи:**

1. Да се разработят NURBS модели на криви за интерактивно и в реално време моделиране, чрез които да бъдат представени кривите, които се задават с едно общо уравнение от втора степен: елипса, хипербола и парабола.
2. Да се разработят NURBS модели на повърхнини за интерактивно и в реално време моделиране, чрез които да бъдат представени повърхнините, които се задават с едно общо уравнение от втора степен: елипсоид, елиптичен параболоид, хиперболоид, двоен хиперболоид, конус, хиперболичен параболоид, елиптичен цилиндър, параболичен цилиндър и хиперболичен цилиндър.
3. Да се изследват възможностите на съвременните 3D графични системи за разширяване на графичния им инструментариум и да се разработят и внедрят нови графични инструменти, които работят на базата на предложените NURBS модели на кривите и повърхнините от втора степен, в 3D графична система за моделиране.

Глава втора

ПРЕДСТАВЯНЕ НА КРИВИТЕ ОТ ВТОРА СТЕПЕН ЧРЕЗ NURBS КРИВИ

Разработването на нови графични примитиви за моделиране с кривите от втора степен в 3D графичните системи, налага тяхното представяне чрез подходящ модел на криви в ГМ. В първа глава, като най-подходящ модел след аналитичния обзор, беше посочен NURBS. Това преопределя представянето на кривите да направим чрез разработка на NURBS криви.

Определянето на дъга от съответната крива (елипса, парабола или хипербола) е целесъобразно да направим въз основа на стойностите на параметрите от скаларно-параметричните ѝ уравнения. Затова първо ще бъдат дадени аналитичните задавания на квадратичните криви, а след това и техните представяния в NURBS.

2.1. Аналитично представяне на кривите от втора степен

Коничните сечения са кривите, получени при пресичането на кръгов конус с произволна равнина под специално избран ъгъл спрямо оста на конуса [33]. Следователно трите типа криви, които се получават са равнинни – лежат в равнината на пресичане. Те обикновено се разглеждат в \mathbb{R}^2 с фиксирана дясна декартова координатна система $k = \{O, \vec{e}_1, \vec{e}_2\}$.

Общото уравнение на тези криви (фиг. 2.1) се дава с уравнение (1.1). Едно матрично записване на уравнение (1.1) в нехомогенни координати е

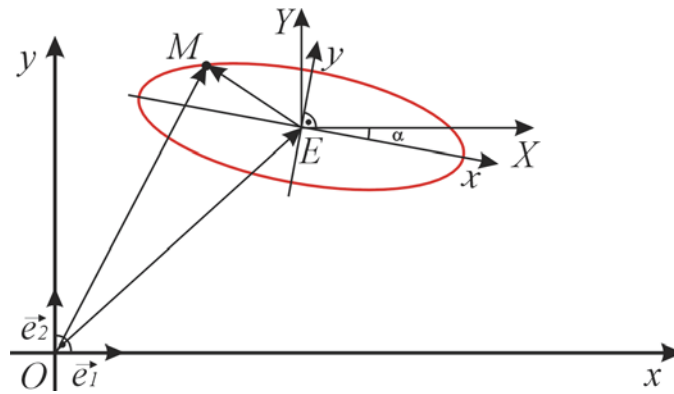
$$C: (x, y) \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + (x, y) \begin{pmatrix} 2b_1 \\ 2b_2 \end{pmatrix} + c = 0 \quad (2.1)$$

и в хомогенни координати е

$$C: (x, y, w) A (x, y, w)^T = 0, \quad (2.2)$$

където $A = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{12} & a_{22} & b_2 \\ b_1 & b_2 & c \end{pmatrix}$ е симетрична 3x3 матрица.

Според броя на общите точки на C с безкрайната права $\omega: w=0$ кривите са елиптични (0 общи точки), параболични (1 обща точка) и хиперболични (2 общи точки).



Фиг. 2.1 Равнинна крива от втора степен.

В [33] е посочен един вариант на метрична класификация за криви от втора степен. Чистото аналитико-геометрично изследване дава по-пълна информация за геометрията на кривите. След извършване на канонизиране на уравнението на кривата се получават добре известните девет типа метрично нееквивалентни криви от втора степен, като за целите на компютърната графика са интересни трите *неразпадащи се* (централни и нецентрални) *реални криви*: *елипса* и *хипербола* (имащи център на симетрия) и *парабола* (без център на симетрия). Това са коничните сечения, които се наричат още коники (фиг. 2.1).

Известна е следната обща геометрична характеристика на тези криви: ако в равнината са дадени точка F наречена *фокус* и права d наречена *директриса*, то геометричното място от всички точки P , за които частното от разстоянията им до F и до

$$d \text{ е } \frac{|\overrightarrow{FP}|}{\delta(P, d)} = e = \text{const} \text{ е точно:}$$

- елипса, при $e < 1$.
- парабола, при $e = 1$;
- хипербола, при $e > 1$.

Числото e се нарича *ексцентрицитет*.

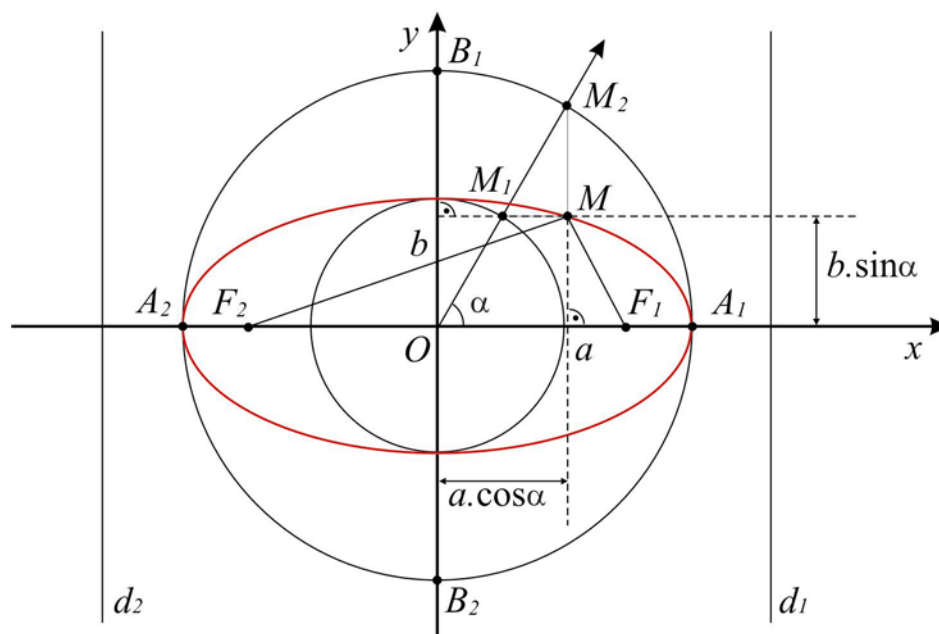
Каноничното уравнение на елипсата е:

$$\varepsilon: \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad a \geq b > 0. \quad (2.3)$$

Скаларно-параметричните уравнения (СПУ) на елипсата ε (фиг. 2.2) се получават от основното тригонометрично тъждество $\sin^2 \alpha + \cos^2 \alpha = 1$ и са:

$$\varepsilon: \begin{cases} x = a \cdot \cos \alpha \\ y = b \cdot \sin \alpha \end{cases}, \quad \alpha \in [0; 2\pi]. \quad (2.4)$$

Тук α е мярката на ъгъл, чийто геометричен смисъл е изяснен в [33].



Фиг. 2.2 Графиката на елипса.

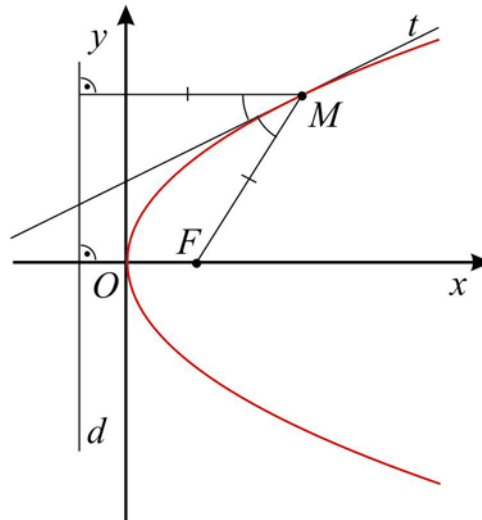
Числата: a, b се наричат съответно *голяма* и *малка полуос*; $d_{1,2} : x = \pm \frac{a}{e}$ - *директриси*; $2c$ - *междупокусно разстояние*, $c = \sqrt{a^2 - b^2}$; $e = \frac{c}{a} < 1$ - *ексцентрицитет*. Точките $A_{1,2}(\pm a, 0)$, $B_{1,2}(0, \pm b)$ (фиг. 2.2) се наричат *върхове*, а $F_{1,2}(\pm c, 0)$ - *фокуси*.

Каноничното уравнение на параболата (фиг. 2.3) е:

$$\pi : y^2 = 2px, \quad p \in \mathbb{R}, p \neq 0. \quad (2.5)$$

СПУ на параболата се намират непосредствено:

$$\pi : \begin{cases} x = \frac{u^2}{2p} \\ y = u \end{cases}, \quad u \in \mathbb{R}, p \in \mathbb{R}, p \neq 0. \quad (2.6)$$



Фиг. 2.3 Графиката на параболата.

Числото p се нарича *фокален параметър* на π ; т. O - *върх*; т. $F\left(\frac{p}{2}, 0\right)$ - *фокус*; правата $d: x = -\frac{p}{2}$ - *директриса*; правата t – *тангента* към π в т. M (фиг. 2.3).

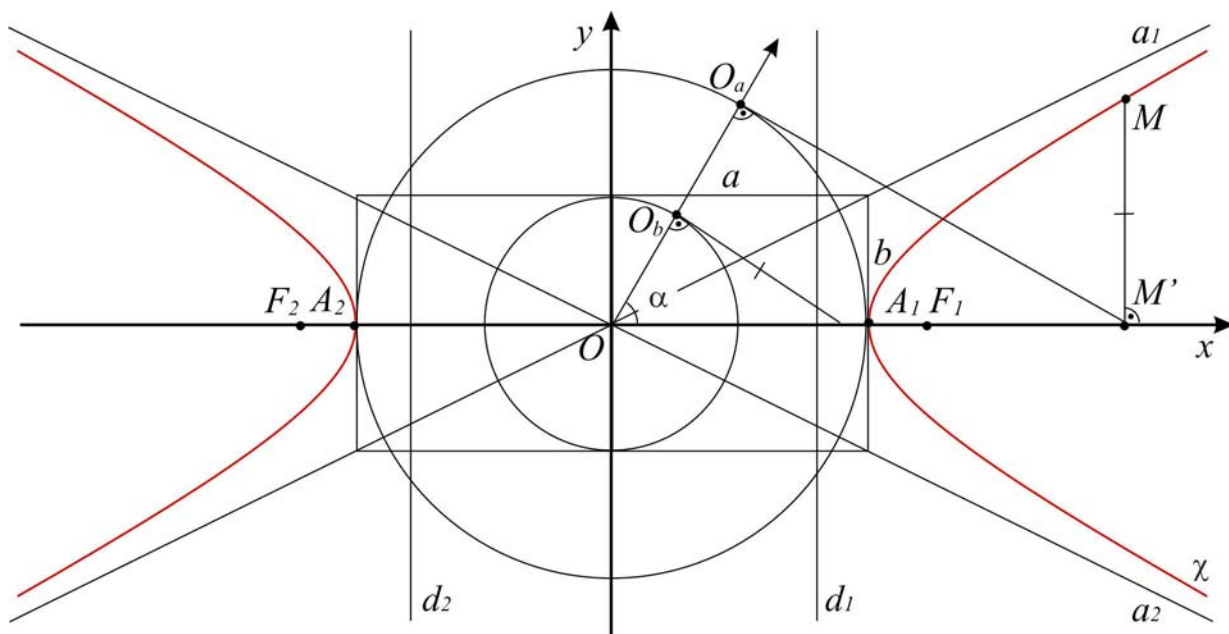
Каноничното уравнение на хипербола (фиг. 2.4) е:

$$\chi: \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \quad a \geq b > 0. \quad (2.7)$$

СПУ на хиперболата χ се получават от основното тригонометрично тъждество $\sin^2 \alpha + \cos^2 \alpha = 1$, приложено във варианта $\frac{1}{\cos^2 \alpha} - \operatorname{tg}^2 \alpha = 1$ и са

$$\chi: \begin{cases} x = \frac{a}{\cos \alpha}, & \alpha \in [0; 2\pi], \alpha \neq \frac{\pi}{2}, \frac{3\pi}{2}. \\ y = b \cdot \operatorname{tg} \alpha \end{cases} \quad (2.8)$$

Тук α е мярката на ъгъл, чийто геометричен смисъл е изяснен в [33].



Фиг. 2.4 Графиката на хипербола.

Числата: a, b се наричат съответно *голяма и малка полуос*; $d_{1,2} : x = \pm \frac{a}{e}$ - *директриси*; $2c$ - *междупокусно разстояние*, $c = \sqrt{a^2 + b^2}$; $e = \frac{c}{a} > 1$ - *ексцентрицитет*. Точките $A_{1,2}(\pm a, 0)$ се наричат *върхове*, а $F_{1,2}(\pm c, 0)$ - *фокуси* (фиг. 2.4). По диагоналите на правоъгълника $\Pi : \{x = \pm a, y = \pm b\}$ са двете *асимптоти* $a_{1,2} : y = \pm \frac{b}{a}x$.

Кривите, които са обект на изследването е нужно да бъдат разгледани в дясна декартова координатна система $K = \{O, \vec{e}_1, \vec{e}_2, \vec{e}_3\}$ в \mathbb{R}^3 . Това се обуславя от факта, че крайната цел е те да бъдат приложени в 3D ГСГМ като графични примитиви.

2.2 Идеята за представяне на кривите от втора степен чрез NURBS криви

Ще бъде изложен приетия подход за представяне на кривите от втора степен чрез NURBS криви, при който се изисква да се знае типа на кривата и параметрите, които я специфицират в нейните СПУ. Ако тя е зададена с параметрите от уравнение (1.1), то тогава първо трябва да се извърши канонизиране до (2.3), (2.5) или (2.7).

Представянето на конично сечение ще намерим като определим точно мястото на контролните върхове и стойностите на техните тегла в NURBS модела на кривата. Така NURBS кривата ще може да бъде лесно подразделена, което ще осигури възможност

за модификация на кривата само в отделни нейни части – *сегменти*. По този начин сегмент от представената в NURBS модела крива ще може да бъде променян, а останалата част ще запазва геометричните свойства и характеристиките на съответната крива от втора степен. Това осигурява нови възможности и по-голяма гъвкавост при моделиране на реални обекти, които (или поне част от тях) са във формата на конично сечение.

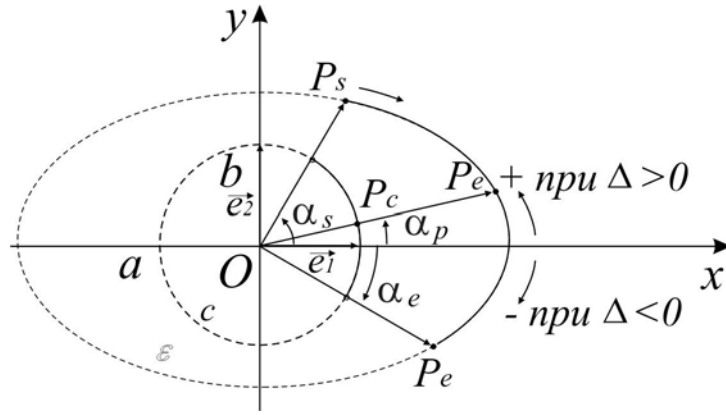
След като това е направено и ако е било извършено канонизиране, може да се приложи обратния ход на канонизирането върху получените контролни върхове на NURBS кривата (тъй като те и самите криви са афинни инварианти). Така ще се получи първоначалното положение на кониката в равнината или пространството, но представена вече чрез NURBS крива.

Каноничните уравнения (2.3, 2.5 и 2.7) на квадратичните криви не дават ограничение за дъга, която да може да се начертае с помощта на компютърно представяне. Следователно е нужно да зададем ограничения, които да определят дъгата. Предимство на задаването на коничните сечения чрез скаларно-параметричните им уравнения (2.4, 2.6 и 2.8) пред задаването им в канонични уравнения е, че те подготвят процедура за компютърно чертане чрез варирането на параметъра в определен диапазон. Затова за NURBS представянето на кривите от втора степен ще използваме задаването им чрез СПУ.

2.3. Представяне на дъга от елипса чрез NURBS крива

Можем да специфицираме дъга от елипсата ε като задаваме начална и крайна стойност на параметъра α от СПУ (2.4), а именно $\alpha \in [\alpha_s, \alpha_e]$ (*s-start, e-end*).

Равнината на кониките означаваме с λ и нека $K_\lambda = \{O, \vec{e}_1, \vec{e}_2\}$ да е локалната дясна декартова координатна система в λ на вече канонизираната крива ε с уравнение (2.3). Двойката (\vec{e}_1, \vec{e}_2) задава положителна посока на обхождане на точките в λ (фиг. 2.5).



Фиг. 2.5 Дъга от елипсата ε .

2.3.1. Посока на построяване на дъгата

За скалярно-параметричните уравнения (2.4) имаме изискване $\alpha \in [0; 2\pi]$, но ние ще освободим α да варира в диапазона $(-\infty; +\infty)$. Следствието от това е, че когато $|\alpha_s - \alpha_e| > 2\pi$, варирането на α в диапазон $[\alpha_s; \alpha_e]$, където $\alpha_s, \alpha_e \in \mathbb{R}$, ще предизвика (многократно) обхождане на елипсата от радиус-вектора.

Нека α_s определя точка $P_s(x_s, y_s)$ от ε с радиус вектор \vec{r}_s , а α_e определя точка $P_e(x_e, y_e)$ от ε с радиус вектор \vec{r}_e . Знаем (от [33, 22-24 с.]), че векторите \vec{r}_s и \vec{r}_e са линейно независими $\Leftrightarrow \Delta = \begin{vmatrix} x_s & y_s \\ x_e & y_e \end{vmatrix} \neq 0 \Leftrightarrow \text{rang} \begin{pmatrix} x_s & y_s \\ x_e & y_e \end{pmatrix} = 2$, и двойката (\vec{r}_s, \vec{r}_e) е дясна, като задава положителна посока на обхождане $\Leftrightarrow \Delta > 0$ и лява, като задава отрицателна посока на обхождане $\Leftrightarrow \Delta < 0$ (фиг. 2.5).

Следователно ако $\alpha_s < \alpha_e$ ще имаме положителна посока на обхождане на точките от дъгата, ако $\alpha_s > \alpha_e$ ще имаме отрицателна посока, а ако $\alpha_s = \alpha_e$ - не съществува дъга. P_s ще наричаме *начална точка*, P_e - *крайна точка*, α_s - *начален ъгъл* и α_e - *краен ъгъл*.

Определение 2.1 Положителна дъга от ε (фиг. 2.5) ще наричаме кривата получена чрез движението на радиус-вектора \vec{r} в положителна посока от P_s до P_e когато $\Delta > 0$ за $\alpha \in [\alpha_s; \alpha_e]$. Отрицателна дъга от ε ще наричаме кривата получена чрез движението на радиус-вектора \vec{r} в отрицателна посока от P_s до P_e когато $\Delta < 0$ за $\alpha \in [\alpha_s; \alpha_e]$.

Като имаме предвид поясненията, направени до момента, задачата която трябва да се реши, може да бъде формулирана така:

Задача 2.1 Да се построи дъга от елипса представена чрез NURBS крива. Елипсата да е в каноничен вид спрямо K_λ с полуос по $Ox - a$ и полуос по $Oy - b$. Дъгата да се задава със стартов ъгъл α_s и краен ъгъл α_e , на които съответстват точките P_s и P_e от кривата, $\alpha_s, \alpha_e \in \mathbb{R}$. Ако $\alpha_s < \alpha_e$ дъгата да се построява в положителна посока; ако $\alpha_s > \alpha_e$ - дъгата да се построява в отрицателна посока; ако $\alpha_s = \alpha_e$ - не се построява дъга.

$$\text{От уравнение (2.4) следва, че } P_s : \begin{cases} x_s = a \cos \alpha_s \\ y_s = b \sin \alpha_s \end{cases} \text{ и } P_e : \begin{cases} x_e = a \cos \alpha_e \\ y_e = b \sin \alpha_e \end{cases}.$$

2.3.2. Основна идея за решение

Първо ще разгледаме дъга γ_c от единичната окръжност c определена от α_s и α_e (фиг. 2.5). Ще намерим представянето ѝ чрез NURBS крива. След това, чрез подходяща афинна трансформация на получената крива ще намерим представянето на търсената дъга γ_ε , която е също определена от α_s и α_e , но представяща елипсата ε .

2.3.3. Афинна трансформация на единична окръжност в елипса

Нека α_p определя точките: P_c от единичната окръжност c и P_l от елипсата ε (фиг. 2.5). Тогава имаме $P_c : \begin{cases} x_c = \cos \alpha_p \\ y_c = \sin \alpha_p \end{cases}$, $P_l : \begin{cases} x_l = a \cos \alpha_p \\ y_l = b \sin \alpha_p \end{cases}$. Известно е, че точката P_l

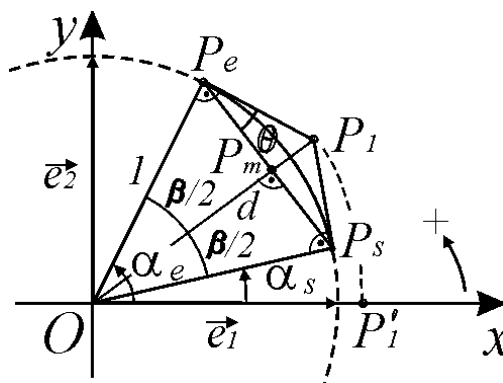
можем да получим от точката P_c чрез уравнението:

$$P_l = TP_c, \quad T = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}. \quad (2.9)$$

Матрицата T задава нужната афинна трансформация (биективно налагаща c върху ε), като композиции на две дилатации с взаимно перпендикулярни оси – по Ox с коефициент a и по Oy с коефициент b .

2.3.4. Алгоритъм за построяване на NURBS крива, представяща дъга от единичната окръжност

Ще предложим алгоритъм, който построява NURBS крива, представяща дъга от единичната окръжност c , определена от α_s и α_e ($\alpha_s, \alpha_e \in (-\infty; +\infty)$) с условието $\alpha_s < \alpha_e$, т.е. изграждащ само положителна дъга. Цялата дъга ще разделяме на сегменти, всеки от които ще бъде най-много 90° .



Фиг. 2.6 Дъга от единичната окръжност s , представена чрез крива на Безие от втора степен.

Нека първо да разгледаме частния случай когато $|\alpha_s - \alpha_e| \leq \pi/2$ (фиг. 2.6). Знаем, че $|OP_s| = |OP_e| = 1$. Нека допирателните към точките P_s и P_e да се пресичат в точката P_1 . Тогава P_s, P_1, P_e са точките, по които може да се изгради рационална крива на Безие от втора степен, която да представлява дъгата. Уравнението на тази крива с три характеристични точки $M_i, i = 0, 1, 2$ се задава с векторната функция:

$$\beta(t) = \frac{\sum_{i=0}^2 M_i w_i B_{i,2}(t)}{\sum_{j=0}^2 w_j B_{j,2}(t)} = \sum_{i=0}^2 M_i \left\{ \frac{w_i B_{i,2}(t)}{\sum_{j=0}^2 w_j B_{j,2}(t)} \right\}, \quad t \in [0,1]. \quad (2.10)$$

За да имаме нормална параметризация полагаме $w_s = w_e = 1$. Остава да намерим P_1 и w_1 .

Имаме $|P_s P_1| = |P_e P_1|$ и $\sphericalangle OP_s P_1 = \sphericalangle OP_e P_1 = 90^\circ$. Нека $\sphericalangle P_s O P_e = \beta \Rightarrow \beta = |\alpha_s - \alpha_e|$ и $\sphericalangle P_s O P_1 = \sphericalangle P_e O P_1 = \beta/2$. От $\perp_{\Delta} OP_s P_1$ имаме $\cos \beta/2 = \frac{|OP_s|}{|OP_1|} \Rightarrow |OP_1| = \frac{|OP_s|}{\cos \beta/2}$.

Нека $|OP_1| = d$ и точка $P_1' = (d, 0)$. Тогава точката P_1' чрез ротация на ъгъл $\alpha_s + \beta/2$

ще се изобрази в точката $P_1: \begin{cases} x_1 = \cos(\alpha_s + \beta/2).d \\ y_1 = \sin(\alpha_s + \beta/2).d \end{cases}$.

Нека $P_s P_e \cap OP_1 = P_m$ и $\sphericalangle P_1 P_e P_m = \theta$, тогава от уравнение (1.29) \Rightarrow
 $w_1 = \frac{|P_e P_m|}{|P_e P_1|} = \cos \theta$. От $\triangle OP_e P_1$ и $\triangle P_e P_m P_1$ имаме $\sphericalangle OP_e P_1 = \sphericalangle P_e P_m P_1 = 90^\circ$ и
 $\sphericalangle OP_1 P_e = \sphericalangle P_e P_1 P_m \Rightarrow \sphericalangle P_e OP_1 = \sphericalangle P_1 P_e P_m = \beta/2 = \theta$. Следователно $w_1 = \cos \beta/2$.

Точките P_s, P_1, P_e са трите характеристични точки с тегла w_s, w_1, w_e в модела на Безие на рационална крива от втора степен (уравнение (2.10)).

Сега ще предложим алгоритъм за изграждане на цялостна дъга γ_c от c съставена от няколко по-малки сегмента. Всеки сегмент представлява дъга от c определена от начален ъгъл α_i и краен ъгъл α_{i+1} , като $\alpha_0 = \alpha_s$, $\alpha_n = \alpha_e$ и $|\alpha_i - \alpha_{i+1}| = \beta \leq 90^\circ$ за $i = 0, 1, \dots, n-1$, т.е. интервалът $[\alpha_s; \alpha_e]$ се разделя на n равни части. Тогава броят на контролните върхове трябва да бъде

$$m = 2n + 1. \quad (2.11)$$

Нека самите върхове да бъдат P_j , $j = 0, \dots, 2n$. Всеки сегмент ще се състои от върховете P_l, P_{l+1}, P_{l+2} , $l = 0, 2, 4, \dots, 2(n-1)$. Първият връх от кривата определяме по следния начин:

$$P_0: \begin{cases} x_0 = \cos(\alpha_0) \\ y_0 = \sin(\alpha_0) \\ w_0 = 1 \end{cases}, \quad (2.12)$$

а всеки следващи два така:

$\alpha = \alpha_0$ и тогава

$$P_{l+1}: \begin{cases} x_{l+1} = \cos(\alpha + \beta/2).d \\ y_{l+1} = \sin(\alpha + \beta/2).d \\ w_{l+1} = \cos(\beta/2) \end{cases}, P_{l+2}: \begin{cases} x_{l+2} = \cos(\alpha + \beta) \\ y_{l+2} = \sin(\alpha + \beta) \\ w_{l+2} = 1 \end{cases}, \alpha + = \beta, \quad (2.13)$$

за $l = 0, 2, 4, \dots, 2(n-1)$.

Алгоритъм 2.1 ComputeNURBSCircularArc

Вход: α_s (начален ъгъл), α_e (краен ъгъл), segments (брой на сегментите);

Изход: aP (масив от контролните върхове), nNP (брой на контролните върхове), aK (възлов вектор), nNK (брой на възлите);

Процедура:

- 1.) Определя се броя n на дъгите (по-малки или равни на 90°).
- 2.) Ако броят на дъгите е по-голям от 0 то
 - намира се броя (m) на контролните върхове от уравнение (2.11);
 - намират се контролните върхове $P_j, j = 0, \dots, 2n$, на всеки сегмент $P_l, P_{l+1}, P_{l+2}, l = 0, 2, 4, \dots, 2(n-1)$ по уравнения (2.12) и (2.13);
 - определя се броя на възлите, който трябва да бъде $m + 3$ (следва от Дефиниция 1.3) и стойностите им във възловия вектор като интервалът $[0;1]$ се разделя на равни части.

в противен случай "Край".

- 3.) Край.

Описанието на алгоритъма чрез псевдопрограмен код изглежда така:

```
1) ComputeNURBSCircularArc( $\alpha_s, \alpha_e, \text{segments}, aP[], nNP, aK[], nNK$ ) {
2)   buf  =  $\alpha = |\alpha_s - \alpha_e|$ ;
3)   narcs = 0;
4)   while( buf > 0.0 ) do { narcs++; buf -=  $\pi/2$ ; }
5)   if( 0 < narcs && narcs < segments ) then narcs = segments;
6)   if( narcs > 0 ) then {
7)      $\beta = \alpha / \text{narcs}$ ; nNP = 2*narcs + 1;
8)     w1 = cos( $\beta/2$ );
9)     aP[0].x = cos( $\alpha_s$ ); aP[0].y = sin( $\alpha_s$ ); aP[0].w = 1.0;
10)    index = 0; d = 0.0; angle =  $\alpha_s$ ;
11)    for i in 0 to narcs-1 do {
12)      a. d = 1/cos( $\beta/2$ );
13)      b. aP[index+1].x = cos(angle+ $\beta/2$ )*d;
14)      c. aP[index+1].y = sin(angle+ $\beta/2$ )*d; aP[index+1].w = w1;
15)      d. aP[index+2].x = cos(angle+ $\beta$ );
16)      e. aP[index+2].y = sin(angle+ $\beta$ ); aP[index+2].w = 1;
17)      f. index += 2; angle +=  $\beta$ ; }
18)    nNK = nNP + 3; step = 1.0; g = 0.0;
19)    if( nNK > 4 ) then step = 1/( (nNK-4)/2 + 1 );
20)    for i in 0 to nNK-1 do {aK[i]=aK[i+1]=g; g+=step; i+=2; }
```

15) }
16) }

Коментари по алгоритъм 2.1:

1. На редове от 2) до 5) се определя броя на дъгите, от които ще бъде изградена NURBS кривата.

2. Ако броя на сегментите е по-голям от броя на дъгите, то се взема под внимание броя на сегментите (ред 5)) – прави се подразделяне на NURBS кривата.

3. Ако броят на сегментите е по-голям от 0 (ред 6)) започва процедура за определяне на контролните върхове: определя се техния брой (ред 7)); стойността на теглата на средните точки (ред 8)); изчисляват се координатите на всички контролни върхове (редове от 9) до 11)); на ред 13) се изчислява стъпката на промяна на стойностите на възловия вектор, а на ред 14) се изчисляват стойностите му.

4. След изпълнението на Алгоритъм 2.1 получаваме контролните върхове в $aP[nNP]$ и възловия вектор в $aK[nNK]$, които са нужни за изграждането на NURBS кривата. Тази крива точно представя търсената дъга γ_c .

5. Коректността на Алгоритъм 2.1 се определя от коректността на изложения по-горе подход за построяване на NURBS кривата, представляваща дъга от единичната окръжност.

2.3.5. Представяне на дъга от елипса чрез дъга от единичната окръжност представена чрез NURBS крива от втора степен

След като имаме представянето на дъгата от единичната окръжност като NURBS крива, остава да извършим трансформирането ѝ до дъгата от елипсата ε чрез прилагането на трансформацията от уравнение (2.9). В резултат получаваме търсената дъга γ_ε представена като NURBS крива от втора степен. Алгоритъм 2.2 съдържа стъпките на тази процедура, а фигура 2.7 показва резултати при различни входни данни.

Алгоритъм 2.2 ComputeNURBSEllipticalArc

Вход: a, b, α_s , α_e , segments;

Изход: aP, nNP, aK, nNK;

Процедура:

- 1.) Определя се посоката на дъгата по подхода, описан в подраздел 2.3.1.
- 2.) Изчислява се NURBS кривата γ_c чрез прилагане на Алгоритъм 2.1 – ComputeNURBSCircularArc.

- 3.) Когато $\alpha_s > \alpha_e$ се обръща реда на контролните върхове в NURBS представянето (с функцията ReverseOrderOfTheControlPoints) за да извърши построяване на кривата в обратна посока.
- 4.) Върху получените контролни върхове на γ_c се прилага афинната трансформация (2.9) за да се получат контролните върхове на γ_ε .
- 5.) Край.

Псевдопрограмният код е:

```

1) ComputeNURBSEllipticalArc(a,b,  $\alpha_s$ ,  $\alpha_e$ , segments, aP[], nNP, aK[], nNK) {
2)   if( $\alpha_s > \alpha_e$ ) then {
       a.  $\beta_s = \alpha_e$ ;  $\beta_e = \alpha_s$ ; bAnglesExchanged := true; };
3)   else {
       a.  $\beta_s = \alpha_s$ ;  $\beta_e = \alpha_e$ ;
       b. bAnglesExchanged = false; };
4)   ComputeNURBSCircularArc( $\beta_s$ ,  $\beta_e$ , segments, aP, nNP, aK, nNK);
5)   if(bAnglesExchanged) then ReverseOrderOfTheControlPoints(aP);
6)   for i in 0 to nNK-1 do { aP[i].x *= a; aP[i].y *= b; }
7) }
```

Коментари по Алгоритъм 2.2:

- 1.) Като резултат от изпълнението на Алгоритъм 2.2 получаваме контролни върхове aP[nNP] и възлов вектор aK[nNK]. Те са необходимите входни данни за изчертаването на NURBS кривата γ_ε .
- 2.) Коректността на Алгоритъм 2.2 се определя от коректността на изложения по-горе подход за построяване на NURBS кривата γ_ε , представляваща дъга от елипсата ε .
- 3.) NURBS кривата γ_ε представя точно търсената дъга от елипсата ε .

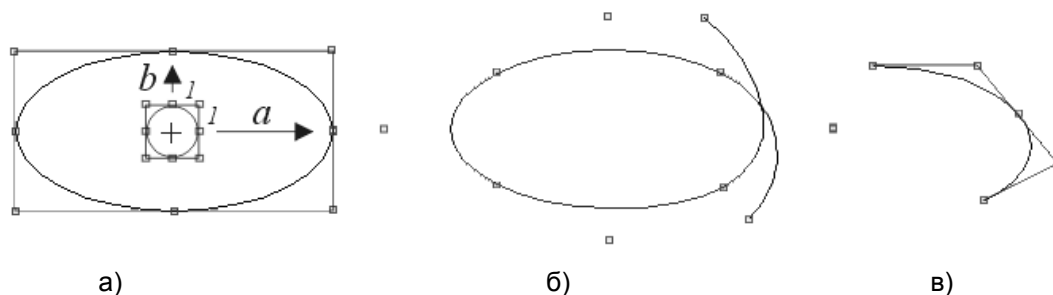
NURBS кривата има уравнение от вида:

$$\gamma(t) = \sum_{i=0}^n R_{i,2}(t) P_i, \quad t \in [0;1],$$

където P_i са $n+1$ контролни върха, с тегла w_i , възлов вектор $T = \{t_0 = 0, 0, 0, \tau_1, \tau_1, \tau_2, \tau_2, \dots, \tau_m, \tau_m, 1, 1, t_{n+3} = 1\}$ и състояща се от $m+1$ на брой сегменти.

Локална модификация: При промяна на позицията на контролен връх P_i или стойността на неговото тегло, кривата се променя само за $t \in [t_i, t_{i+3}]$, $i = 0, \dots, n$.

На фигура 2.7б е показана NURBS крива състояща се от пет сегмента, представляваща дъга от елипса. Контролните ѝ върхове са $P_i, i = 0, \dots, 10$ и възловият вектор е $T = \{0, 0, 0, 1/5, 1/5, 2/5, 2/5, 3/5, 3/5, 4/5, 4/5, 1, 1, 1\}$. Позициите на крайните два контролни върха са променени. Следователно, частите от кривата за $t \in ([0, 1/5] \cup [4/5, 1])$ вече не са дъги от елипсата, а за $t \in [1/5, 4/5]$ - кривата представлява съответната дъга от елипсата.



Фиг. 2.7 NURBS представяне на:

- а) елипса; б) елиптична дъга от -45° до 405° като позицията на крайните две характеристични точки е променена; в) елиптична дъга от -45° до 90° .

Пример 2.1 Ще намерим контролните върхове, техните тегла и уравнението на NURBS крива γ представляваща дъга от елипса ε , която е определена от $\alpha_s = 0$, $\alpha_e = \pi/2$, $a = 3$ и $b = 2$. За контролните върхове на NURBS кривата, която представлява дъгата от единичната окръжност, получаваме: $(1,0)$, $(1,1)$, $(0,1)$, съответно с тегла 1 , $\sqrt{2}/2$, 1 . За векторното равенство на тази NURBS крива получаваме (фиг. 2.8а):

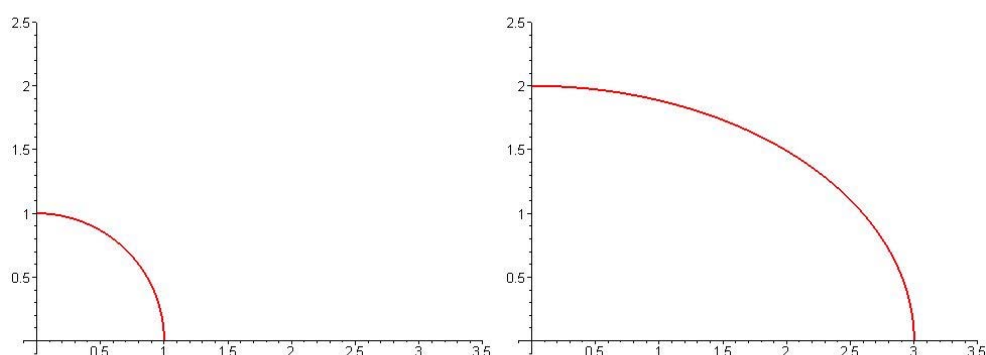
$$\gamma: \vec{r} = \gamma(t) = \begin{cases} x(t) = \frac{(1-t)^2 + \sqrt{2}t(1-t)}{(1-t)^2 + \sqrt{2}t(1-t) + t^2} \\ y(t) = \frac{\sqrt{2}t(1-t) + t^2}{(1-t)^2 + \sqrt{2}t(1-t) + t^2} \end{cases}, t \in [0, 1].$$

След прилагането на мащабиране върху контролните върхове на γ с матрицата на афинна трансформация $T = \begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$ получаваме контролните върхове: $(3,0)$, $(3,2)$,

$(0,2)$, съответно с тегла $1, \sqrt{2}/2, 1$. С тези контролни върхове, γ вече представлява търсената дъга от ε (фиг. 2.8б). NURBS кривата има уравнението:

$$\gamma: \vec{r} = \gamma(t) = \begin{cases} x(t) = \frac{3 \cdot (1-t)^2 + 3 \cdot \sqrt{2} \cdot t(1-t)}{(1-t)^2 + \sqrt{2} \cdot t(1-t) + t^2} \\ y(t) = \frac{2 \cdot \sqrt{2} \cdot t(1-t) + 2 \cdot t^2}{(1-t)^2 + \sqrt{2} \cdot t(1-t) + t^2} \end{cases}, t \in [0,1].$$

Ако позицията на някой от контролните върхове бъде променена, то тогава ново-получената крива няма да бъде дъга от елипсата.

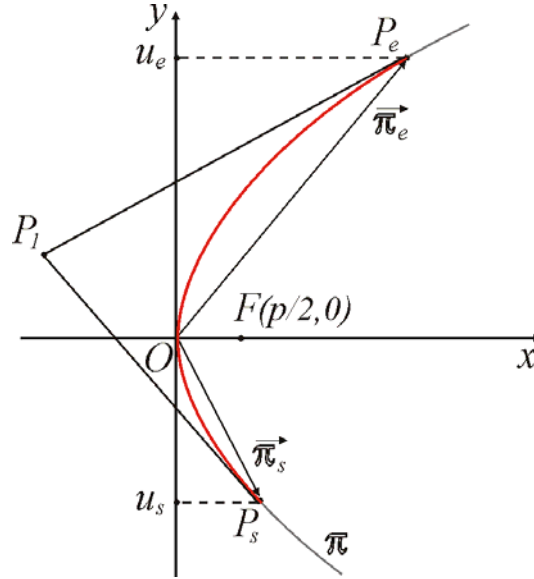


Фиг. 2.8 NURBS крива представляваща дъга от:
а) единичната окръжност, б) елипса.

2.4. Представяне на дъга от парабола чрез NURBS крива

Задача 2.2 Да се начертае дъга от парабола представена като NURBS крива. Параболата да се специфицира с параметрите от нейните СПУ (2.6).

Можем да специфицираме дъга от параболата π като задаваме начална и крайна стойност на фокалния параметър p , а именно $p \in [p_s, p_e], p \neq 0, p_s, p_e \in \mathbb{R}$.



Фиг. 2.9 Дъга от параболата π ,
представена чрез NURBS крива от втора степен.

Нека λ да е равнина на параболата π и $K_\lambda = \{O, \vec{e}_1, \vec{e}_2\}$ е дясна декартова координатна система в нея. Нека параболата π бъде зададена с уравнение (2.6) (следователно параметърът p е даден).

Нека u_s определя точката $P_s(x_s, y_s)$ от π и u_e определя точката $P_e(x_e, y_e)$ от π (фиг. 2.9). Точка P_s ще наричаме *начална точка*, P_e - *крайна точка*, u_s - *начална стойност на параметъра* и u_e - *крайна стойност на параметъра*. Така варирайки параметъра u от u_s до u_e в уравнение (2.6) ще получим дъга от π . Означаваме тази дъга с δ_π .

На базата на тези означения формулираме следното твърдение:

Твърдение 2.1 Графиката на дъгата δ_π от параболата π и NURBS кривата γ зададена с уравнение (1.15) от степен $p=2$, възлов вектор $U = \{0, 0, 0, 1, 1, 1\}$ и контролни върхове

$$P_s = (u_s^2, 2pu_s), \quad P_1 = \left(u_e \left((u_e - u_s)^2 (u_e + u_s) - u_e \right), p(u_e - u_s)^2 (u_e + u_s) \right) \quad \text{и} \\ P_e = (u_e^2, 2pu_e) \quad \text{с тегла } w_0 = w_1 = w_2 = 2p \text{ са идентични.}$$

Доказателство:

Уравнението на NURBS кривата зададена с уравнение (1.15) за $p=2$, възлов вектор $U = \{0, 0, 0, 1, 1, 1\}$ и произволен избор на три характеристични точки M_0, M_1, M_2 е:

$$\gamma(t) = \frac{\sum_{i=0}^2 N_{i,2}(t)w_i M_i}{\sum_{j=0}^2 N_{j,2}(t)w_j}, \quad 0 \leq t \leq 1. \quad (2.14)$$

Известно е, че допирателните към двете крайни точки винаги ще се пресичат в т. M_1 , т.е. M_0M_1 е допирателната към кривата в т. M_0 и M_2M_1 е допирателната към кривата в т. M_2 . Това следва от свойствата за NURBS кривите дадени в подраздел 1.2.4.

В раздел 1.4 (фиг. 1.18) показахме, че когато за рационална крива на Безие от втора степен се избере $w_1=1$ се получава крива, която представлява дъга от парабола.

Дъгата δ_π определяме със стойностите u_s и u_e на фокалния параметър u , определящи точките $P_s(x_s, y_s)$ и $P_e(x_e, y_e)$.

В подраздел 1.2.4. (от посочените там геометрични характеристики на NURBS крива) се вижда, че NURBS крива без вътрешни възли е рационална крива на Безие. От тази характеристика и теорията дадена в точка 1.2.2.1 следва, че NURBS крива зададена с уравнение (2.14) за степен $p=2$, възлов вектор $U = \{0, 0, 0, 1, 1, 1\}$ и произволен избор на три характеристични точки M_0, M_1, M_2 с тегла $w_0 = w_1 = w_2 = 1$ представлява дъга от парабола. Нека NURBS крива, която отговаря на тези условия да бележим с γ .

Нека за γ да изберем $M_0 = P_s$ и $M_2 = P_e$. Нека тангентите към π в точките P_s и P_e ($P_s \neq P_e$) се пресичат в точка P_1 . Тъй като пресечната точка на допирателните към γ в точките M_0 и M_2 е M_1 , следователно $M_1 \equiv P_1$.

За тангентите l_{P_s} и l_{P_e} към параболата π съответно в точките P_s и P_e ($P_s \neq P_e$) от [33, 112 с.] е в сила:

$$\begin{aligned} l_{P_s} &= y \cdot y_s = p(x + x_s) \\ l_{P_e} &= y \cdot y_e = p(x + x_e) \end{aligned}$$

Тогава координатите на пресечната точка $P_1 = l_{P_s} \cap l_{P_e} \left(P_1(x_1, y_1) \right)$ са решенията x_1 и y_1 на системата линейни уравнения:

$$\begin{cases} px_1 - y_s y_1 + px_s = 0 \\ px_1 - y_e y_1 + px_e = 0 \end{cases}.$$

След решаването на тази система по формулите на Крамер, получаваме $x_1 = \frac{u_e}{2p} \left((u_e - u_s)^2 (u_e + u_s) - u_e \right)$ и $y_1 = \frac{(u_e - u_s)^2 (u_e + u_s)}{2}$.

Координатите на точките P_s и P_e намираме директно от СПУ на параболата за стойностите на фокалния параметър съответно u_s и u_e . След това хомогенните координати на трите контролни върхове умножаваме с $2p$. С това Твърдение 2.1 е доказано.

Построяването на NURBS кривата γ на дъга δ_π от параболата π следва директно от Твърдение 2.1. Тук обаче възниква въпросът относно построяването на дъга от параболата чрез NURBS крива, но с произволен брой на сегментите с цел възможност за локална модификация. За тази цел правим подразделяне (чрез вмъкване на нови възли във възловия вектор и добавяне на контролни върхове [91, 95]) на получената NURBS крива чрез следния алгоритъм:

Алгоритъм 2.3 SubdivideTheNURBSCurve

Вход: segments, P;

Изход: aP, nNP, aK, nNK;

Процедура:

- 1.) Определят се броя на контролните върхове, броя на възлите и възловия вектор.
- 2.) На базата на броя на сегментите се подразделя NURBS кривата.
- 3.) Край.

Псевдопрограмният код изглежда така:

```

1) SubdivideTheNURBSCurve(segments, P[], aP[], nNP, aK[], nNK) {
2)   step=1.0/segments; g=0;
3)   nNP=2*segments+1; numKnots = nNP+3; q=1; i=3;
4)   while( q < segments ) do {
5)     a.g=g+step; aK[i+1]=g; aK[i+2]=g; i+=2; q++; }
6)   for i in (numKnots-3+1) to numKnots do aK[i]=1.0;
7)   nNK=numKnots; k=segments; t=1.0/k; e=1; s=0;
   while( k > 1 ) do {
     a.t=1.0/k; e=3;
     b.for i in 1 to 3 do {

```

```

1. aP[2*s+i]=P[1]; aL[e]=P[e];
2. for j in 1 to e-1 do P[j]=(1.0-t)*P[j]+t*P[j+1];
3. e=e-1; }
c. for i in 1 to 3 do P[i]=aL[i];
d. s=s+1; k=k-1; }
8) for i in 1 to 3 do aP[nNP-3+i]=P[i];
9) }

```

Коментари по алгоритъм 2.3:

1. Като входни параметри се подават броя на сегментите *segments* и масивът *P* с трите контролни върхове на NURBS кривата γ .
2. Подразделянето на γ на *segments* на брой сегменти става в точки 7) и 8).
3. След изпълнението на Алгоритъм 2.3 получаваме контролните върхове в *aP[nNP]* и възловия вектор в *aK[nNK]*, които са нужни за изграждането на NURBS кривата. Тази крива точно представя търсената дъга γ .
4. NURBS кривата има уравнение от вида:

$$\gamma(t) = \sum_{i=0}^n R_{i,2}(t)P_i, \quad t \in [0;1],$$

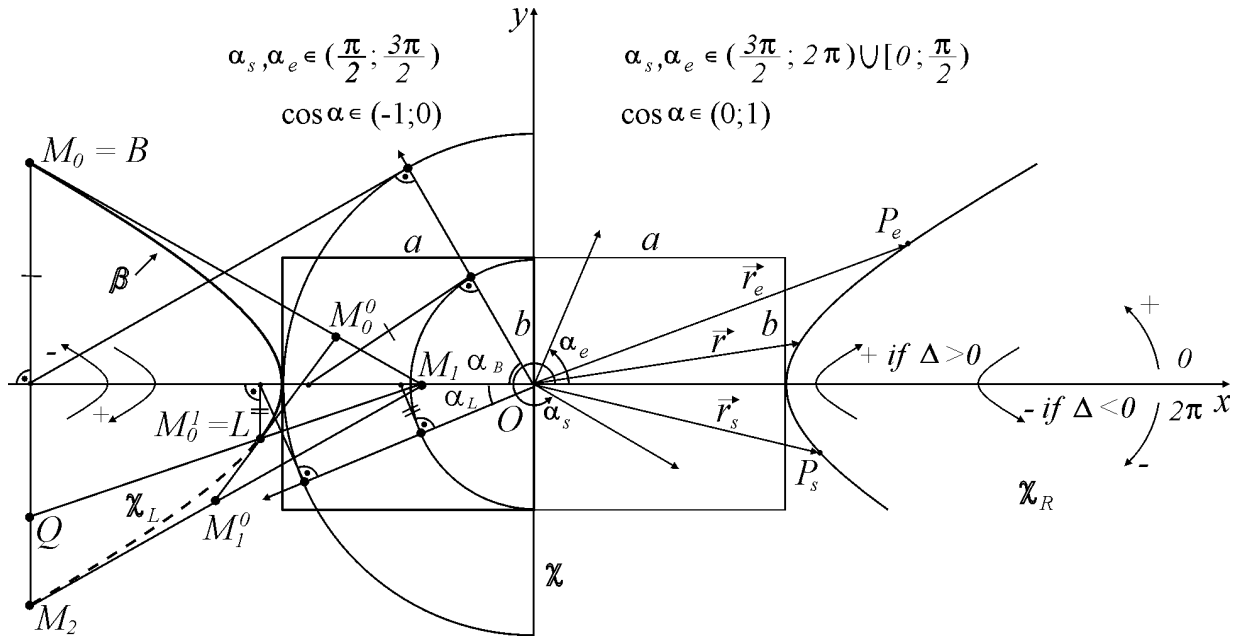
където P_i са $n+1$ контролни върха, с тегла w_i , възлов вектор $T = \{t_0 = 0, 0, 0, \tau_1, \tau_1, \tau_2, \tau_2, \dots, \tau_m, \tau_m, 1, 1, t_{n+3} = 1\}$ и състояща се от $m+1$ на брой сегменти.

Локална модификация: При промяна на позицията на контролен връх P_i или стойността на неговото тегло, кривата се променя само за $t \in [t_i, t_{i+3}]$, $i = 0, \dots, n$.

2.5. Представяне на дъга от хипербола чрез NURBS крива

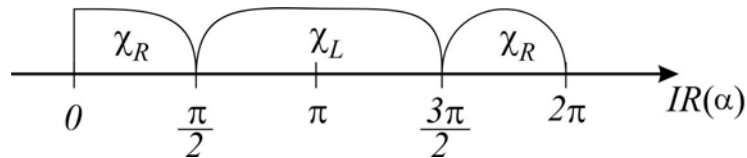
Определение 2.2 Дясно крило на хиперболата χ (уравнение (2.7)) ще наричаме частта от χ , лежаща в I и IV квадрант на K_λ и ще го бележим с χ_R (*R-right*), а ляво крило на χ ще наричаме частта от χ лежаща в II и III квадрант на K_λ и ще го бележим с χ_L (*L-left*) (фиг. 2.10).

Така $\chi_R(\chi_L): \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$ за $x > a$ ($x < a$).



Фиг. 2.10 Графиката на хиперболатата χ .

Задача 2.3 Да се начертае дъга от хипербола зададена със специфициращите я параметри от нейните СПУ (уравнение (2.8)) като тази част от хиперболатата да бъде представена като NURBS крива.



Фиг. 2.11 Допустимите стойности на ъгъла α , определящ дъгите χ_R и χ_L на хиперболатата χ .

Дъгата, която трябва да се начертае, се определя от $a, b > 0$ и варирането на параметъра α в някакъв зададен диапазон $[\alpha_s, \alpha_e]$. По дефиниция:

$\alpha \in [0; 2\pi], \alpha \neq \frac{\pi}{2}, \frac{3\pi}{2}$. Това може да бъде записано и така

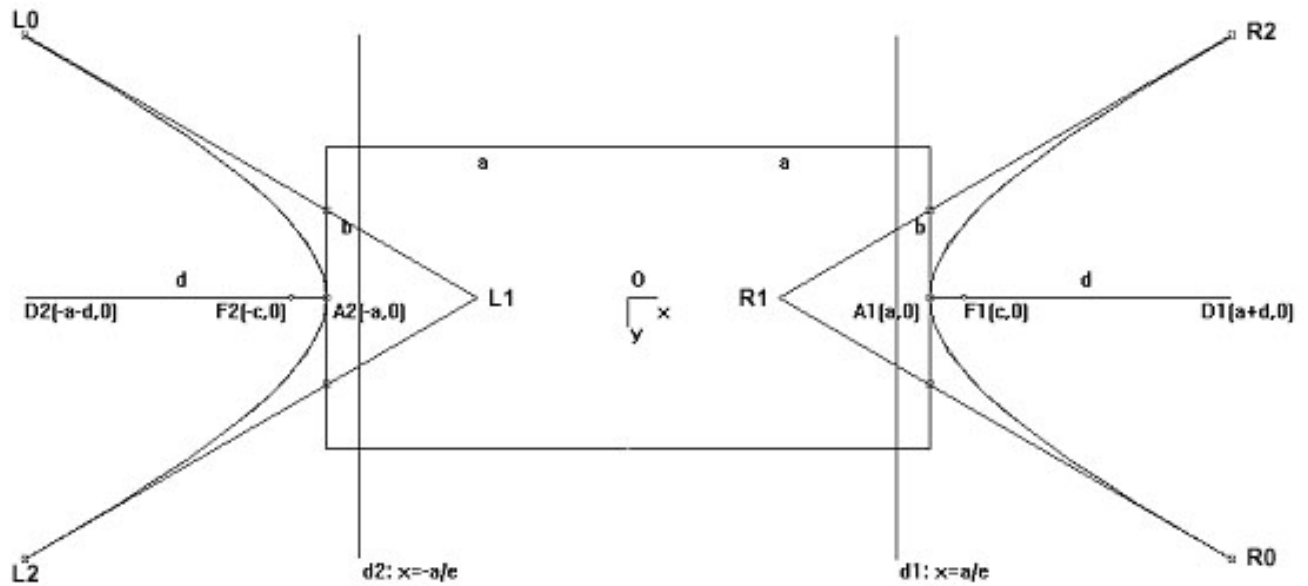
$\alpha \in \left(\left(\frac{3\pi}{2}; 2\pi \right) \cup \left[0; \frac{\pi}{2} \right) \right) \cup \left(\frac{\pi}{2}; \frac{3\pi}{2} \right)$. Следователно $\alpha \in \left(\frac{3\pi}{2}; 2\pi \right) \cup \left[0; \frac{\pi}{2} \right)$ определя

χ_R , а $\alpha \in \left(\frac{\pi}{2}, \frac{3\pi}{2} \right)$ определя χ_L . За да се начертае непрекъснатата дъга от хипербола

следва, че $\alpha_s, \alpha_e \in \left(\frac{3\pi}{2}; 2\pi\right) \cup \left[0; \frac{\pi}{2}\right)$ или $\alpha_s, \alpha_e \in \left(\frac{\pi}{2}; \frac{3\pi}{2}\right)$ (фиг. 2.11). Имайки предвид това, е естествено да дадем следното определение:

Определение 2.3 Дъга от χ_R или χ_L (фиг. 2.10) ще наричаме кривата получена от варирането на параметъра α в интервала $[\alpha_s; \alpha_e]$ от СПУ (2.8) на χ , където $\alpha_s, \alpha_e \in \left(\frac{3\pi}{2}; 2\pi\right) \cup \left[0; \frac{\pi}{2}\right)$ или $\alpha_s, \alpha_e \in \left(\frac{\pi}{2}; \frac{3\pi}{2}\right)$, т.е. α определя непрекъснатата крива от χ_R или χ_L . Дъга от χ_R или χ_L ще бележим с β .

Определение 2.4 Дълбочина на т. $M \in \chi_R$ ($M \in \chi_L$) ще наричаме разстоянието, определено от върха A_1 (A_2) на хиперболата и ортогоналната проекция M' на т. M върху абцисната ос Ox . Дълбочината ще бележим с d (фиг. 2.12).



Фиг. 2.12 Представяне на две дъги от хипербола χ чрез криви на Безие от втора степен.

Лема 2.1 Дълбочината на т. M , получена за стойност на параметъра $\alpha = \alpha^*$, от χ_R или χ_L е $d = \frac{a}{|\cos \alpha^*|} - \text{sign}(\cos \alpha^*)a$.

Доказателство:

1. Ако $M \in \chi_R$ и $\alpha^* \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right) \Rightarrow \text{sign}(\cos \alpha^*) = +1$:

От Определение 2.4 става ясно, че $d = |A_1 M'|$.

1.) Т. $M = \left(\frac{a}{\cos \alpha^*}, b \cdot \text{tg} \alpha^*\right) \Rightarrow$ т. $M' = \left(\frac{a}{\cos \alpha^*}, 0\right) \Rightarrow |OM'| = \left|\frac{a}{\cos \alpha^*}\right|$, но

$$a > 0 \Rightarrow |OM'| = \frac{a}{|\cos \alpha^*|}.$$

2.) $|OA_1| = a$ и $|A_1 M'| = |OM'| - |OA_1| \Rightarrow |A_1 M'| = d = \frac{a}{|\cos \alpha^*|} - (+a)$. Но

$$\text{sign}(\cos \alpha^*) = +1 \Rightarrow d = \frac{a}{|\cos \alpha^*|} - \text{sign}(\cos \alpha^*) a.$$

2. Ако $M \in \chi_L$ и $\alpha^* \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right) \Rightarrow \text{sign}(\cos \alpha^*) = -1$:

От Определение 2.4 става ясно, че $d = |A_2 M'|$.

1.) Т. $M = \left(\frac{a}{\cos \alpha^*}, b \cdot \text{tg} \alpha^*\right) \Rightarrow$ т. $M' = \left(\frac{a}{\cos \alpha^*}, 0\right) \Rightarrow |OM'| = \left|\frac{a}{\cos \alpha^*}\right|$, но

$$a > 0 \Rightarrow |OM'| = \frac{a}{|\cos \alpha^*|}.$$

2.) $|OA_2| = a$ и $|A_2 M'| = |OM'| - |OA_2| \Rightarrow |A_2 M'| = d = \frac{a}{|\cos \alpha^*|} - (-a)$. Но

$$\text{sign}(\cos \alpha^*) = -1 \Rightarrow d = \frac{a}{|\cos \alpha^*|} - \text{sign}(\cos \alpha^*) a.$$

Определение 2.5 Дълбочинно отместване на т. $M \in \chi_R$ ($M \in \chi_L$) получена за

ъгъл $\alpha = \alpha^*$ от СПУ (2.8) на χ , ще наричаме числото $d_{\alpha^*} = \frac{a}{\cos \alpha^*} - \text{sign}(\cos \alpha^*) a$.

Нека първо да разгледаме крилото на хиперболатата χ_R .

Твърдение 2.2 За всяка перпендикулярна на абцисната ос права $g_R: x = a + d$, където $d > 0$ следва, че за дъгата от крилото χ_R , стегната от хордата $R_0 R_2$, с краища

пресечните точки на g_R и χ_R , съществува нейно точно представяне с рационална крива на Безие β_R от втора степен. Хомогенните координати на характеристичните точки R_0, R_1, R_2 на β_R са:

$$R_0^w = \left(a(a+d), b\sqrt{d^2+2ad}, a \right), R_1^w = \left(a^3, 0, (a+d)^2 \right),$$

$$R_2^w = \left(a(a+d), -b\sqrt{d^2+2ad}, a \right), \text{ където } d \text{ е дълбочината на т. } R_0 \text{ и } R_2.$$

Доказателство:

Правата $g_R \perp Ox$, с уравнение $g_R: x = a + d$, където $d \in \mathbb{R}$ и $d > 0$ и нека $g_R \cap Ox = D_1$ (фиг. 2.12). Тогава $D_1 \in Ox$, и $|A_1 D_1| = d$. Необходими са ни три точки в хомогенни координати за рационалното Безие представяне на β_R (според теорията дадена в точка 1.2.2.1., уравнения (1.8) и (2.10)). Нека те бъдат $R_i^w(x_i w_i, y_i w_i, w_i)$, $i = 0, 1, 2$. Правата g_R пресича χ_R в т. $R_0(x_0, y_0) \Rightarrow x_0 = a + d$ и

$$y_0 = \frac{b}{a} \sqrt{x_0^2 - a^2}. \text{ Аналогично определяме и т. } R_2(x_2, y_2) = \left(a + d, -\frac{b}{a} \sqrt{x_0^2 - a^2} \right).$$

След като фиксираме $w_0 = w_2 = 1 \Rightarrow R_0^w = \left(a + d, \frac{b}{a} \sqrt{x_0^2 - a^2}, 1 \right)$ и

$$R_2^w = \left(a + d, -\frac{b}{a} \sqrt{x_0^2 - a^2}, 1 \right). \text{ Остава да намерим } R_1^w. \text{ Нека } g_0 \text{ и } g_2 \text{ са допирателни}$$

към χ_R съответно в т. R_0 и R_2 . В точка 1.2.2.1. е посочено, че ако крива на Безие е от втора степен и е зададена с трите характеристични точки $N_i, i = 0, 1, 2$, то N_1 е пресечна точка на допирателните към кривата в N_0 и N_2 . От тук следва, че R_1 е пресечна точка на g_0 и g_2 . Но χ_R е осево симетрична крива спрямо $Ox \Rightarrow R_1 \in Ox \Rightarrow y_1 = 0$. Допирателната към χ_R в т. $R_0(x_0, y_0)$ има общо уравнение:

$$g_0: \frac{x_0}{a^2} x - \frac{y_0}{b^2} y - 1 = 0 \text{ и } g_0 \text{ трябва да пресечем с оста } Ox, \text{ т.е. } y = 0. \text{ Получаваме:}$$

$$\frac{x_0}{a^2} x_1 - \frac{y_0}{b^2} y_1 - 1 = 0 \Rightarrow x_1 = \frac{a^2}{x_0} \Rightarrow R_1 = \left(\frac{a^2}{x_0}, 0 \right). \text{ Остава да намерим стойността на } w_1$$

- третата хомогенна координата (тегло) на т. R_1^w .

$R_1 \in Ox$, R_2 е симетрична на R_0 спрямо Ox , $|R_0O| = |OR_2|$, следователно от уравнение (2.10) $\Rightarrow \beta_R\left(\frac{1}{2}\right) = A_1(a, 0)$. Но при $t = \frac{1}{2}$ получаваме:

$$\gamma_R\left(\frac{1}{2}\right) = \frac{\sum_{i=0}^2 R_i w_i B_{i,2}\left(\frac{1}{2}\right)}{\sum_{j=0}^2 w_j B_{j,2}\left(\frac{1}{2}\right)}. \quad (2.15)$$

Следователно само за x координатата получаваме:

$$x_\gamma\left(\frac{1}{2}\right) = \frac{\sum_{i=0}^2 x_i w_i B_{i,2}\left(\frac{1}{2}\right)}{\sum_{j=0}^2 w_j B_{j,2}\left(\frac{1}{2}\right)} = a, \quad (2.16)$$

но $w_0 = w_2 = 1.0 \Rightarrow$ уравнение (2.16) е линейно уравнение с неизвестно w_1 .

Получаваме: $w_1 = \frac{x_0 + x_2 - 2a}{2(a - x_1)} = \frac{a + d}{a}$.

Накрая хомогенните координати на R_0^w и R_2^w умножаваме с a , а координатите на R_1^w - с $a(a + d)$.

Намерихме представянето на дясното крило на хипербола в каноничен вид, зададена със специфициращите я параметри a , b и d , чрез рационална крива на Безие от втора степен с три характеристични точки в хомогенни координати.

Заради симетрията на χ относно Oy за представянето на лявото крило χ_L на хиперболата, формулираме и аналогично доказващото се на Твърдение 2.2

Твърдение 2.3 За всяка перпендикулярна на абцисната ос права $g_L: x = -(a + d)$, където $d > 0$ следва, че за дъгата от крилото χ_L , стегната от хордата $L_0 L_2$, с краища пресечните точки на g_L и χ_L , съществува нейно точно представяне с рационална крива на Безие β_L от втора степен. Хомогенните координати на характеристичните точки L_0, L_1, L_2 на β_L са:

$$L_0^w = \left(-a(a + d), b\sqrt{d^2 + 2ad}, a \right), L_1^w = \left(-a^3, 0, (a + d)^2 \right),$$

$$L_2^w = \left(-a(a + d), -b\sqrt{d^2 + 2ad}, a \right), \text{ където } d \text{ е дълбочината на т. } L_0 \text{ и т. } L_2.$$

Твърдение 2.4 За всяка дъга от χ_R или χ_L , определена от два ъгъла α_s и α_e , където $\cos \alpha_s = \cos \alpha_e$ (следователно: 1) $\alpha_s, \alpha_e \in \chi_R$ или $\alpha_s, \alpha_e \in \chi_L$; 2) $d_{\alpha_s} = d_{\alpha_e}$), и параметрите a и b от уравнения (2.8), съществува представяне чрез рационална крива на Безие β от втора степен с трите характеристични точки в хомогенни координати:

$$M_0^w = (a, p.b.\operatorname{tg} \alpha_s, p), M_1^w = (a.p^2.w_1, 0, w_1 = \operatorname{sign}(p)(1 + p^2) - p),$$

$$M_2^w = (a, p.b.\operatorname{tg} \alpha_e, p), \text{ където } p := \cos \alpha_s = \cos \alpha_e.$$

Доказателство:

Необходими са ни три точки в хомогенни координати за рационалното Безие представяне на β с уравнение (2.10). Нека те бъдат $M_i^w(x_i w_i, y_i w_i, w_i)$, $i = 0, 1, 2$. Нека

$$p := \cos \alpha_s = \cos \alpha_e. \text{ Нека } \alpha_s \text{ определя точката } M_0(x_0, y_0) \Rightarrow x_0 = \frac{a}{p} \text{ и}$$

$$y_0 = b.\operatorname{tg} \alpha_s. \text{ Нека } \alpha_e \text{ определя точката } M_2(x_2, y_2) \Rightarrow x_2 = \frac{a}{p} \text{ и } y_2 = b.\operatorname{tg} \alpha_e.$$

$$\text{За нормална параметризация на } \beta \text{ изискваме } w_0 = w_2 = 1 \Rightarrow M_0^w = \left(\frac{a}{p}, b.\operatorname{tg} \alpha_s, 1 \right)$$

$$\text{и } M_2^w = \left(\frac{a}{p}, b.\operatorname{tg} \alpha_e, 1 \right). \text{ Остава да намерим } M_1^w.$$

Нека g_0 и g_2 са допирателни към χ_R съответно в т. M_0 и M_2 . Следователно, M_1 ще е пресечната точка на g_0 и g_2 . От осевата симетрия на χ_R спрямо $Ox \Rightarrow M_1 \in Ox \Rightarrow y_1 = 0$. Допирателната към χ_R в т. $M_0(x_0, y_0)$ има общо уравнение:

$$g_0: \frac{x_0}{a^2}x - \frac{y_0}{b^2}y - 1 = 0 \text{ и пресичаме } g_0 \text{ с оста } Ox: y = 0. \text{ Получаваме:}$$

$$\frac{x_0}{a^2}x_1 - \frac{y_0}{b^2}y_1 - 1 = 0 \Rightarrow x_1 = \frac{a^2}{x_0} \Rightarrow M_1 = \left(\frac{a^2}{x_0}, 0 \right) = (a.p, 0).$$

Търсим стойността на w_1 - третата хомогенна координата на т. M_1^w .

$M_1 \in Ox$, M_2 е симетрична на M_0 спрямо Ox , $|M_0O| = |OM_2|$, следователно

$$\beta\left(\frac{1}{2}\right) = A(\operatorname{sign}(p)a, 0).$$

За $t = \frac{1}{2}$ в уравнение (2.10) получаваме:

$$\beta\left(\frac{1}{2}\right) = \frac{\sum_{i=0}^2 M_i w_i B_{i,2}\left(\frac{1}{2}\right)}{\sum_{j=0}^2 w_j B_{j,2}\left(\frac{1}{2}\right)}. \quad (2.17)$$

Следователно само за x координатата получаваме:

$$x_\beta\left(\frac{1}{2}\right) = \frac{\sum_{i=0}^2 x_i w_i B_{i,2}\left(\frac{1}{2}\right)}{\sum_{j=0}^2 w_j B_{j,2}\left(\frac{1}{2}\right)} = \text{sign}(p)a, \quad (2.18)$$

но $w_0 = w_2 = 1 \Rightarrow$ уравнение (2.18) е линейно уравнение с неизвестно w_1 . Получаваме:

$w_1 = \text{sign}(p)\left(\frac{1}{p} + p\right) - 1$. Накрая всички координати на контролните точки, които

получаваме умножаваме с p . С това Твърдение 2.4 е доказано.

Нека α_s определя точката $P_s(x_s, y_s)$ от χ с радиус-вектор \vec{r}_s и α_e определя точката $P_e(x_e, y_e)$ от χ с радиус-вектор \vec{r}_e .

Определение 2.6 Положителна дъга от χ_R или χ_L ще наричаме кривата получена чрез движението на радиус-вектора \vec{r} в положителна посока от P_s до P_e ($P_s, P_e \in \chi_R$ или $P_s, P_e \in \chi_L$) за $\alpha \in [\alpha_s; \alpha_e]$. Отрицателна дъга от χ_R или χ_L ще наричаме кривата получена чрез движението на радиус-вектора \vec{r} в отрицателна посока от P_s до P_e за $\alpha \in [\alpha_s; \alpha_e]$.

Всяка положителна или отрицателна дъга от χ_R или χ_L е непрекъсната.

Обобщена задача: Непрекъсната дъга χ_β от χ да се представи като NURBS крива с отнапред зададена ориентация на обхождане на точките. Дъгата χ_β да се определя чрез a, b, α_s и α_e .

Твърдение 2.5 За всяка дъга χ_β от χ_R или χ_L (фиг. 2.9), определена от два ъгъла α_B и α_L , където $|\alpha_B| > |\alpha_L|$ (B -Bigger, L -Less), и параметрите a и b от уравнения (2.8), съществува точно представяне чрез NURBS крива γ зададена с

уравнение (2.14) (за степен $p=2$), възлов вектор $U = \{0, 0, 0, 1, 1, 1\}$ и с трите контролни върхове в хомогенни координати:

$$\gamma: \begin{cases} M_0^w = (a, b \cdot \sin \alpha_B, \cos \alpha_B) \\ M_0^{0w} = ((1-t^*) \cdot a + \text{sign}(\cos \alpha_B) \cdot t^* \cdot a \cdot \cos \alpha_B, (1-t^*) \cdot b \cdot \sin \alpha_B, w \cdot \cos \alpha_B), \text{ където} \\ M_0^{1w} = (a, b \cdot \sin \alpha_L, \cos \alpha_L) \end{cases}$$

$$w = (1-t^*) + t^* \cdot (\text{sign}(p_B) \cdot ((1/p_B) + p_B) - 1), \quad p_B = \cos \alpha_B, \quad t^* = \frac{p}{1+p}, \quad p = \sqrt{\frac{|M_0 Q|}{|QM_2|}}.$$

Доказателство:

1.) Нека т. $B = \left(\frac{a}{\cos \alpha_B}, b \cdot \text{tg} \alpha_B \right)$ и т. $L = \left(\frac{a}{\cos \alpha_L}, b \cdot \text{tg} \alpha_L \right)$ са двете крайни точки на дъгата χ_β определени съответно от ъглите α_B и α_L .

2.) Първо ще намерим Безие представянето на дъгата, определена от т. $M_0 = \left(\frac{a}{\cos \alpha_B}, b \cdot \text{tg} \alpha_B \right)$ и т. $M_2 = \left(\frac{a}{\cos(\alpha_B)}, b \cdot (-\text{tg} \alpha_B) \right)$, считайки $M_0 \equiv B$.

3.) От Твърдение 2.4 следва, че може да се намери представянето на тази дъга чрез рационална крива на Безие β от втора степен с трите характеристични точки $M_i^w(x_i \cdot w_i, y_i \cdot w_i, w_i), i=0,1,2$ в хомогенни координати:

$$\beta: \quad M_0^w = \left(\frac{a}{p}, b \cdot \text{tg} \alpha_B, 1 \right), \quad M_1^w = \left(a \cdot p \cdot w_1, 0, w_1 = \text{sign}(p) \frac{x_0 + x_2 - 2a}{2(a + x_1)} \right),$$

$$M_2^w = \left(\frac{a}{p}, b \cdot (-\text{tg} \alpha_B), 1 \right), \text{ където } p = \cos \alpha_B.$$

4.) Търсената в условието дъга χ_β се определя от три точки в хомогенни координати в рационалното NURBS представяне (уравнение (2.14)).

Ако намерим подразделянето на кривата β в т. L , тогава сегментът определен от дъгата $\widehat{M_0 L}$ ще е търсеният. Нужна ни е стойността на t ($t \in [0;1]$), за която се получава т. L . Нека тази стойност да я бележим с t^* .

От [91, 312 с.] знаем, че $\frac{|M_0Q|}{|QM_2|} = \frac{t^{*2}}{(1-t^{*2})}$. Нека $p_B := \cos \alpha_B$, $p_L := \cos \alpha_L$,

$t_B := \operatorname{tg} \alpha_B$ и $t_L := \operatorname{tg} \alpha_L$.

Ако означим с $\rho = \sqrt{\frac{|M_0Q|}{|QM_2|}}$, то $t^* = \frac{\rho}{1+\rho}$. Точката $Q(x_Q, y_Q) \in M_0M_2$ и се явява

пресечна точка на правите $g: x = \frac{a}{\cos \alpha_B}$ и $q: \begin{cases} \text{т. } M_1 = (x_1 = a.p_B, y_1 = 0) \\ \vec{l} = \overrightarrow{M_1L} = (x_l = a/p_L - a.p_B, y_l = b.t_L) \end{cases}$.

Но $M_0M_2 \perp Ox \Rightarrow x_Q = \frac{a}{p_B}$.

$$\text{От } g \cap q = Q \Rightarrow \begin{cases} g: x = \frac{a}{p_B} \\ q: \frac{x - x_1}{x_l} = \frac{y - y_1}{y_l} = \rho \end{cases} \Rightarrow \rho = \frac{a/p_B - x_1}{x_l} \Rightarrow y = \rho.y_l + y_1 \Rightarrow$$

$$\text{т. } Q \begin{cases} x_Q = x = a/p_B \\ y_Q = y = \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L \end{cases}.$$

$$\text{Следователно } |M_0Q| = \left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L - b.t_B \right| \text{ и}$$

$$|M_2Q| = \left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L + b.t_B \right|, \text{ а } \rho = \frac{\left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L + b.t_B \right|}{\left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L - b.t_B \right|} \text{ и}$$

$$t^* = \frac{\rho}{1+\rho} = \frac{\left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L + b.t_B \right|}{2 \cdot \left| \frac{a/p_B - a.p_B}{a/p_L - a.p_B} . b.t_L \right|}.$$

5.) Подразделянето на кривата със стойност на параметъра $t = t^*$ ще даде:

$$\begin{aligned} (M_0^0, w_0^0) &= (1-t^*)(M_0, w_0) + t^*(M_1, w_1) \\ (M_1^0, w_1^0) &= (1-t^*)(M_1, w_1) + t^*(M_2, w_2), \end{aligned} \quad w_0 = 1 \quad \Rightarrow$$

$$M_0^0 = \frac{(1-t^*).M_0 + t^*.M_1.w_1}{(1-t^*) + t^*.w_1}, \quad w_0^0 = (1-t^*) + t^*.w_1; \rightarrow$$

$$M_0^0 = \left(x_0^0 = \frac{(1-t^*).x_0 + t^*.x_1.w_1}{w_0^0}, y_0^0 = \frac{(1-t^*).y_0 + t^*.y_1.w_1}{w_0^0} \right), \text{ но}$$

$$x_0 = \frac{a}{p}, y_0 = b.tg \alpha_B, w_0 = 1, \quad x_1 = a.p, y_1 = 0, w_1 = sign(p_B) \left(\frac{1}{p_B} + p_B \right) - 1 \Rightarrow$$

$$M_0^0 = \left(x_0^0 = \frac{(1-t^*). (a/p_B) + t^*.a.p_B.w_1}{w_0^0}, y_0^0 = \frac{(1-t^*).b.t_B}{w_0^0} \right),$$

$$w_0^0 = (1-t^*) + t^*. (sign(p_B) ((1/p_B) + p_B) - 1),$$

$$M_0^1 = \frac{\sum_{i=0}^2 M_i w_i B_{i,2}(t^*)}{\sum_{j=0}^2 w_j B_{j,2}(t^*)} \text{ и}$$

$$w_0^1 = \sum_{i=0}^2 w_i B_{i,2}(t^*) = (1-t^*)^2 + 2t^*. (1-t^*). (sign(p_B) ((1/p_B) + p_B) - 1) + (t^*)^2.$$

$$\text{Но } M_0^1 \equiv L \text{ и } M_0^{1w} = (x_0^1.w_0^1, y_0^1.w_0^1, w_0^1).$$

Намерихме характеристичните точки M_0, M_0^0, M_0^1 и теглата им w_0, w_0^0, w_0^1 , които в рационалното представяне на Безие изграждат крива, която е идентична с χ_B .

Следователно този сегмент ще има форма-фактор ([91, 241-263 с.]) $\kappa = \frac{w_0.w_0^1}{(w_0^0)^2}$,

където $w_0^0 \neq 1$ и $w_0^1 \neq 1$.

6.) За да получим нормална параметризация ще искаме теглото на т. M_0^1 да е равно на 1 ($w_{M_0^1} = 1$), т.е. ще направим репараметризация на кривата. $M_0^1 \equiv L$

Нека новите тегла на M_0, M_0^0, M_0^1 да бъдат съответно $w_{M_0}, w_{M_0^0}, w_{M_0^1}$, тогава форма-фактора можем да запишем така $\kappa = \frac{w_{M_0} \cdot w_{M_0^1}}{(w_{M_0^0})^2}$, където $w_{M_0} = w_{M_0^1} = 1$.

Следователно
$$\frac{w_0 \cdot w_0^1}{(w_0^0)^2} = \frac{w_{M_0} \cdot w_{M_0^1}}{(w_{M_0^0})^2} \Rightarrow \frac{w_0^1}{(w_0^0)^2} = \frac{1}{(w_{M_0^0})^2}.$$

Избрали сме да работим с положителни тегла, следователно $w_{M_0^0} = \frac{w_0^0}{\sqrt{w_0^1}}.$

7.) Като имаме предвид, че $M_0^1 \equiv L$ и позовавайки се на геометричната характеристика на NURBS кривите (от точка 1.2.4.), че NURBS крива без вътрешни възли е рационална крива на Безие, за дъгата χ_β получаваме, че може да се построи като NURBS крива γ от втора степен с трите контролни върха в хомогенни координати

$$\gamma: \begin{cases} M_0^w = (x_0 = a/p_B, y_0 = b.t_B, w_{M_0} = 1) \\ M_0^{0w} = \left(x_0^0 = \frac{(1-t^*) \cdot (a/p_B) + t^* \cdot a \cdot p_B \cdot w_1}{\sqrt{w_0^1}}, y_0^0 = \frac{(1-t^*) \cdot b.t_B}{\sqrt{w_0^1}}, w_{M_0^0} = \frac{w_0^0}{\sqrt{w_0^1}} \right), \text{ където} \\ M_0^{1w} = (x_0^1 = a/p_L, y_0^1 = b.t_L, w_{M_0^1} = 1) \end{cases}$$

$$w_0^1 = (1-t^*)^2 + 2t^* \cdot (1-t^*) \cdot (\text{sign}(p_B) \cdot ((1/p_B) + p_B) - 1) + (t^*)^2,$$

$$t^* = \frac{\left| \frac{a/p_B - a \cdot p_B}{a/p_L - a \cdot p_B} \cdot b.t_L + b.t_B \right|}{2 \cdot \left| \frac{a/p_B - a \cdot p_B}{a/p_L - a \cdot p_B} \cdot b.t_L \right|} \text{ и } p_B = \cos \alpha_B, p_L = \cos \alpha_L, t_B = \text{tg} \alpha_B, t_L = \text{tg} \alpha_L.$$

След премахване на рационалните изрази получаваме:

$$\gamma: \begin{cases} M_0^w = (a, b.\sin \alpha_B, \cos \alpha_B) & (M_0 \equiv B) \\ M_0^{0w} = ((1-t^*).a + \text{sign}(\cos \alpha_B).t^*.a.\cos \alpha_B, (1-t^*).b.\sin \alpha_B, w.\cos \alpha_B), \text{ където} \\ M_0^{1w} = (a, b.\sin \alpha_L, \cos \alpha_L) & (M_0^1 \equiv L) \end{cases}$$

$$w_0^0 = (1-t^*) + t^* \cdot (\text{sign}(p_B)((1/p_B) + p_B) - 1), \quad p_B = \cos \alpha_B, \quad t^* = \frac{p}{1+p}, \quad p = \sqrt{\frac{|M_0 Q|}{|QM_2|}}.$$

С това Твърдение 2.5 е доказано.

Построяването на NURBS кривата, която представлява дъгата от хиперболата χ следва директно от Твърдение 2.5. Тук обаче възниква въпросът относно построяването на дъга от хипербола чрез NURBS крива, но с произволен брой на сегментите с цел възможност за локална модификация. За решаването на тази задача правим подразделяне на получената NURBS крива с Алгоритъм 2.3.

2.6 Изводи

От направените теоретични разработки в Глава II могат да бъдат направени следните изводи:

1. Предложен е един подход за представяне на дъга от елипса чрез NURBS крива от втора степен. Той позволява построяване на дъги за стойности на параметъра α от уравнение (2.4) по-големи от $|360^\circ|$. Подходът е реализиран в Алгоритъм 2.2, който позволява построяване на кривата с отчитане на ориентацията на нейното обхождане и с произволен брой сегменти, предоставящи възможност за локална модификация. Тези резултати бяха докладвани и обсъждани на международната конференция “Автоматика и информатика”, София (2005 г.).

2. Реализирано е представяне на дъга от парабола чрез NURBS крива от втора степен, което представяне следва директно от Твърдение 2.1. Предимството на това преставане е, че стойностите на координатите на контролните върхове и техните тегла се пресмятат по формули, т.е. избягва се алгоритмичното изчисляване (както е в методите на Пигъл, на които беше направен анализ в раздел 1.4). Предложен е алгоритъм (Алгоритъм 2.3) за подразделяне на NURBS кривата, предоставящо възможност за локална модификация. Получените резултати бяха докладвани на международната конференция “Автоматика и информатика”, София (2005 г.).

3. Дадено е решение на задачата за представяне на дъга от хипербола чрез NURBS крива от втора степен, което представяне следва директно от Твърдение 2.5. Предимството на това преставане е, че стойностите на координатите на контролните

върхове и техните тегла се изчисляват по формули, т.е. избягва се алгоритмичното пресмятане (както е в методите на Пигъл). Възможността за локална модификация на кривата се постига чрез прилагането на Алгоритъм 2.3. Подходът за това представяне е докладван на Деветнадесетата международна конференция “SAER”, Варна (2005 г.)

4. Всяка една от разгледаните криви (дъги от елипса, парабола и хипербола) е представена чрез NURBS крива от втора степен с уравнение от вида:

$$\gamma(t) = \sum_{i=0}^n R_{i,2}(t)P_i, \quad t \in [0;1], \quad (2.19)$$

където P_i са $n+1$ контролни върха, с тегла w_i , възлов вектор $T = \{t_0 = 0, 0, 0, \tau_1, \tau_1, \tau_2, \tau_2, \dots, \tau_m, \tau_m, 1, 1, t_{n+3} = 1\}$ и състояща се от $m+1$ на брой сегменти. При промяна на позицията на контролен връх P_i или стойността на неговото тегло, кривата се променя само за $t \in [t_i, t_{i+3}]$, $i = 0, \dots, n$.

Представянето на кониките с криви от този вид води след себе си всички предимства на NURBS кривите. Освен това, представената по този начин NURBS крива, позволява модифицирането в отделни нейни сегменти да става с изключителна лекота. Представянето на трите типа криви в един и същ модел прави моделирането с тях еднотипно и интуитивно. Това повишава лекотата при работа с тях, следствие от което се повишава и производителността. Предложените NURBS модели на квадратичните криви са докладвани на международната конференция “Doctoral-Students Conference”, Гърция (2007 г.) и на международната конференция “UniTech”, Габрово (2004 г.).

Глава трета

ПРЕДСТАВЯНЕ НА ПОВЪРХНИНТЕ ОТ ВТОРА СТЕПЕН ЧРЕЗ NURBS ПОВЪРХНИНИ

Разработването на нови 3D графични примитиви за моделиране с реалните неразпадащи се повърхнини от втора степен (елипсоид, параболоид, хиперболоид, двоен хиперболоид, конус, елиптичен цилиндър, параболичен цилиндър, хиперболичен цилиндър или хиперболичен параболоид), налага тяхното представяне чрез подходящ модел на повърхнини от ГМ. В първа глава, като най-подходящ модел, беше посочен NURBS, а във втора глава беше направено представяне на кривите от втора степен чрез квадратични (подразделени) NURBS криви. В тази глава ще направим представяне на повърхнините от втора степен чрез NURBS повърхнини.

Определянето на повърхнината за построяване ще извършим въз основа на стойностите на параметрите от скаларно-параметричните ѝ уравнения. Затова първо ще бъдат представени аналитичните задавания на квадратичните повърхнини, а след това и техните представяния в NURBS.

3.1. Аналитично представяне на повърхнините от втора степен

Аналитичното дефиниране на една повърхнина от втора степен е добре известно [33]:

Дефиниция 3.1 Под повърхнина от втора степен се разбира всяка съвкупност от точки в пространството, която относно дясна декартова координатна система $K = \{O, \vec{e}_1, \vec{e}_2, \vec{e}_3\}$ се задава с уравнение от втора степен.

Така общото уравнение на една повърхнина S от втора степен е

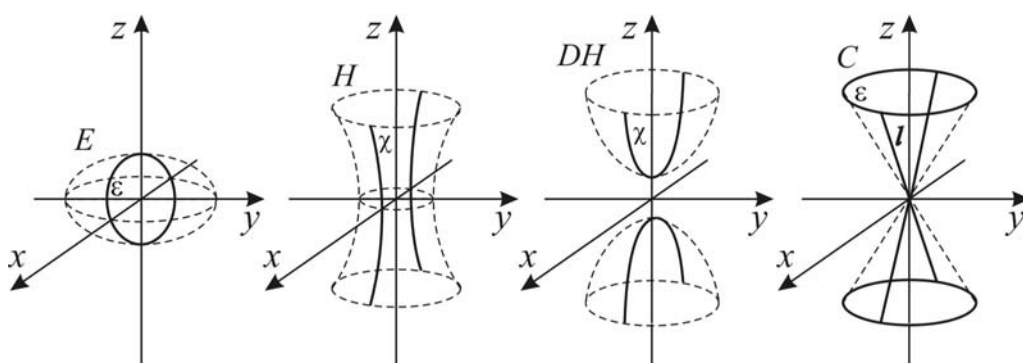
$$S: a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{23}yz + 2a_{13}xz + 2b_1x + 2b_2y + 2b_3z + c = 0, \quad (3.1)$$

където $a_{ij}, b_i, c \in \mathbb{R}$.

В [33, 145-155 с.] е дадена една схема за метрична класификация на повърхнини от втора степен, която копира тази за криви от втора степен. Чрез описаният метод се получават всички движения в пространството, чрез които се преминава към канонична за повърхнината декартова координатна система. Добре известно е, че съществуват седемнадесет типа метрично нееквивалентни повърхнини от втора степен, разделени в три групи: *централни повърхнини (с един център на симетрия)*; *повърхнини, имащи прави от центрове на симетрия*; *повърхнини, имащи равнина от центрове на*

симетрия. За целите на ГМ ще разглеждаме само реалните такива, които не се разпадат на равнини. Това са следните девет повърхнини:

- централните повърхнини елипсоид (E), хиперболоид (прост, с една повърхнина) (H), двоен хиперболоид (DH), конус (C) (фиг. 3.1);
- повърхнините елиптически цилиндър (EC), хиперболически цилиндър (HC), имащи права от центрове на симетрия (фиг. 3.2);
- нецентралните повърхнини параболически цилиндър (PC), елиптически параболоид или само параболоид (P), хиперболически параболоид (HP) (фиг. 3.3).



Фиг. 3.1 E – елипсоид; H – хиперболоид; DH – двоен хиперболоид; C – конус.

Каноничното уравнение на елипсоида е:

$$E: \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (3.2)$$

където числата $a, b, c: a \geq b \geq c > 0$ се наричат полуоси на елипсоида. При $a = b = c = R$ елипсоидът E се трансформира в сфера S с център $O(0,0,0)$ и радиус R . Каноничното уравнение на сферата е:

$$S: x^2 + y^2 + z^2 = R^2. \quad (3.3)$$

Едни СПУ на сферата се дават с:

$$S: \begin{cases} x = R \cos u \cos v \\ y = R \cos u \sin v \\ z = R \sin u \end{cases}, \quad u \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right], \quad v \in [0; 2\pi]. \quad (3.4)$$

По-точно, елипсоидите и сферите са афинно-еквивалентни повърхнини при афинната трансформация δ , композирана от три дилатации: по Ox , Oy , Oz съответно с коефициенти $\frac{a}{R}$, $\frac{b}{R}$, $\frac{c}{R}$:

$$\delta: x = \frac{a}{R}X, y = \frac{b}{R}Y, z = \frac{c}{R}Z \Rightarrow E \xrightarrow{\delta} S: x^2 + y^2 + z^2 = R^2. \quad (3.5)$$

Скаларно-параметричните уравнения на елипсоида са:

$$E: \begin{cases} x = a \cos u \cos v \\ y = b \cos u \sin v \\ z = c \sin u \end{cases}, \quad u \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right], \quad v \in [0; 2\pi]. \quad (3.6)$$

Каноничното уравнение на хиперболоида е:

$$H: \frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1. \quad (3.7)$$

В [33] могат да се видят едни СПУ на хиперболоида. По-нататък за нашите цели ще бъдат използвани следните СПУ:

$$H: \begin{cases} x = \frac{a}{\cos u} \cdot \cos v \\ y = \frac{b}{\cos u} \cdot \sin v \\ z = c \cdot \operatorname{tgu} \end{cases}, \quad u \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right), \quad v \in [0; 2\pi]. \quad (3.8)$$

Каноничното уравнение на двойния хиперболоид е:

$$DH: \frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = -1. \quad (3.9)$$

Равенствата

$$DH: \begin{cases} x = a \cdot \operatorname{tgu} \cdot \cos v \\ y = b \cdot \operatorname{tgu} \cdot \sin v \\ z = c \cdot \cos u \end{cases}, \quad u \in \left[0; \frac{\pi}{2}\right), \quad v \in [0; 2\pi] \quad (3.10)$$

са едни СПУ на двойния хиперболоид.

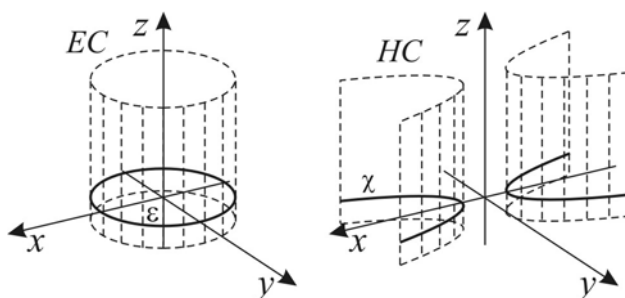
Каноничното уравнение на конуса е:

$$C: \frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0. \quad (3.11)$$

Равенствата

$$C: \begin{cases} x = a.u.\cos v \\ y = b.u.\sin v, \quad u \in [-\infty; \infty], v \in [0; 2\pi] \\ z = c.u \end{cases} \quad (3.12)$$

са СПУ на C .



Фиг. 3.2 EC – елиптически цилиндър; HC – хиперболически цилиндър; и управителни криви ε – елипса и χ – хипербола.

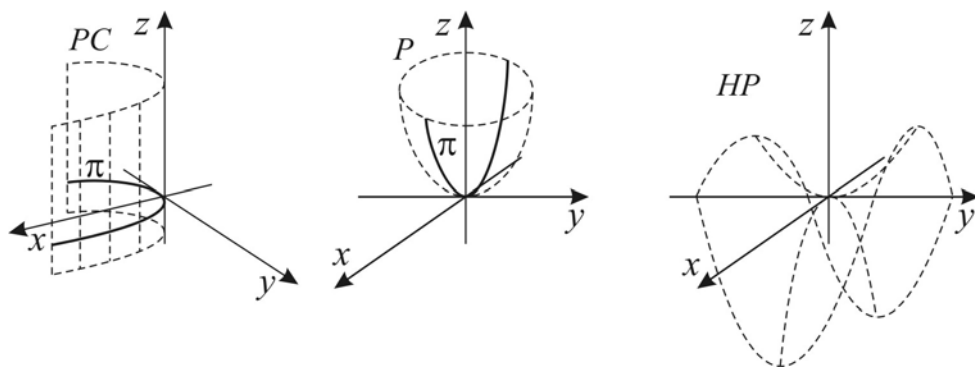
Правите цилиндри от втора степен, с образувателни успоредни на Oz и управителни криви – криви от втора степен, са EC (елиптичен цилиндър), HC (хиперболически цилиндър) и PC (параболически цилиндър) и техните канонични уравнения са:

$$EC: \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad (3.13)$$

$$HC: \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \quad (3.14)$$

$$PC: y^2 = 2px. \quad (3.15)$$

За EC и HC върху Oz лежат всичките им центрове на симетрия, а PC е нецентрална повърхнина. Едни техни СПУ са СПУ за x, y на съответната управителна крива и свободната текуща z - втори параметър.



Фиг. 3.3 *PC* – параболичен цилиндър и управителна линия π - парабола;
P – параболоид и образувателна линия π - парабола;
HP – хиперболичен параболоид.

Каноничното уравнение на елиптическия параболоид е:

$$P: \frac{x^2}{a^2} + \frac{y^2}{b^2} = 2z. \quad (3.16)$$

СПУ на елиптическия параболоид са:

$$P: \begin{cases} x = a.u.\cos v \\ y = b.u.\sin v \\ z = \frac{1}{2}.u^2 \end{cases}, \quad u \in [-\infty; \infty], v \in [0; 2\pi]. \quad (3.17)$$

Елиптическият параболоид е неограничена повърхнина, разположена над координатната повърхнина Oxy , т.е. в обединението на четирите октанта, за които $z \geq 0$.

Хиперболичният параболоид има канонично уравнение:

$$HP: \frac{x^2}{a^2} - \frac{y^2}{b^2} = 2z. \quad (3.18)$$

СПУ на *HP* се получават чрез основното тъждество за хиперболичните функции и едни от тях са:

$$HP: \begin{cases} x = a.u.\operatorname{ch} v \\ y = b.u.\operatorname{sh} v \\ z = \frac{1}{2}.u^2 \end{cases} \cup \begin{cases} x = a.u.\operatorname{sh} v \\ y = b.u.\operatorname{ch} v \\ z = -\frac{1}{2}.u^2 \end{cases}, \quad u, v \in [-\infty; \infty]. \quad (3.19)$$

Хиперболичният параболоид е неограничена повърхнина.

Квадратичните повърхнини включват класовете от ротационни и праволинейни такива. В [33, 172-182 с.] са характеризирани праволинейните и ротационните повърхнини от втора степен. Тук цитираме само двете заключителни теореми:

Теорема 3.1 Реалните, ротационни повърхнини от втора степен с ротационни оси Oz са: сфера (елипсоид при $a = b = c$); елипсоид при $a = b$; елиптически параболоид при $a = b$; хиперболоид при $a = b$; двоен хиперболоид при $a = b$; конус при $a = b$; елиптически цилиндър при $a = b$.

Теорема 3.2 Реалните, праволинейни повърхнини от втора степен са: конус; елиптически цилиндър; параболически цилиндър; хиперболически цилиндър; хиперболоид; хиперболически параболоид.

След като се изследват сеченията на всяка една от повърхнините с всяка от равнините

$$\alpha_t : z = t (\parallel Oxy), \quad \beta_p : y = p (\parallel Oxz), \quad \gamma_q : x = q (\parallel Oyz),$$

се прави заключение за това как те разсложават (изграждат) съответната повърхнина. Главната цел е да се открият подходящите *генератори* – линии, с помощта на които да се изграждат повърхнините (*генераторни линии*). Установихме, че всички случаи на повърхнините споменати в Теорема 3.1 и Теорема 3.2, с изключение на хиперболическия параболоид (HP), имат генератори, т.е. могат да бъдат построени чрез ротационно или праволинейно движение на генератор.

Следователно, ако съществуват методи за построяване на ротационна и праволинейна NURBS повърхнина на базата на генератор (NURBS крива), тогава ще е възможно да бъдат построени осемте повърхнини (частните им случаи) посочени в Теорема 3.1 и Теорема 3.2 като изключение се явява HP .

3.2. Представяне на повърхнините от втора степен чрез биквадратни NURBS повърхнини

Повърхнините от втора степен разделяме условно в три групи по отношение на методите за тяхното представяне чрез NURBS повърхнини:

- елипсоид, параболоид, хиперболоид, двоен хиперболоид и конус;
- цилиндри: елиптически цилиндър, параболически цилиндър и хиперболически цилиндър;
- хиперболически параболоид.

В Глава II разработихме подходи за представяне на дъги от елипса, парабола и хипербола чрез NURBS криви от втора степен. Представени са алгоритми, които на базата на подадените стойности на параметрите от скаларно-параметричните

уравнения на кривите (уравнения 2.4, 2.6 и 2.8), директно се конструира подразделена на желания брой сегменти NURBS крива от втора степен. Следователно, тези NURBS криви можем да използваме за генератори за да построим чрез тях NURBS повърхнините.

Повърхнините разглеждаме в дясна декартова координатна система $K = \{O, \vec{e}_1, \vec{e}_2, \vec{e}_3\}$ ($K = Oxyz$) с представянията им в каноничен вид и в тази координатна система търсим представянията им в NURBS модела.

3.2.1. Представяне на елипсоид, параболоид, хиперболоид, двоен хиперболоид и конус чрез биквадратни NURBS повърхнини

Тук ще бъде направено представянето на елипсоид, параболоид, хиперболоид, двоен хиперболоид и конус чрез NURBS повърхнини. Общото между тези повърхнини е, че техни частни случаи могат да се построят чрез ротация на генератор. Тези частни случаи са: елипсоид (E) при $a = b$ и $a = b = c$ (сфера (S)); параболоид (P) при $a = b$; хиперболоид (H) при $a = b$; двоен хиперболоид (DH) при $a = b$ и конус (C) при $a = b$. Тези повърхнини са ротационни (Теорема 3.1).

Повърхнините могат да бъдат определяни за построяване чрез два параметъра u и v , които да се изменят в определени интервали $u \in [\bar{u}_s; \bar{u}_e]$, $v \in [\bar{v}_s; \bar{v}_e]$. За всяка една от повърхнините, допустимите стойности на тези параметри са различни. Построяването на повърхнината за фиксирани стойности \bar{u}_s , \bar{u}_e , \bar{v}_s и \bar{v}_e ще дава като резултат (в общия случай) парче от нея, т.е. някаква част от онази повърхнина, която се получава при цялостното вариране на параметрите в диапазоните им дадени заедно с нейните СПУ.

Чрез сечението на повърхнините с равнината $\beta_p : y = p$ ($\parallel Oxz$) се получават генераторите. Подробно изследване относно тези сечения може да се види в [33]. Тук даваме уравненията (от (3.20) до (3.24)), които се получават като възможни генератори за разглежданите повърхнини.

За сечението $\varepsilon^E = E \cap \beta_p$, при $p = 0$ се получава уравнението:

$$\varepsilon^E : \frac{x^2}{a^2} + \frac{z^2}{c^2} = 1. \quad (3.20)$$

Сечението $\pi^{EP} = EP \cap \beta_p$, при $p = 0$ са параболи с уравнение:

$$\pi^{EP} : x^2 = 2a^2 z \quad (3.21)$$

разположени в $\beta_{p=0}$ по направление $\parallel Oz^+$.

При изследване на сечението $\chi^H = H \cap \beta_p$, при $p = 0$ се получават хиперболи с уравнение:

$$\chi^H : \frac{x^2}{a^2} - \frac{z^2}{c^2} = 1. \quad (3.22)$$

Сечението $\chi^{DH} = DH \cap \beta_p$, при $p = 0$ са хиперболите:

$$\chi^{DH} : \frac{z^2}{c^2} - \frac{x^2}{a^2} = 1. \quad (3.23)$$

За сечението $l^K = K \cap \beta_p$, при $p = 0$ сеченията l^K са двойките пресекателни прави

$$l_{1,2}^K : z = \pm \frac{c}{a} x. \quad (3.24)$$

Първо ще бъде направено NURBS представяне на ротационните повърхнини, които се явяват частни случаи (Теорема 3.1) на разглежданите тук повърхнини и след това ще бъде доказано, че лесно могат да се получат и общите случаи. Генераторите (3.20)÷(3.24) са разположени в равнината Oxz . Ротационните NURBS повърхнини ще бъдат построени чрез ротация на генераторите около Oz от ъгъл α_s до ъгъл α_e . Всеки един от генераторите (от (3.20) до (3.24)) може да бъде представена чрез NURBS крива от втора степен чрез един от подходите дадени в Глава II.

3.2.1.1. Построяване на ротационна биквадратна NURBS повърхнина

Нека $\gamma(v) = \sum_{j=0}^m R_{j,2}(v) P_j$ да бъде NURBS крива от втора степен (уравнение 1.17)

дефинирана върху възлов вектор V , която ще се завърта около ос. Избираме $\gamma(v)$ да лежи в Oxz равнината, т.е. да е равнинната крива:

$$\gamma(v) : \begin{cases} x(v) = \sum_{j=0}^m R_{j,2}(v) x_j \\ y(v) = 0 \\ z(v) = \sum_{j=0}^m R_{j,2}(v) z_j \end{cases}, v \in [0;1] \quad (3.25)$$

с $m+1$ контролни върхове $P_j(x_j, y_j, z_j)$ с тегла w_j и възлов вектор

$V = \{0, 0, 0, v_1, v_1, v_2, v_2, \dots, v_k, v_k, 1, 1, 1\}$. Кривата $\gamma(v)$ ще завъртаме около оста Oz .

Означаваме повърхнината, която се получава чрез $S(u, v)$. Тогава $S(u, v)$ има следните характеристики: 1) за фиксирано \bar{u} , $S(\bar{u}, v)$ е кривата $\gamma(v)$ завъртяна на някакъв ъгъл около Oz оста; 2) за фиксирано \bar{v} , $S(u, \bar{v})$ е дъга от окръжност, която лежи в равнина перпендикулярна на Oz оста и с център на Oz оста.

Възловият вектор U ще е от вида $U = \{0, 0, 0, u_1, u_1, u_2, u_2, \dots, u_h, u_h, 1, 1, 1\}$ и теглата ще са $w_i = \left\{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, \dots, 1, \frac{\sqrt{2}}{2}, 1\right\}$, за $i = 0, \dots, n$. Теглата $w_i = \frac{\sqrt{2}}{2}$ за $i = 1, 3, \dots, n-1$,

защото това е стойността на теглото на междинната характеристична точка при представянето на дъга от окръжност (раздел 1.4, уравнение 1.29). Тогава търсената повърхнина се задава с векторната функция:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}. \quad (3.26)$$

Възловите вектори са U и V . Контролните върхове и теглата им се определят както следва: За $i = 0$, $P_{i,j} = P_{0,j} = P_j$. Всички контролни върхове $P_{i,j}$ за фиксирано j , лежат в равнината $z = z_j$. Теглата се определят чрез произведението на w_j с кръговите

тегла w_i , т.е. за фиксирано j , $w_{0,j} = w_j$, $w_{1,j} = \frac{\sqrt{2}}{2} w_j$, $w_{2,j} = w_j$, $w_{3,j} = \frac{\sqrt{2}}{2} w_j$,

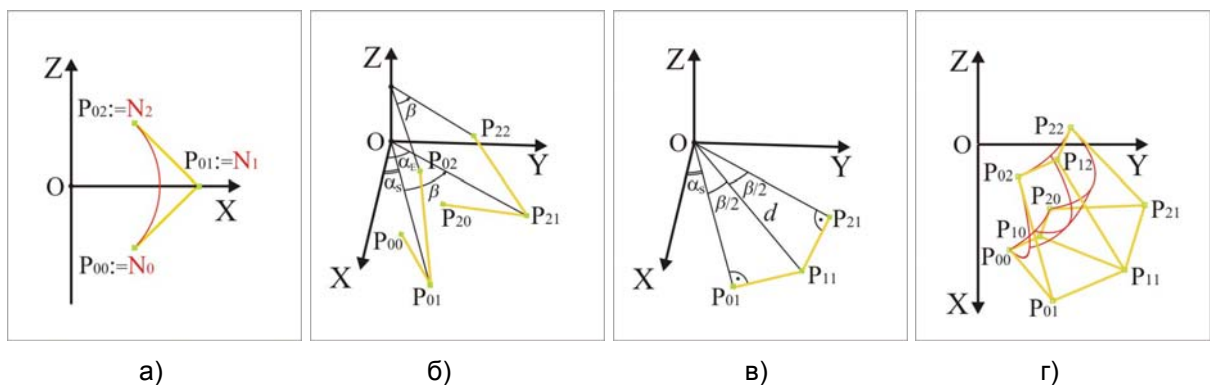
$w_{4,j} = w_j, \dots, w_{n-1,j} = \frac{\sqrt{2}}{2} w_j, w_{n,j} = w_j$.

Локална модификация: При промяна на позицията на контролен връх $P_{i,j}$ или стойността на неговото тегло $w_{i,j}$, формата на повърхнината се променя само в правоъгълника $[u_i, u_{i+3}) \times [v_j, v_{j+3})$.

По този метод е разработен Алгоритъм 3.1, който построява биквадратна NURBS повърхнина по подадени данни за NURBS крива от втора степен и начален ъгъл α_s и краен ъгъл α_e на завъртане.

Нека генераторната NURBS крива да е зададена с nArcs на брой дъги, numCPs на брой контролни върхове съхранени в едномерния масив N и с nKnots на брой възли дадени в едномерния масив K, които се подават на алгоритъма. Трябва да бъде

подаден и броя seg_rev на сегментите, на които ще бъде подразделена повърхнината по направлението v (по което извършваме завъртането). Контролните върхове на повърхнината, която алгоритъмът построява, ще бъдат записани в двумерния масив P . Тази повърхнина ще има u_seg на брой сегмента по u и v_seg на брой сегмента по v . Функцията `ComputeTheCVsNumAndKnotVectors` изчислява броя на контролните върхове по двете направления u и v в променливите u_CVs и v_CVs , броя на възлите по двете направления в u_Knots и v_Knots , и самите възли в едномерните масиви U и V .



Фиг. 3.4 Построяване на ротационна NURBS повърхнина.

За да бъде алгоритъма по-ясен, нека да вземем за пример конструирането на един сегмент от ротационната повърхнина. Ето стъпките, които следваме съгласно (фиг. 3.4):

1. Полагаме $\theta = |\alpha_s - \alpha_e|$, $\beta = \theta / v_seg$, $m = numCPs$.
2. $P_{0j} = N_j$, за $j = 1, \dots, m$ (фиг. 3.4а).
3. Завъртаме P_{0j} на ъгъл α_s (фиг. 3.4б).

P_{2j} се получават чрез ротацията на P_{0j} на ъгъл β , за $j = 1, \dots, m$ (фиг. 3.4в).

Ротацията поражда криви по направлението v , които са части от окръжности. Следователно, както при построяването на дъга от окръжност чрез квадратична NURBS крива, контролните върхове не лежащи на кривата ще имат тегла $w = \cos(\beta/2)$ (раздел 1.4, уравнение 1.29).

4. За точките P_{1j} , $j = 1, 2, \dots, m$, знаем, че:

- Отсечките OP_{1j} разделят ъглите $\angle P_{0j}OP_{2j}$ на равни части, за $j = 1, \dots, m$.

Но $\angle P_{0j}OP_{2j} = \beta$ и следователно $\angle P_{0j}OP_{1j} = P_{1j}OP_{2j} = \beta/2$.

- Ъглите $\angle OP_{0j}P_{1j} = \angle OP_{2j}P_{1j} = 90^\circ$, защото $P_{0j}P_{1j}$ и $P_{2j}P_{1j}$ са допирателни към повърхнината по направление на ротацията.

5. Полагаме $\delta_j = |OP_{0j}| = |OP_{2j}|$. Тогава $d_j = |OP_{1j}| = \delta_j / \cos(\beta/2)$. Нека също $\theta_{buf} = \alpha_s + \beta/2$.

6. Изчисляваме (фиг. 3.4г):

$$P_{1j}.x = d * \cos(\theta_{buf})$$

$$P_{1j}.y = d * \sin(\theta_{buf})$$

$$P_{1j}.z = P_{0j}.z$$

$$P_{1j}.w = w * P_{0j}.w$$

Алгоритъм 3.1 ConstructBiquadraticRevolvedSurface

Вход: N, arcs, nCPs, alphas, alphas, seg_rev;

Изход: P, u_seg, v_seg;

Процедура:

- 1.) Определя се ъгъла на завъртане на кривата и се осигурява правилна посока на завъртане.
- 2.) NURBS кривата се поставя в началния ъгъл.
- 3.) Конструира се повърхнината като се използва формулата за ротация на точка в пространството около оста Oz и метода за изчисляване на междинните точки и теглата им на всеки сегмент даден по-горе.
- 4.) Край.

Псевдопрограмният код е:

```

1) ConstructBiquadraticRevolvedSurface(N, arcs, nCPs, alphas, alphas,
    seg_rev, P, u_seg, v_seg) {
2) thita = abs(alphas - alphas);
3) th = thita; thitabuf = 0.0; u_seg = arcs; v_seg = 0;
4) while( th > 0.0 ) do {v_seg = v_seg+1; th = th-pi/2; }
5) if( v_seg < seg_rev ) then v_seg = seg_rev;
6) beta = thita/v_seg;
7) if( alphas < alphas ) then beta = -beta;
8) m = nCPs;
9) for j in 1 to m do {
    a. P[1][j] = N[j];
    b. xbuf = P[1][j].x*cos(alphas) - P[1][j].y*sin(alphas);
    c. ybuf = P[1][j].x*sin(alphas) + P[1][j].y*cos(alphas);
    d. P[1][j].x = xbuf; P[1][j].y = ybuf; }

```

```

10) w1 = cos(beta/2); index = 1; d = 0.0;
11) for k in 1 to v_seg do {
12)   for j in 1 to m do {
        a. xbuf = P[index][j].x*cos(beta) - P[index][j].y*sin(beta);
        b. ybuf = P[index][j].x*sin(beta) + P[index][j].y*cos(beta);
        c. P[index+2][j].x = xbuf; P[index+2][j].y = ybuf;
        d. P[index+2][j].z = P[index][j].z;
        e. P[index+2][j].w = P[index][j].w;
        f. xbuf = P[index][j].x; ybuf = P[index][j].y;
        g. d = sqrt(xbuf*xbuf + ybuf*ybuf)/cos(beta/2);
        h. thitabuf = alphas + (k-1)*beta + beta/2;
        i. P[index+1][j].x = d * cos( thitabuf );
        j. P[index+1][j].y = d * sin(thitabuf);
        k. P[index+1][j].z = P[index][j].z;
        l. P[index+1][j].w = w1*P[index][j].w; }
13)   index = index + 2; }
14) }

```

Коментари по алгоритъм 3.1:

1. На редове от 2) до 7) се определя ъгъла на завъртане на кривата и се осигурява правилна посока на завъртането ѝ. Ъгълът θ може да бъде по-голям от 360° по абсолютна стойност. В този случай повърхнината ще се получи чрез многократно завъртане. Това определя и стойността на v_seg – броя на сегментите по направление v , който може да е по-голям от 4.

2. Чрез цикъла на ред 9) кривата се поставя в началната позиция, от която започва построяването на повърхнината.

3. С циклите на редове 11) и 12) се намират контролните върхове и техните тегла определящи NURBS повърхнината. Изчисляват се точките на v_seg на брой сегменти на повърхнината по направление v .

4. След изпълнението на Алгоритъм 3.1 получаваме контролните върхове на NURBS повърхнината в P , подразделена на u_seg сегменти по u направление и v_seg сегменти по v направление.

5. Коректността на Алгоритъм 3.1 се определя от коректността на изложения по-горе подход за построяване на ротационна биквадратна NURBS повърхнина.

С помощта на следващата функция се изчислява: броя на контролните върхове по двете направления u и v на получената повърхнина в променливите u_CVs и v_CVs ; броя на възлите по двете направления в u_Knots и v_Knots и самите възли в едномерните масиви U и V .

Функция ComputeTheCVsNumAndKnotVectors:


```

ComputeTheCVsNumAndKnotVectors(u_seg,v_seg,u_CVs,v_CVs,u_Knots,
    v_Knots,U,V) {
    u_CVs = 2*u_seg+1; u_Knots = k = nKnots;
    for i in 1 to k do U[i] = K[i];
    v_CVs = 2*v_seg+1; v_order = 3; v_Knots = k = v_CVs+v_order;
    step = 1.0; g = 0.0;
    if( k > 6 ) then step = 1.0/((k-6)/2.0+1.0);
    i = 4;
    while( i <= k-3 ) do { g = g+step; V[i] = V[i+1] = g; i = i+2; }
    i = k;
    while( i >= k - 2 ) do { V[i] = 1.0; i = i-1; }
}

```

3.2.1.2. Построяване на NURBS елипсоид-повърхнина (NE-повърхнина)

Задачата за построяване на повърхнината на елипсоид можем да формулираме така:

Задача 3.1 Да се построи повърхнина на елипсоид по зададени стойности на параметрите (полуоси) a, b, c и зададени интервали за u и v като $u \in [u_s, u_e]$, $u_s, u_e \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$ и $v \in [v_s, v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$ (от уравнение (3.6)),

представена чрез биквадратна NURBS повърхнина.

Използваме следното добре известно твърдение:

Лема 3.1 Елипсоидът E зададен с уравнение (3.6) може да бъде получен от

единичната сфера S_1 :
$$\begin{cases} x^{S_1} = \cos u \cos v \\ y^{S_1} = \cos u \sin v \\ z^{S_1} = \sin u \end{cases}, \quad u \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right], \quad v \in [0; 2\pi]$$
 мащабирана с

матрицата на афинната трансформация
$$T = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}.$$

Ако k е равнинна крива в координатната равнина $Oxz: y=0$, зададена със СПУ

$k: \begin{cases} x = f(u) \\ z = g(u) \end{cases}, u \in J \subseteq \mathbb{R}$, като f и g са непрекъснати функции в J и $f: J \rightarrow f(J)$ е

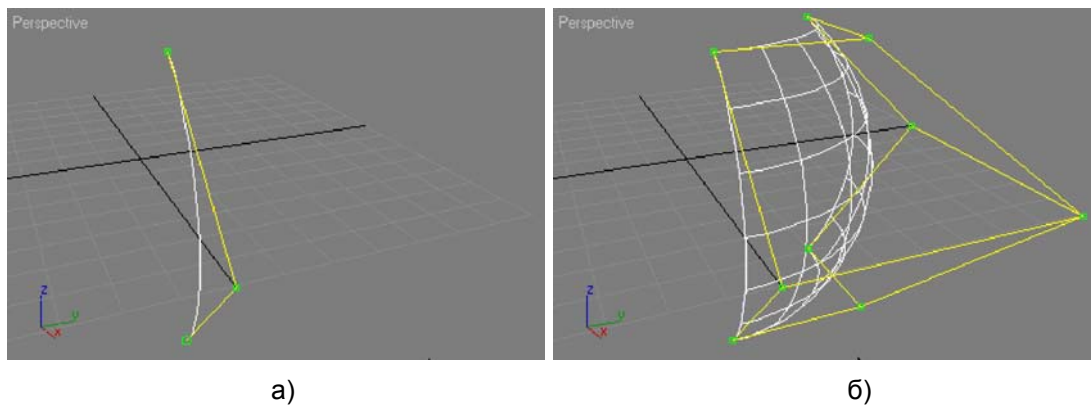
неотрицателна, обратима върху $f(J)$. Тогава (според Теорема 24.1 в [33]) ротационната повърхнина има скаларно-параметрични уравнения:

$$S_{(k,Oz)}: \begin{cases} x = f(u) \cos v \\ y = f(u) \sin v, \quad u \in J, v \in [0; 2\pi]. \\ z = g(u) \end{cases}$$

Следователно, сферата S_1 е ротационна повърхнина с генератор равнинната крива $k: \begin{cases} x = \cos(u) \\ z = \sin(u) \end{cases}, u \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$ в $Oxz: y=0$. За да получим единичната сфера S_1 , можем да намерим генератора – дъгата от окръжността лежаща в $Oxz: y=0$ за $u \in [u_s, u_e]$, $u_s, u_e \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right]$, която да завъртим от ъгъл v_s до ъгъл v_e . За стойности на v извън интервала $[0; 2\pi]$, точките, които ще получаваме също ще принадлежат на тази повърхнина. Следователно за изграждането на NURBS повърхнината можем да допуснем $v \in [v_s; v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$. Повърхнината да се изгражда непрекъснато чрез необходимия брой завъртания на генератора.

В Глава II представихме Алгоритъм 2.1 за построяване на равнинна NURBS дъга от единичната окръжност. Чрез него построяваме дъгата от единичната сфера в равнината $Oxz: y=0$ и му подаваме следните стойности за параметрите: $a=b=1$, $\alpha_s = u_s$ и $\alpha_e = u_e$, и с желан брой на подразделяне на NURBS кривата. Получаваме генераторната NURBS крива K_N с контролни върхове N_j (фиг. 3.5a):

$$K_N(v) = \sum_{j=0}^m R_{j,2}(v) N_j.$$



Фиг. 3.5 Построяване на един сегмент от единичната сфера.

Построяваме NURBS повърхнината $E_p(u, v)$ чрез завъртане на K_N от ъгъл v_s до

ъгъл v_e , като прилагаме метода даден в точка 3.2.1.1. (реализиран с Алгоритъм 3.1). Контролните върхове на E_p бележим с $P_{i,j}$. Тази повърхнина е точно парче от единичната сфера представено като биквадратна NURBS повърхнина (фиг. 3.5б):

$$E_p(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}.$$

Накрая, позовавайки се на Лема 1, извършваме мащабиране на $E_p(u, v)$ с коефициенти на мащабиране a, b, c :

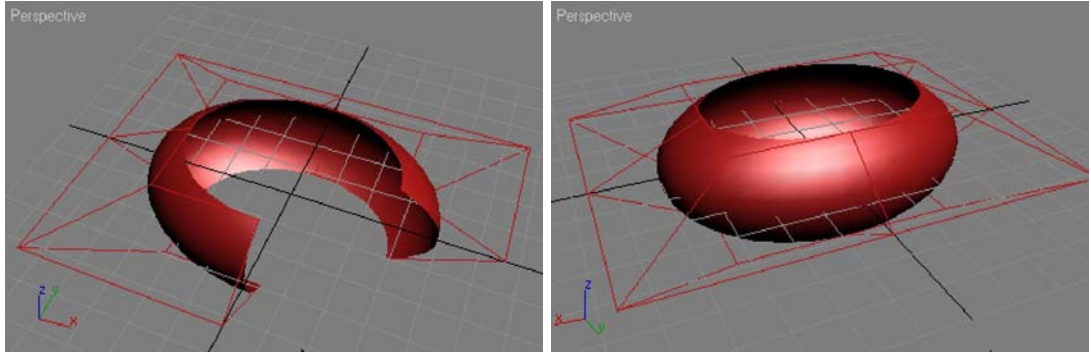
$$Q_{i,j} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} P_{i,j}, \text{ за } j = 0, \dots, m.$$

Това можем да направим поради геометричното свойство на NURBS кривите за афинна инвариантност на характеристичните точки и точките от кривата посочено в подраздел 1.2.4.

Получаваме биквадратната NURBS повърхнина $E_Q: \vec{r} = E_Q(u, v)$, чрез която точно представяме E , зададена с параметрите a, b, c , $u \in [u_s, u_e]$, $u_s, u_e \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ и $v \in [v_s, v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$, където

$$E_Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) Q_{i,j}. \quad (3.27)$$

На фигура 3.6 могат се видят две NURBS елипсоид-повърхнини (NE-повърхнини) получени за определени стойности на параметрите a, b, c, u_s, u_e, v_s и v_e , и изобразени с характеристичните си мрежи, състоящи се от контролните върхове $Q_{i,j}$.



а)

б)

Фиг. 3.6 а) NE-повърхнината за $a = 40$; $b = 30$; $c = 20$; $u_s = -45^\circ$; $u_e = 45^\circ$; $v_s = 0^\circ$; $v_e = 270^\circ$. б) NE-повърхнината за $a = 40$; $b = 30$; $c = 20$; $u_s = -45^\circ$; $u_e = 45^\circ$; $v_s = 0^\circ$; $v_e = 360^\circ$.

3.2.1.3. Построяване на NURBS параболоид-повърхнината (NP-повърхнината)

Задачата за построяване на повърхнината на параболоид можем да формулираме така:

Задача 3.2 Да се построи повърхнината на параболоид по зададени стойности на параметрите $a, b > 0$ и зададени интервали за u и v като $u \in [u_s; u_e]$, $u_s, u_e \in J \subseteq \mathbb{R}^+$ и $v \in [v_s; v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$ (от уравнение (3.17)), представена чрез биквадратна NURBS повърхнината.

Генераторната крива на тази повърхнината аналитично е дадена с уравнение (3.21). Намира се в равнината Oxz и представлява дъга от парабола. Тази крива може да бъде представена чрез NURBS крива от втора степен от вида даден с уравнение (2.19) и получена чрез метода, даден в раздел 2.4.

При ротационно построяване на биквадратна NURBS повърхнината по метода даден в точка 3.2.2.1. получаваме част от повърхнината на параболоид в частния случай когато $a = b$ и определена от параметрите u_s, u_e, v_s, v_e . Нека параболоида при $b = a$ от уравнение (3.17) да е $P_{b=a}$.

Биквадратната NURBS повърхнината, представляваща $P_{b=a}$ ще бъде от вида даден в точка 3.2.1.1. (реализиран с Алгоритъм 3.1) изразена с уравнение (3.26). Можем да я запишем така:

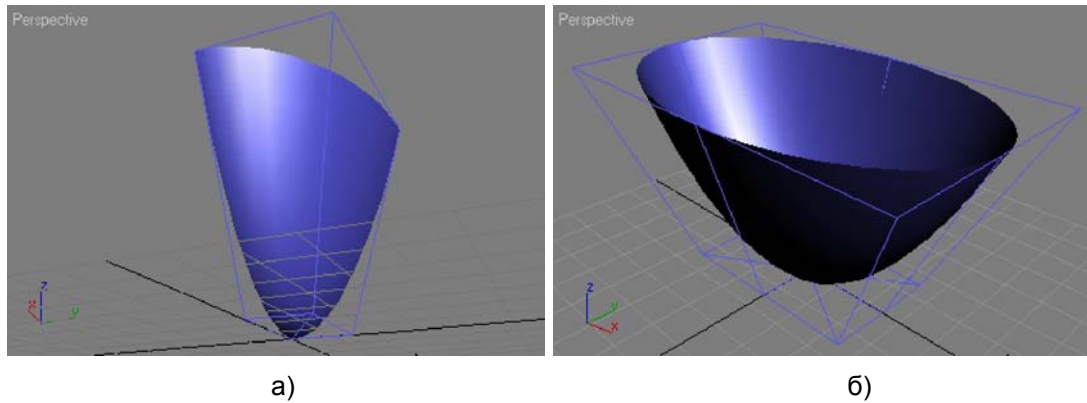
$$P_p(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}.$$

Параболоида P можем да получим от $P_{b=a}$ чрез мащабиране с матрица на трансформацията $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & b/a & 0 \\ 0 & 0 & 1 \end{pmatrix}$: $P = T.P_{b=a}$ (аналогично на Лема 3.1).

Следователно, ако приложим мащабиране на P_p с матрица на трансформация T , ще получим NURBS повърхнина, представяща P . Това извършваме като прилагаме тази афинна трансформация върху контролните върхове $P_{i,j}$ на повърхнината P_p . Получаваме новите контролни върхове $Q_{i,j} = T.P_{i,j}$, определящи новата NURBS повърхнина $P_Q: \vec{r} = P_Q(u, v)$. А уравнението ѝ ще се задава чрез:

$$P_Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) Q_{i,j}. \quad (3.28)$$

На фигура 3.7 могат се видят две NURBS параболоид-повърхнини (NP-повърхнини) получени за определени стойности на параметрите a, b, u_s, u_e, v_s и v_e , и изобразени с характеристичните си мрежи (състоящи се от контролните върхове $Q_{i,j}$).



Фиг. 3.7 а) NP-повърхнина за $a = 5$; $b = 3$; $u_s = 0$; $u_e = 10$; $v_s = 0^\circ$; $v_e = 90^\circ$;
б) NP-повърхнина за $a = 5$; $b = 3$; $u_s = 0$; $u_e = 10$; $v_s = 0^\circ$; $v_e = 360^\circ$.

3.2.1.4. Построяване на NURBS хиперболоид-повърхнина (NH-повърхнина)

Задачата за построяване на повърхнина на хиперболоид можем да формулираме така:

Задача 3.3 Да се построи повърхнина на хиперболоид по зададени стойности на параметрите $a, b, c > 0$ и зададени интервали за u и v като

$$u \in [u_s; u_e], \quad u_s, u_e \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right) \quad \text{и} \quad v \in [v_s; v_e], \quad v_s, v_e \in J \subseteq \mathbb{R} \quad (\text{от уравнение (3.8)}),$$

представена чрез биквадратна NURBS повърхнина.

Генераторната крива на тази повърхнина аналитично е дадена с уравнение (3.22). Намира се в равнината Oxz и представлява дъга от хипербола. Тази крива може да бъде представена чрез NURBS крива от втора степен от вида даден с уравнение (2.19) и получена чрез метода, даден в раздел 2.4.

При ротационно построяване на биквадратна NURBS повърхнина по подхода даден в точка 3.2.1.1. (реализиран с Алгоритъм 3.1) получаваме повърхнина от хиперболоид в частния случай когато $a = b$ (т.е. ротационен) и определена от параметрите u_s, u_e, v_s, v_e . Нека хиперболоидът от уравнение (3.8) при $b = a$ да е $H_{b=a}$.

Биквадратната NURBS повърхнина, представляваща $H_{b=a}$ ще бъде от вида, даден в точка 3.2.1.1. изразена с уравнение (3.26). Можем да я запишем чрез векторната функция:

$$H_P(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}.$$

Хиперболоида H можем да получим от $H_{b=a}$ чрез мащабиране с матрица на

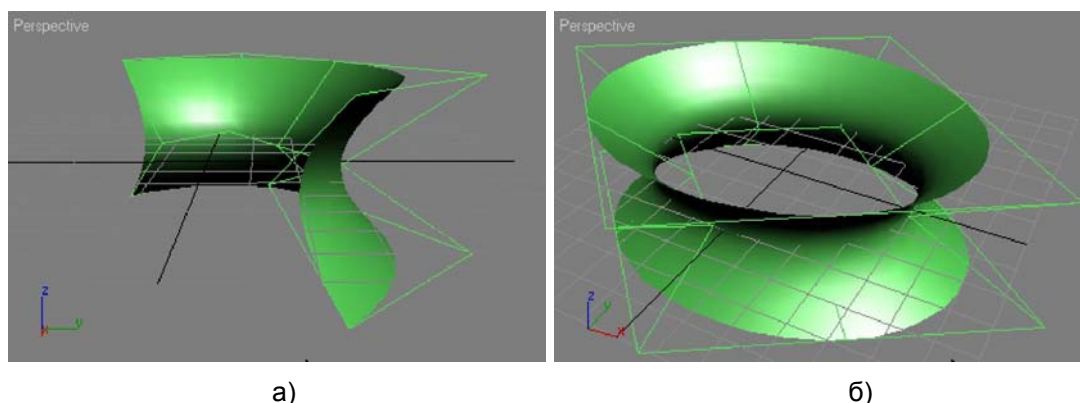
$$\text{трансформацията} \quad T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & b/a & 0 \\ 0 & 0 & 1 \end{pmatrix}: \quad H = T.H_{b=a} \quad (\text{аналогично на Лема 3.1}).$$

Следователно, ако приложим мащабиране на H_P с матрица на трансформация T , ще получим NURBS повърхнина, представяща H . Това извършваме като прилагаме тази афинна трансформация върху контролните върхове $P_{i,j}$ на повърхнината H_P . Получаваме новите контролни върхове $Q_{i,j} = T.P_{i,j}$, определящи новата NURBS повърхнина $H_Q: \vec{r} = H_Q(u, v)$. А уравнението ѝ ще се управлява от векторната функция:

$$H_Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) Q_{i,j}. \quad (3.29)$$

На фигура 3.8 могат се видят две NURBS хиперболоид-повърхнини (NH-повърхнини) получени за определени стойности на параметрите a, b, c, u_s, u_e, v_s и v_e , и изобразени с характеристичните си мрежи, състоящи се от контролните върхове

$Q_{i,j}$.



Фиг. 3.8 а) NH-повърхнина за $a = 40$; $b = 30$; $c = 20$; $u_s = -45^\circ$; $u_e = 45^\circ$; $v_s = 40^\circ$; $v_e = 220^\circ$; б) NH-повърхнина за $a = 40$; $b = 30$; $c = 20$; $u_s = -45^\circ$; $u_e = 45^\circ$; $v_s = 40^\circ$; $v_e = 400^\circ$.

3.2.1.5. Построяване на NURBS двоен-хиперболоид-повърхнина (NDH-повърхнина)

Задачата за построяване на повърхнина на двоен хиперболоид можем да формулираме така:

Задача 3.4 Да се построи повърхнина на двоен хиперболоид по зададени стойности на параметрите $a, b, c > 0$ и зададени интервали за u и v като $u \in [u_s; u_e]$, $u_s, u_e \in \left[0; \frac{\pi}{2}\right)$ и $v \in [v_s; v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$ (от уравнение (3.10)), представена чрез биквадратна NURBS повърхнина.

Генераторната крива на тази повърхнина аналитично е дадена с уравнение (3.23). Лежи в равнината Oxz и представлява дъга от хипербола. Тази крива може да бъде представена чрез NURBS крива от втора степен от вида, даден с уравнение (2.19) и получена чрез метода даден в раздел 2.4.

При ротационно построяване на биквадратна NURBS повърхнина по метода даден в точка 3.2.2.1. (реализиран с Алгоритъм 3.1) получаваме повърхнината на двоен хиперболоид в частния случай когато $a = b$ и определена от параметрите u_s, u_e, v_s, v_e . Нека двойния хиперболоид при $b = a$ от уравнение (3.10) да е $DH_{b=a}$.

Биквадратната NURBS повърхнина, представляваща $DH_{b=a}$ ще бъде от вида, даден в точка 3.2.1.1. изразена с уравнение (3.26). Можем да я запишем чрез векторната функция:

$$DH_P(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}.$$

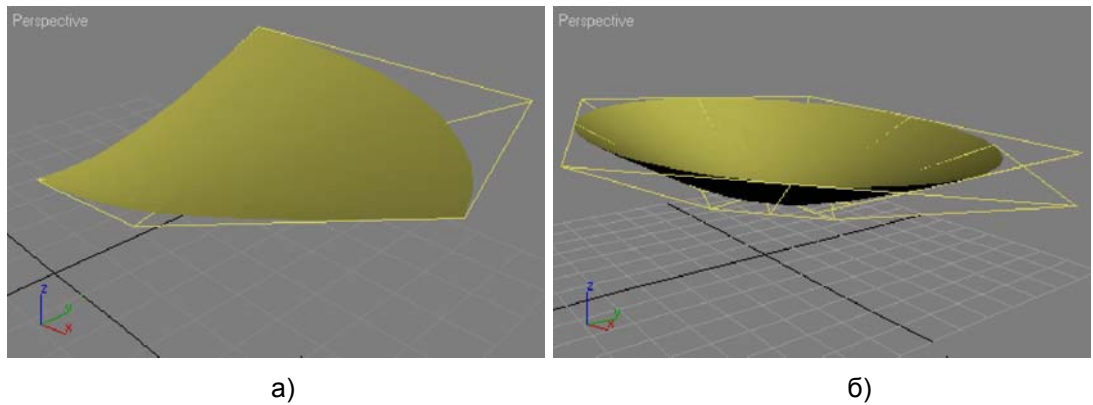
Двойния хиперболоид DH можем да получим от $DH_{b=a}$ чрез мащабиране с

матрица на трансформацията $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & b/a & 0 \\ 0 & 0 & 1 \end{pmatrix}$: $DH = T.DH_{b=a}$ (аналогично на Лема

3.1). Следователно, ако приложим мащабиране на DH_P с матрица на трансформация T , ще получим NURBS повърхнина, представяща DH . Това извършваме като прилагаме тази афинна трансформация върху контролните върхове $P_{i,j}$ на повърхнината DH_P . Получаваме новите контролни върхове $Q_{i,j} = T.P_{i,j}$, определящи новата NURBS повърхнина $DH_Q: \vec{r} = DH_Q(u, v)$. А уравнението ѝ ще се задава чрез:

$$DH_Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) Q_{i,j}. \quad (3.30)$$

На фигура 3.9 са показани две NURBS двоен-хиперболоид-повърхнини (NDH-повърхнини) получени за определени стойности на параметрите a, b, c, u_s, u_e, v_s и v_e , и изобразени с характеристикните си мрежи, състоящи се от контролните върхове $Q_{i,j}$.



Фиг. 3.9 а) NDH-повърхнина за $a = 45$; $b = 28$; $c = 20$; $u_s = 0^\circ$; $u_e = 60^\circ$; $v_s = 10^\circ$; $v_e = 85^\circ$; б) NDH-повърхнина за $a = 45$; $b = 28$; $c = 20$; $u_s = 0^\circ$; $u_e = 60^\circ$; $v_s = 0^\circ$; $v_e = 400^\circ$.

3.2.1.6. Построяване на NURBS конус-повърхнина (NC-повърхнина)

Задачата за построяване на повърхнина на конус можем да формулираме така:

Задача 3.5 Да се построи повърхнина на конус по зададени стойности на параметрите $a, b, c > 0$ и зададени интервали за u и v като $u \in [u_s; u_e]$, $u_s, u_e \in J \subseteq \mathbb{R}$ и $v \in [v_s; v_e]$, $v_s, v_e \in J \subseteq \mathbb{R}$ (от уравнение (3.12)), представена чрез биквадратна NURBS повърхнина.

Генераторната крива на тази повърхнина аналитично е дадена с уравнение (3.24), намира се в равнината Oxz и представлява отсечка. Тя може да бъде представена чрез NURBS крива от втора степен от вида даден с уравнение (2.19), с $n+1$ контролни върхове P_i с тегла w_i и възлов вектор $T = \{t_0 = 0, 0, 0, t_1, t_1, t_2, t_2, \dots, t_m, t_m, t_{m+1} = 1, 1, 1\}$ по следния начин: Броят на сегментите е $m+1$. Възлите t_l избираме така: $t_0 = 0$ и

$$t_{l+1} = t_l + \frac{1}{m+1}, \quad \text{за } l = 0, \dots, m. \quad \text{Контролните върхове са } P_0 = (au_s, 0, cu_s),$$

$$P_n = (au_e, 0, cu_e) \text{ и } P_i = i \frac{1}{2(m+1)} (P_0 + P_n) \text{ за } i = 1, \dots, n-1 \text{ с тегла } w_i = 1 \text{ за } i = 0, \dots, n.$$

При ротационно построяване на биквадратна NURBS повърхнина по метода, даден в точка 3.2.1.1. (реализиран с Алгоритъм 3.1), получаваме повърхнина на конус в частния случай когато $a = b$ и определена от параметрите u_s, u_e, v_s, v_e . Нека конуса при $b = a$ от уравнение (3.12) да е $C_{b=a}$.

Биквадратната NURBS повърхнина, представляваща $C_{b=a}$ ще бъде от вида даден в точка 3.2.1.1. изразена с уравнение (3.26). Можем да я запишем така:

$$C_P(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j}.$$

Можем да получим конуса C от $C_{b=a}$ чрез мащабиране с матрица на

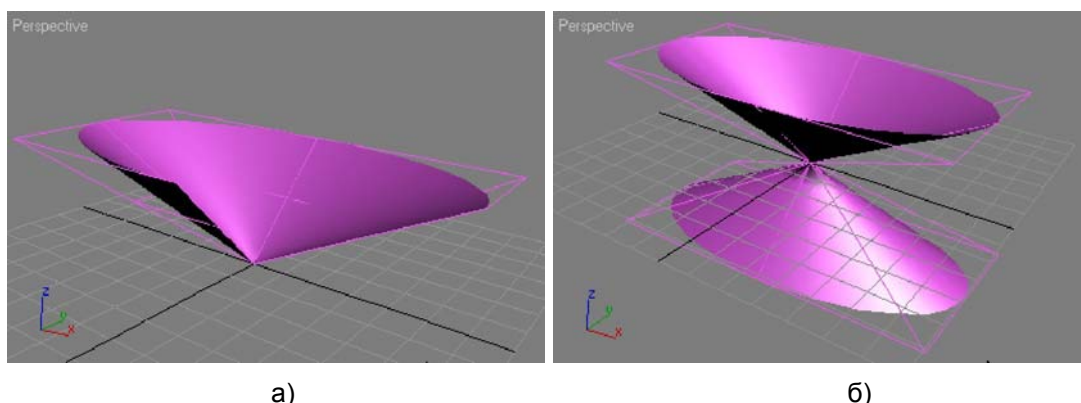
трансформация $T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & b/a & 0 \\ 0 & 0 & 1 \end{pmatrix}$: $C = T.C_{b=a}$ (аналогично на Лема 3.1).

Следователно, ако приложим мащабиране на C_P с матрица на трансформация T , ще получим NURBS повърхнина, представяща C . Това извършваме като прилагаме тази афинна трансформация върху контролните върхове $P_{i,j}$ на повърхнината C_P .

Получаваме новите контролни върхове $Q_{i,j} = T.P_{i,j}$, определящи новата NURBS повърхнина $C_Q : \vec{r} = C_Q(u, v)$. А уравнението ѝ ще бъде:

$$C_Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) Q_{i,j}. \quad (3.31)$$

На фигура 3.10 могат се видят две NURBS конус-повърхнини (NC-повърхнини) получени за определени стойности на параметрите a, b, c, u_s, u_e, v_s и v_e , и изобразени с характеристикните си мрежи, състоящи се от контролните върхове $Q_{i,j}$.



Фиг. 3.10 а) NC-повърхнина за $a = 2; b = 1; c = 1; u_s = 0; u_e = 30; v_s = 0^\circ; v_e = 270^\circ$;
б) NC-повърхнина за $a = 2; b = 1; c = 1; u_s = -30; u_e = 30; v_s = 0^\circ; v_e = 360^\circ$.

3.2.2. Представяне на цилиндрите чрез биквадратни NURBS повърхнини

Повърхнината на който да е от цилиндрите може да бъде определена за построяване чрез параметрите от СПУ за конкретната повърхнина. И трите цилиндъра имат генератори, които могат да бъдат представени чрез NURBS криви. Тези генератори откриваме като намерим сечението на всеки от цилиндрите с равнината $\alpha_t : z = t$ ($\parallel Oxy$). При $t = 0$ за елиптичния цилиндър се получава елипсата:

$$\varepsilon^{EC} : \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1; \quad \varepsilon^{EC} : \begin{cases} x = a \cdot \cos \alpha \\ y = b \cdot \sin \alpha \\ z = 0 \end{cases}, \quad \alpha \in [0; 2\pi], \quad (3.32)$$

за хиперболичния цилиндър се получава хиперболата:

$$\chi^{HC} : \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \text{ и } \chi^{HC} : \begin{cases} x = \frac{a}{\cos \alpha} \\ y = b \cdot \operatorname{tg} \alpha, \alpha \in [0; 2\pi], t \neq \frac{\pi}{2}, \frac{3\pi}{2} \\ z = 0 \end{cases} \quad (3.33)$$

и за параболичния цилиндър се получава параболата:

$$\pi^{PC} : y^2 = 2px; \pi^{PC} : \begin{cases} x = \frac{u^2}{2p} \\ y = u, \quad u \in (-\infty, +\infty). \\ z = 0 \end{cases} \quad (3.34)$$

Тези криви можем да представим чрез NURBS криви по подходите, дадени в Глава II. За всяка крива имаме по два допълнителни параметъра, които задават начало и край на дъгата. Следователно, тези параметри трябва да участват в дефинирането на частите от цилиндрите, които ще се построяват. Тези повърхнини са неограничени и по направление на техните управителни. Следователно е необходимо и въвеждането на още два параметъра, които да задават начало и край на построяване на повърхнината по това направление. Това направление в канонично представяне на цилиндрите е оста Oz . Нека тези два допълнителни параметъра да бъдат t_s и t_e ($t_s, t_e \in \mathbb{R}$).

Задачите за построяване на части от цилиндричните повърхнини можем да формулираме така:

Задача 3.6 Да се построи повърхнина на елиптичен цилиндър (уравнение 3.13) по зададени стойности на параметрите $a, b > 0$ и зададени интервали за α (от уравнение (3.32)) и t като $\alpha \in [\alpha_s; \alpha_e]$, $\alpha_s, \alpha_e \in J \subseteq \mathbb{R}$ и $t \in [t_s; t_e]$, $t_s, t_e \in J \subseteq \mathbb{R}$, представена чрез биквадратна NURBS повърхнина (NEC- повърхнина).

Задача 3.7 Да се построи повърхнина на хиперболичен цилиндър (уравнение 3.14) по зададени стойности на параметрите $a, b > 0$ и зададени интервали за α (от уравнение (3.33)) и t като $\alpha \in [\alpha_s; \alpha_e]$, $\alpha_s, \alpha_e \in J \subseteq \mathbb{R}$ и $t \in [t_s; t_e]$, $t_s, t_e \in J \subseteq \mathbb{R}$, представена чрез биквадратна NURBS повърхнина (NHC- повърхнина).

Задача 3.8 Да се построи повърхнина на параболичен цилиндър (уравнение 3.15) по зададени стойности на параметрите $p > 0$, $u \in [u_s; u_e]$, $u_s, u_e \in J \subseteq \mathbb{R}$ (от уравнение (3.34)) и $t \in [t_s; t_e]$, $t_s, t_e \in J \subseteq \mathbb{R}$, представена чрез биквадратна NURBS повърхнина

(NPC- повърхнина).

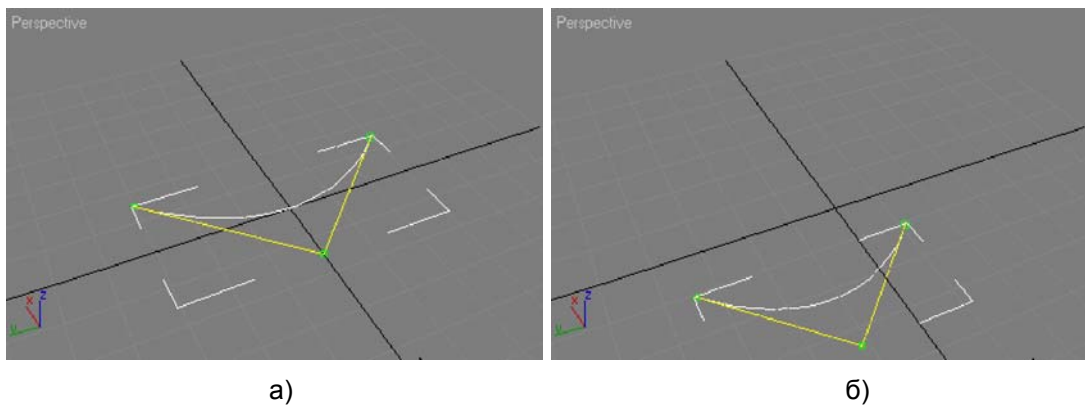
Всеки един от генераторите нека да бъде представен като NURBS крива (уравнение 1.17) лежаща в равнината Oxy :

$$\gamma(u) : \begin{cases} x(u) = \sum_{i=0}^n R_{i,2}(u) x_i \\ y(u) = \sum_{i=0}^n R_{i,2}(u) y_i, \quad u \in [0;1] \\ z(u) = 0 \end{cases} \quad (3.35)$$

с $n+1$ контролни върхове $P_i(x_i, y_i, z_i)$ с тегла w_i и възлов вектор $U = \{0, 0, 0, u_1, u_1, u_2, u_2, \dots, u_k, u_k, 1, 1, 1\}$ (фиг. 3.11а).

Избираме единичен вектор $\vec{Q} = (0, 0, 1)$ ако $t_s < t_e$ и $\vec{Q} = (0, 0, -1)$ ако $t_s > t_e$, и $d = |t_s - t_e|$. Числото d можем да наричаме дистанция. Поставяме кривата γ в начална позиция като $z_i = t_s, i = 0, \dots, n$ (фиг. 3.11б). Нека $S(u, v)$ да бъде търсената NURBS повърхнина. Построяваме S чрез замитане на γ на дистанция d по направление на \vec{Q} . Обозначаваме параметъра за посоката на замитане с v , $0 \leq v \leq 1$ и тогава $S(u, v)$ трябва да удовлетворява двете условия:

- 1) за фиксирано \bar{u} , $S(\bar{u}, v)$ е права линия от $\gamma(\bar{u})$ до $\gamma(\bar{u}) + d\vec{Q}$;
- 2) за фиксирано \bar{v} , $S(u, \bar{v}) = \gamma(u) + \bar{v}d\vec{Q} = \sum_{i=0}^n R_{i,2}(u) (P_i + \bar{v}d\vec{Q})$.

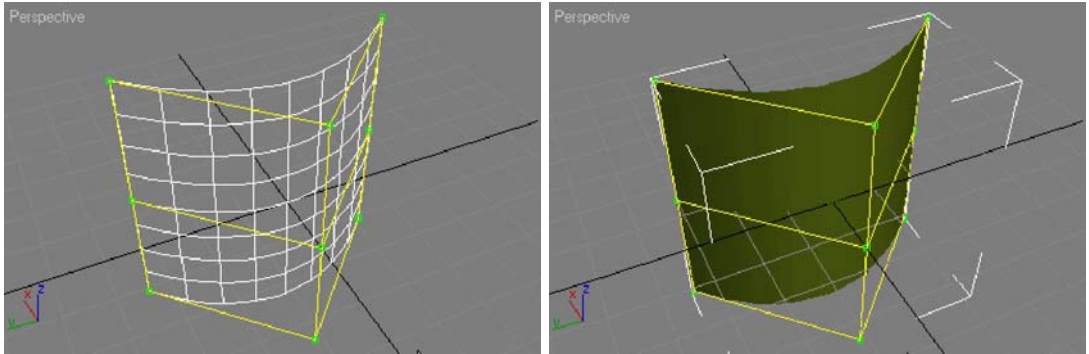


Фиг. 3.11 Генераторната NURBS крива.

Следователно, за търсената повърхнина получаваме (фиг. 3.12):

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^2 R_{i,2,j,2}(u, v) P_{i,j}. \quad (3.36)$$

Възловите вектори са U и V , където $V = \{0, 0, 0, 1, 1, 1\}$, а U е възловия вектор на γ . Контролните върхове са $P_{i,0} = P_i$, $P_{i,2} = P_i + d\vec{Q}$ и $P_{i,1}$ е $P_{i,1} = (P_{i,0} + P_{i,2})/2$. Теглата са $w_{i,0} = w_{i,1} = w_{i,2} = w_i$.



Фиг. 3.12 Построяване на NURBS повърхнината на цилиндър.

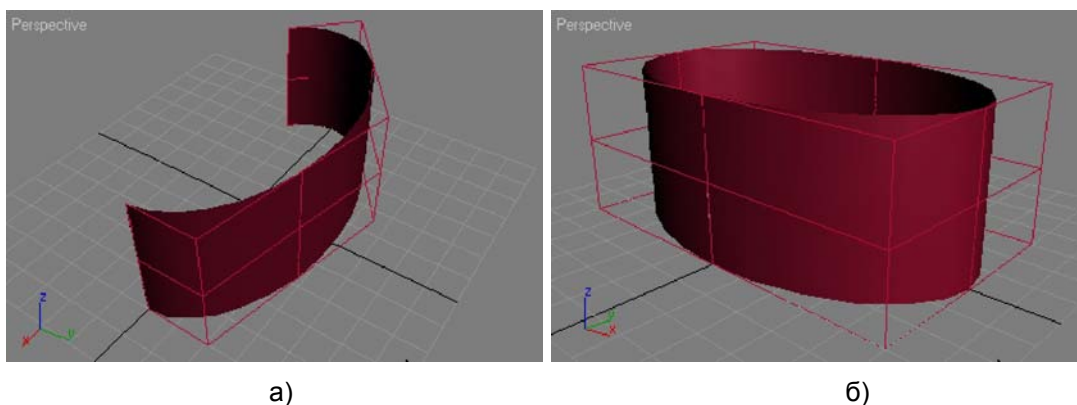
За да осигурим възможност за локална модификация на получената повърхнина е нужно да направим подразделяне на $S(u, v)$ по направлението v . Тъй като по това направление повърхнината е праволинейна, следователно $S(u, v)$ можем да представим така:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2,j,2}(u, v) P_{i,j}, \quad (3.37)$$

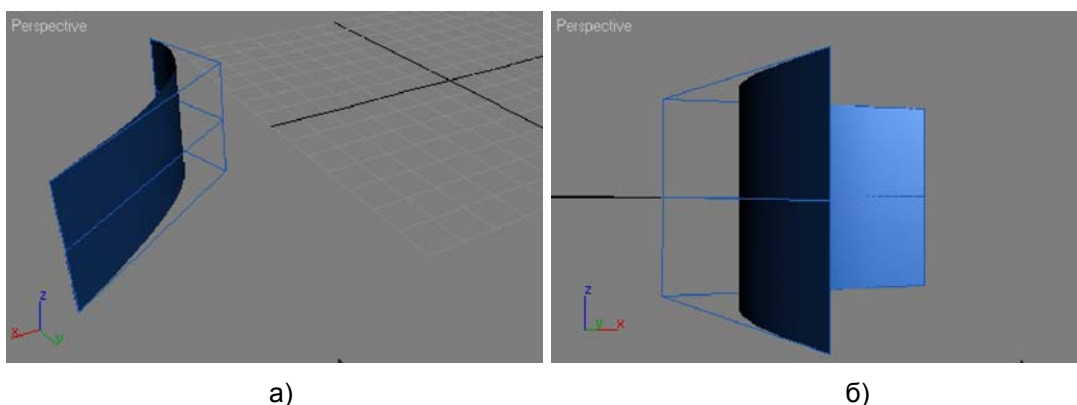
където $V = \{v_0 = 0, 0, 0, v_1, v_1, \dots, v_{h-1}, v_{h-1}, v_h = 1, 1, 1\}$ като h е броят на сегментите, а v_l избираме така: $f = \frac{1}{h}$, $v_0 = 0$ и $v_{l+1} = v_l + f$, за $l = 0, \dots, h-1$. Контролните върхове са $P_{i,0} = P_i$ и $P_{i,j}$, $j = 1, \dots, m$ определяме така $x_{i,j} = x_{i,0}$, $y_{i,j} = y_{i,0}$, $z_{i,j} = z_{i,0} + j \cdot q$, където $q = \frac{d}{2h}$. Теглата са $w_{i,j} = w_i$.

Локална модификация: При промяна на позицията на контролен връх $P_{i,j}$ или стойността на неговото тегло $w_{i,j}$, формата на повърхнината се променя само в правоъгълника $[u_i, u_{i+3}) \times [v_j, v_{j+3})$.

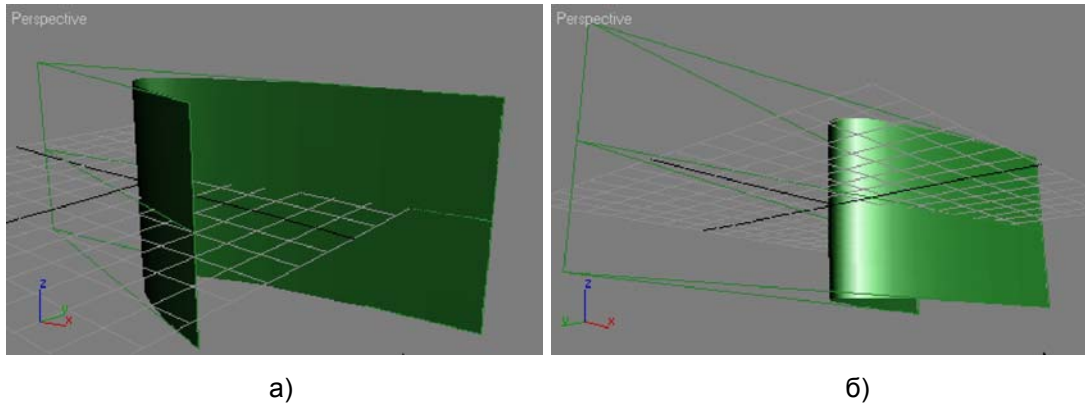
На фигури от 3.13 до 3.15 са показани повърхнините на EC , HC и PC представени чрез биквадратни NURBS повърхнини съответно NEC-повърхнини, NHC-повърхнини и NPC-повърхнини. Те са получени за определени стойности на параметрите a , b , α_s , α_e , t_s и t_e .



Фиг. 3.13 а) NEC-повърхнина за $a = 50$; $b = 30$; $\alpha_s = 0^\circ$; $\alpha_e = 210^\circ$; $t_s = 0$; $t_e = 30$;
 б) NEC-повърхнина за $a = 50$; $b = 30$; $\alpha_s = 0^\circ$; $\alpha_e = 360^\circ$; $t_s = 0$; $t_e = 30$.



Фиг. 3.14 а) NHC-повърхнина за $a = 100$; $b = 50$; $\alpha_s = -10^\circ$; $\alpha_e = 45^\circ$; $t_s = 0$; $t_e = 30$;
 б) NHC-повърхнина за $a = 100$; $b = 50$; $\alpha_s = -10^\circ$; $\alpha_e = 45^\circ$; $t_s = -30$; $t_e = 30$.



Фиг. 3.15 а) NPC-повърхнина за $p = 5; u_s = -10; u_e = 30; t_s = -30; t_e = 30;$
 б) NPC-повърхнина за $p = 5; u_s = -30; u_e = 30; t_s = -30; t_e = 30.$

3.2.3. Представяне на хиперболичен параболоид чрез биквадратна NURBS повърхнина

Задачата за построяване на повърхнина на хиперболичен параболоид можем да формулираме така:

Задача 3.9 Да се построи повърхнина на хиперболичен параболоид по зададени стойности на параметрите $a, b > 0$ (от уравнение (3.19)), представена чрез биквадратна NURBS повърхнина (NHP-повърхнина).

За HP е важно да се види как равнините β_p и γ_q го разсложават. Сеченията $HP \cap \beta_p : y = p$ са параболите:

$$\pi_p : x^2 = 2a^2 \left(z + \frac{p^2}{2b^2} \right) \quad (3.38)$$

разположени в β_p по направление $\parallel Oz^+$ и върхове $F_p \left(0, -\frac{p^2}{2b^2} \right)$. Сеченията

$HP \cap \gamma_q : x = q$ са параболите:

$$\pi_q : y^2 = -2b^2 \left(z - \frac{q^2}{2a^2} \right) \quad (3.39)$$

разположени в γ_q по направление $\parallel Oz^-$ и върхове $G_q \left(0, \frac{q^2}{2a^2} \right)$.

Хиперболичния параболоид е симетрична повърхнина спрямо равнините Oxz и Oyz . С уравнение (3.38) показахме, че ако фиксираме $y = p$ ($HP \cap \beta_p : y = p$), то

всички точки от HP с $y = p$ образуват параболите π_p . Следователно, ако поставим x да се изменя в диапазона $x \in [x_s, x_e]$ можем да намерим съответните z и точките с тези координати ще изграждат π_p . Аналогично с уравнение (3.39) показахме, че ако фиксираме $x = q$ ($HP \cap \gamma_q : x = q$), то всички точки от HP с $x = q$ образуват параболите π_q . Следователно, ако поставим y да се изменя в диапазона $y \in [y_s, y_e]$ можем да намерим съответните z и точките с тези координати ще изграждат π_q .

Следователно, за да начертаем хиперболичен параболоид, ние можем да го ограничим с равнините $HP \cap \gamma_{x_s} : x = x_s$, $HP \cap \gamma_{x_e} : x = x_e$, $HP \cap \beta_{y_s} : y = y_s$, $HP \cap \beta_{y_e} : y = y_e$. За да е напълно определена повърхнината на HP , която построяваме и симетрична спрямо Oxz и Oyz , поставяме условието $x_s = -x_e$ и $y_s = -y_e$, и построяваме за $x \in [x_s, x_e]$ и $y \in [y_s, y_e]$. Така повърхнината HP_d от HP (фиг. 3.16), която получаваме е изградена от параболични криви в направление по x и по направление на y . От това следва, че HP_d можем да построим като биквадратна NURBS повърхнина от вида:

$$HP_d(u, v) = \sum_{i=0}^2 \sum_{j=0}^2 R_{i,2,j,2}(u, v) P_{i,j} \quad (3.40)$$

с възлови вектори без вътрешни възли $U = \{0, 0, 0, 1, 1, 1\}$ и $V = \{0, 0, 0, 1, 1, 1\}$, замествайки направлението x с направлението u , и направлението y с направлението v в NURBS базиса.

Контролните върхове $P_{0,0}, P_{0,2}, P_{2,0}, P_{2,2}$ получаваме от поставените ограничителни стойности x_s, x_e, y_s, y_e и z координати изчислени от уравнение (3.18). Контролните върхове $P_{0,1}, P_{1,0}, P_{1,2}, P_{2,1}$ намираме като пресечни точки на контурните линии на повърхнината, които са параболите $\pi_{p=y_s}, \pi_{p=y_e}, \pi_{q=x_s}, \pi_{q=x_e}$. Повърхнината на HP преминава през т. $O(0, 0, 0)$ и поради симетрията на повърхнината HP_d спрямо т. O следва, че HP_d ще преминава през т. O за $u = v = \frac{1}{2}$. Тогава контролния връх

$P_{1,1}(x_{1,1}, y_{1,1}, z_{1,1})$ може да се определи от системата $S\left(\frac{1}{2}, \frac{1}{2}\right) = (0, 0, 0)$ с единствени неизвестни $x_{1,1}, y_{1,1}, z_{1,1}$.

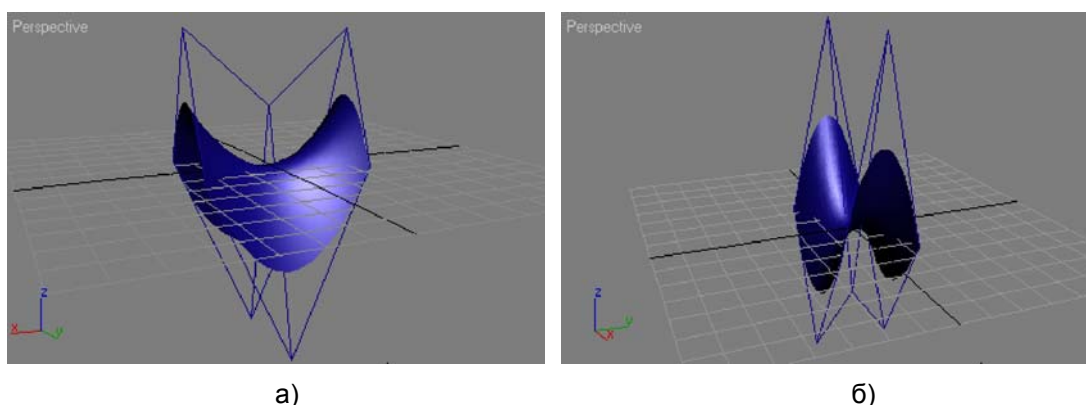
Резултатите, които получаваме за контролните върхове $P_{ij}(x_{ij}, y_{ij}, z_{ij})$, $i, j = 0, 1, 2$, които определят повърхнината, са както следва:

$$\begin{aligned} P_{00}(x_s, y_s, dx - dy), \quad P_{01}(x_s, 0, dx + dy), \quad P_{02}(x_s, y_e, dx - dy) \\ P_{10}(0, y_s, -dx - dy), \quad P_{11}\left(0, 0, -\Sigma / \left(N_{1,2}\left(\frac{1}{2}\right)\right)^2\right), \quad P_{12}(0, y_e, -dx - dy), \quad (3.41) \\ P_{20}(x_e, y_s, dx - dy), \quad P_{21}(x_e, 0, dx + dy), \quad P_{22}(x_e, y_e, dx - dy) \end{aligned}$$

където $dx = \frac{x_s^2}{2a^2}$, $dy = \frac{y_s^2}{2b^2}$ и

$$\begin{aligned} \Sigma = & N_{0,2} \cdot (N_{0,2} \cdot z_{00} + N_{1,2} \cdot z_{01} + N_{2,2} \cdot z_{02}) + \\ & N_{1,2} \cdot (N_{0,2} \cdot z_{10} + N_{2,2} \cdot z_{12}) + \\ & N_{2,2} \cdot (N_{0,2} \cdot z_{20} + N_{1,2} \cdot z_{21} + N_{2,2} \cdot z_{22}) \end{aligned}$$

Формата на хиперболичния параболоид се определя от коефициентите a и b . Следователно той може да бъде построен чрез задаване на размери в широчина (W) по направление на Ox и дължина (L) по направление на Oy . Тогава $x_s = -\frac{W}{2}$, $x_e = \frac{W}{2}$, $y_s = \frac{L}{2}$ и $y_e = -\frac{L}{2}$. Така построяваме NURBS хиперболичен параболоид повърхнина (NHP-повърхнина) чрез задаване на параметрите a , b , W и L . Примери на NHP-повърхнини получени за определени стойности на тези параметри могат да се видят на фигура 3.16.



Фиг. 3.16 а) NHP-повърхнина за $a = 2$; $b = 2$; $W = 50$; $L = 30$;
б) NHP-повърхнина за $a = 2.8$; $b = 2.1$; $W = 47$; $L = 28$.

За да се осигури възможност за локална модификация на NHP е нужно да се направи подразделяне на повърхнината. За тази цел ще дадем един подход за подразделяне на биквадратна NURBS повърхнина (реализиран в Алгоритъм 3.2).

3.2.4. Подход за подразделяне на биквадратна NURBS повърхнина без вътрешни възли в U и V

NURBS повърхнина може да се подраздели по едно направление (u или v) чрез добавяне на междинен възел във възловия вектор, определящ това направление (U или V). Появата на допълнителен възел изисква добавянето и на нови контролни върхове. Повърхнината не се променя, но броят на контролните върхове се увеличава. В [91, 141-162 с.] е представен алгоритъм за вмъкване на възел, а в [97, 351-356 с. и 458-461 с.] са дадени методи за подразделяне на B-сплайн крива и повърхнина.

Тук, ще предложим друг подход за подразделяне на NURBS повърхнини. Той ще бъде реализиран специално за биквадратни такива без вътрешни възли в U и V . В [17, 282-284 с.] е даден методът на Де Кастельо за подразделяне на криви на Безие, който може да се приложи и за повърхнини на Безие. И тъй като NHP (а и другите разглеждани повърхнини) е биквадратна повърхнина от вида даден с уравнение (3.40) (без вътрешни възли в U и V), то тя може да бъде представена и във форматана Безие. Следователно, може да се ползва методът на Кастельо за да се подраздели NURBS повърхнината. Подходът включва две основни стъпки:

- подразделяне на контролната мрежа в избраното направление на повърхнината (u или v) по метода на Де Кастельо, за да се намерят новите контролни върхове;
- добавяне на възел във възловия вектор определящ избраното направление с кратност равна на степента на базовата функция в това направление и стойност за която се получават добавените контролни върхове.

От това, че разглежданите тук повърхнини са биквадратни следва, че добавянето на възел ще е с кратност 2. Стойността на възела може да се избере така, че да се явява средна стойност за възловия вектор. Тогава новите контролни върхове се намират по метода на Кастильо за стойност на параметъра равна на стойността на кратния възел.

Този подход за подразделяне на биквадратна NURBS повърхнина в едно направление (U) ще реализираме чрез Алгоритъм 3.2. Той подразделя повърхнина на seg на брой подразделяния. Контролните върхове на повърхнината се подават с двумерния масив s_aM . Броят на възлите в направлението се подава с $numKnots_u$.

Получените контролни върхове на повърхнината се записват в двумерния масив s_{aP} , а възловият вектор в едномерния масив s_{aU} .

Алгоритъм 3.2 SubdivideTheSurfaceOnU

Вход: s_{aM} , numKnots_u, seg;

Изход: s_{aP} , s_{aU} ;

Процедура:

- 1.) Определя се възловия вектор чрез добавяне на кратни възли.
- 2.) Определя се новият брой на контролните върхове в направлението на подразделяне.
- 3.) Осъществява се подразделянето със съответната стойност на кратен възел.
- 4.) Край.

Псевдопрограмният код изглежда така:

```

1) SubdivideTheSurfaceOnU( $s_{aM}$ , numKnots_u, seg,  $s_{aP}$ ,  $s_{aU}$ ) {
2)  $t = 1.0/seg$ ;  $g = 0.0$ ;  $q = 1$ ;  $i = 3$ ;  $k = seg$ ;
3) while(  $q < seg$  ) do {
    a.  $g = g+t$ ;  $s_{aU}[i+1] = g$ ;  $s_{aU}[i+2] = g$ ;  $i = i+2$ ;  $q = q+1$ ; }
4) for i in numKnots_u- $s_{order}+1$  to numKnots_u do  $s_{aU}[i] = 1.0$ ;
5) numCVs_u =  $2*seg+1$ ;
6) for j in 1 to 3 do {
    a.  $e = 1$ ;  $s = 0$ ;
    b. while(  $k > 1$  ) do {
        i.  $t = 1.0/k$ ;  $e = 3$ ;
        ii. for i in 1 to 3 do {
            1.  $s_{aP}[2*s+i][j] = s_{aM}[1][j]$ ;
            2.  $s_{aL}[e][j] = s_{aM}[e][j]$ ;
            3. for h in 1 to  $e-1$  do {
                4.  $s_{aM}[h][j] = (1.0-t)*s_{aM}[h][j] + t*s_{aM}[h+1][j]$ ; }
                5.  $e = e-1$ ; }
            iii. for i in 1 to 3 do {  $s_{aM}[i][j] = s_{aL}[i][j]$ ; }
            iv.  $s = s+1$ ;  $k = k-1$ ; } }
7) for j in 1 to 3 do
    a. for i in 1 to 3 do  $s_{aP}[numCVs_u-3+i][j] = s_{aM}[i][j]$ ;
8) }
```

Коментари по алгоритъма:

1. На редове от 2) до 4) се определя вектор чрез добавяне на кратни възли. Определя се стъпката t , с която се променят стойностите на възлите във възловия вектор като се започне от 0 и се достигне до 1.
2. На ред 5) в променливата numCVs_u се определя новия брой на контролните върхове в направлението на подразделяне.

3. В точка 6)-b-ii. се осъществява подразделянето на повърхнината за конкретна стойност на кратен възел и намирането на новите контролни върхове в s_{aP} .

NURBS повърхнината е от вида:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,2;j,2}(u, v) P_{i,j},$$

определена от: $(n+1).(m+1)$ контролни върхове $P_{i,j}$ с тегла $w_{i,j}$, възлови вектори от вида $U = \{0, 0, 0, u_1, u_1, u_2, u_2, \dots, u_h, u_h, 1, 1, 1\}$ и $V = \{0, 0, 0, v_1, v_1, v_2, v_2, \dots, v_k, v_k, 1, 1, 1\}$, подразделена на $h+1$ сегмента по направлението u и $k+1$ сегмента по направлението v .

Локална модификация: При промяна на позицията на контролен връх $P_{i,j}$ или стойността на неговото тегло $w_{i,j}$, формата на повърхнината се променя само в правоъгълника $[u_i, u_{i+3}) \times [v_j, v_{j+3})$.

3.3 Изводи

От направените теоретични разработки в Глава III могат да бъдат направени следните изводи:

1. Предвид приетия подход за представяне на повърхнините от втора степен чрез NURBS, те са разделени условно в три групи. За представянето на E, P, H, DH и C (I-ва група) чрез NURBS повърхнини се използват частните им случаи посочени в Теорема 3.1, които преставляват ротационни повърхнини. Първо те се представят в NURBS модела, а след това чрез геометрична трансформация на мащабиране се получават и NURBS моделите на E, P, H, DH и C . За определянето на повърхнините, които трябва да бъдат построени, се използват параметрите им от техните скаларно-параметрични уравнения. Подходите (в съчетание с тези за представяне на генераторите в NURBS модела) позволяват подразделяне на повърхнините на произволен брой сегменти и по двете направления u и v , което дава възможност за локална модификация върху отделни техни части.

2. За получаване на NURBS моделите на E, P, H, DH и C е разработен Алгоритъм 3.1 за построяване на NURBS повърхнина чрез ротиране на генераторна NURBS линия по зададени начален и краен ъгъл. Алгоритъмът е предложен във вариант за построяване на биквадратна повърхнина по дадена NURBS линия от втора степен намираща се в равнината Oxz , която се завърта около оста Oz . Той

позволява подразделяне на повърхнината на произволен брой сегменти по направлението на ротиране. Следователно, предоставена е възможност за локална модификация върху отделни нейни части. Това пък от своя страна, дава възможност една част от нея да бъде променяна, а останалата част да запазва оригиналната си форма, т.е. да запазва геометричните си свойства. Получените резултати са приети за публикувани в списание "Computer Engineering", София (2008 г.).

3. Даден е подход за представяне на цилиндричните повърхнини EC , HC и PC (II-ра група) чрез биквадратни NURBS повърхнини. Моделите на NURBS повърхнините се получават чрез замитане на генераторната NURBS крива от втора степен. Подходът позволява (в съчетание с подходите за представяне на генераторите в NURBS модела) подразделяне на повърхнините на произволен брой сегменти и по двете направления u и v . Следователно, предоставена е възможност за локална модификация върху отделни части от повърхнините. Също както при споменатите по-горе повърхнини, така и тук, това дава възможност да бъде променяна една част от дадената повърхнина, а останалата част да запазва оригиналната си форма. Резултатите са докладвани на международната конференция "Doctoral-students Conference", Гърция (2007 г.).

4. Предложен е подход за представяне на HP (III-та група) чрез биквадратна NURBS повърхнина. Друг подход за това не беше открит в литературните източници. Повърхнината, която се построява, се дефинира чрез параметрите a и b от скаларно-параметричните уравнения и още два допълнителни, които определят височина и дължина на повърхнината. След подразделяне (например, прилагайки Алгоритъм 3.2) на повърхнината по двете направления u и v , става възможна локалната модификация върху отделни нейни части. Също както при споменатите по-горе повърхнини, така и тук, това дава възможност една част да бъде променяна, а останалата - да запазва геометричните си свойства. Предложеният NURBS модел на HP е докладван на юбилейната конференция на Русенския университет "Ангел Кънчев" (2005 г.).

5. Предложен е алгоритъм (Алгоритъм 3.2), който реализира подход за подразделяне (на произволен брой сегменти) на биквадратна NURBS повърхнина дефинирана върху възлови вектори без вътрешни възли. Това дава възможност повърхнините, за които са предложени модели тук, да бъдат подразделяни на повече сегменти, ако е необходимо. Така дизайнерът, работещ с една от тези повърхнини, може свободно да определя нивото на гъвкавост в моделирането.

Глава четвърта
ПРИЛАГАНЕ НА NURBS МОДЕЛИТЕ
НА КРИВИТЕ И ПОВЪРХНИНТЕ ОТ ВТОРА СТЕПЕН
В 3D ГРАФИЧНАТА СИСТЕМА ЗА МОДЕЛИРАНЕ
AUTODESK 3DS MAX

3D ГСГМ непрекъснато се развиват с цел осигуряване на нови, по-ефективни възможности, които подобряват синтеза на изображения. Основна част от този процес са средствата за моделиране, които тези системи предоставят. Това включва: координатното пространство и изгледи от него, графичните примитиви и моделите, които стоят зад тях, графични инструменти, операции между графични обекти, модификатори и средства позволяващи интерактивно моделиране.

В Глава II и Глава III предложихме NURBS модели на криви и повърхнини за добре познати геометрични обекти. Целта е те да се включат като графични примитиви в 3D графичните системи за дизайн.

Ефективността, надеждността и функционалността на предложените в Глава II и Глава III модели и алгоритми могат да бъдат показани и оценени в практиката само ако те бъдат приложени в една 3D графична система за дизайн. Това обикновено става чрез разработването на специализирани софтуерни модули (CCM) за конкретната система. Така може да бъде направен и анализ на тяхната работа и на графичните примитиви, които те създават.

4.1. Възможности за функционално разширяване на 3D графичните системи за моделиране

Една 3D графична система може да бъде разработена така, че да позволява разширяване на функционалните си възможности, с което още повече да се повиши нейното ниво на полезност и конкретност [16, 29, 35, 38, 82, 84, 112]. Такива графични приложения притежават инструментариум за разработка на софтуер (ИРП) (от англ. Software Development Kit (SDK)) и (в повечето случаи) скриптов език (СЕ) (от англ. Script Language (SL)). Основната част от инструментариума се състои от приложен програмен интерфейс (ППИ) (ППИ, от англ. Application Programming Interface (API)), който обикновено е съвкупност от класове, написани на език от високо ниво. Основното предназначение на ИРП и СЕ е предоставянето на възможност за разработване на специализирани софтуерни модули за графичната система.

Приложният програмен интерфейс доставя значителна част от функционалността на графичното приложение. Това може да включва модифицирането на съществуващи данни като геометрии, трансформации, йерархии, взаимовръзки между обекти в сцената и др. Също така позволява създаването на нови обекти, такива като типове светлини или да позволи работа с приложни програмни интерфейси като OpenGL и DirectX.

Приложният програмен интерфейс и скриптовият език в една графична система не са взаимно изключващи се. Понякога много редове код на езика от високо ниво на ИРП съществуват като една скрипт-команда, така че в този случай е по-добре да се използва СЕ. Но в други случаи е необходимо да се разработи приложение, което се базира на класове и методи от ИРП, недостъпни за СЕ. Препоръчва се [29] използването на СЕ за това, което е възможно да се направи чрез него и след това използването на ИРП.

4.2. Избор на 3D графична система за моделиране, притежаваща средства за функционално разширяване

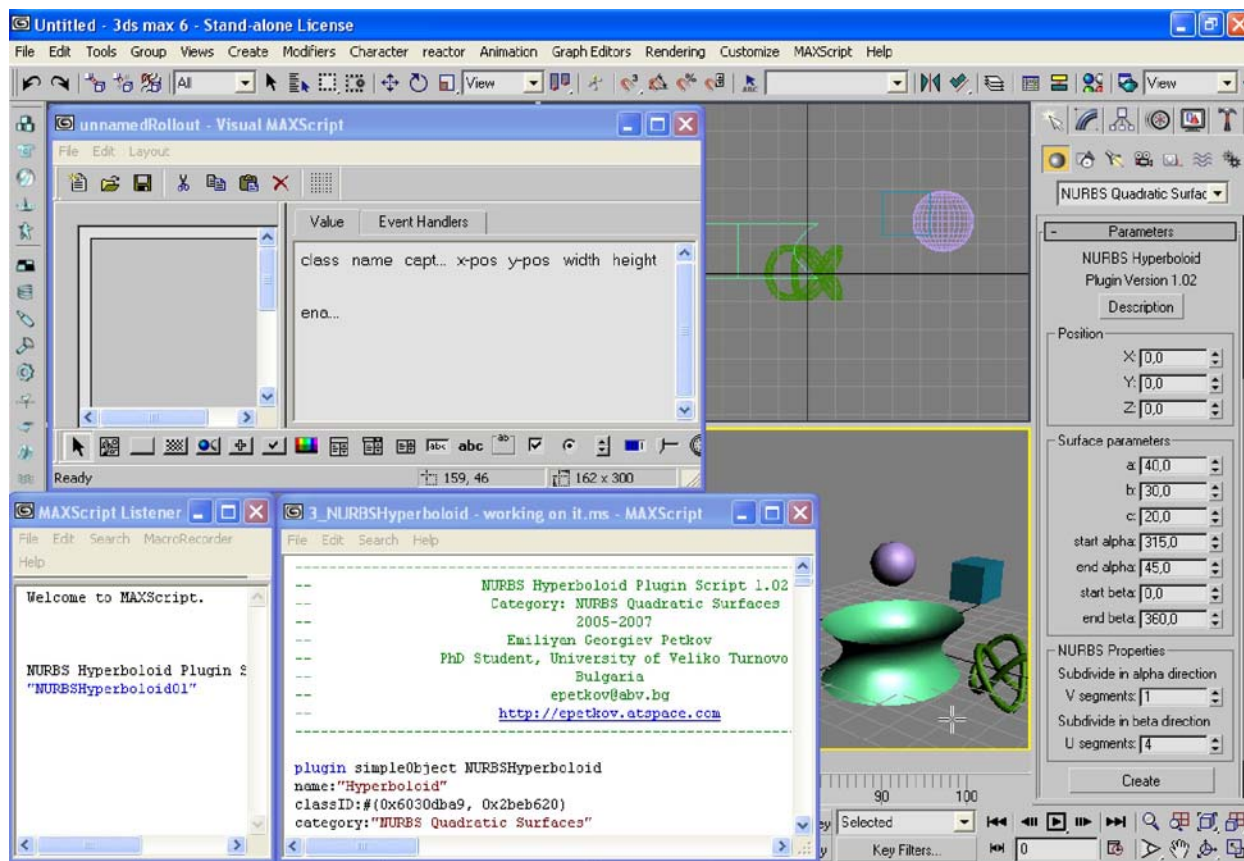
Основно изискване което поставяме към графичната система, в която да бъдат приложени моделите на NURBS кривите и повърхнините, е тя да притежава NURBS апарат в приложния си програмен интерфейс и скриптовия език. Това се обуславя от факта, че моделите, които се предлагат, се базират на NURBS базиса и основните функции за работа с NURBS обекти. Друго важно условие е графичната система да притежава средства, които да оползотворяват свойствата на моделите на кривите и повърхнините, които се прилагат.

Сравнителни характеристики на някои от основните 3D графични системи и техните приложения могат да бъдат намерени в [122, 129]. Между тях са: 3ds Max, (Autodesk), Maya (Autodesk), Cinema 4D (MAXON), Realsoft 3D (Realsoft Graphics), Blender (Blender Foundation), EIAS (EI Technology Group), form-Z (Autodessys Inc.), Houdini (Side Effects Software), Softimage|XSI (Avid), SolidThinking (SolidThinking Ltd), TrueSpace (Caligari Corporation). Споменатите приложения притежават възможности за създаване на NURBS обекти и моделиране с тях, както и възможности за функционално разширяване.

За демонстрационно прилагане на предложените NURBS модели на квадратичните криви и повърхнини беше избрана 3D графичната система Autodesk 3ds Max. Изборът бе направен с отчитане на нейните особености.

Autodesk 3ds Max е мощна, цялостна и добре проектирана 3D графична система за моделиране, анимация и рендериране (фиг. 4.1) [3, 20, 111, 116, 117, 133]. Тя

осигурява на архитекти, проектанти, графични художници и специалисти по визуализация силата да подсигури визуално процеса на моделиране, концептуалното изследване, моделната ратификация и визуалната комуникация. Това се постига чрез интуитивния потребителски интерфейс и помощни инструменти. Тя осигурява и съвместимост с други широко използвани графични приложения като Adobe Photoshop, Adobe Illustrator, AutoCAD, Revit Architecture и др., позволявайки на потребителите бързо и удобно да направят 3ds Max част от тяхната графична продукция.



Фиг. 4.1 Интерфейс на 3D графичната система Autodesk 3ds Max.

В 3ds Max ефективно се създават параметрични форми и обекти. Има няколко типа геометрични примитиви, с които се моделира: poly, mesh, patch, spline и NURBS. Графичните примитиви (наречени *геометрии*) са разделени в няколко групи: стандартни примитиви, разширени примитиви, 2D форми и архитектурни елементи.

В 3ds Max работата с NURBS криви и повърхнини е на изключително високо ниво. NURBS кривите са два вида: Point curve – контролира се от точки, които лежат на кривата и CV curve – контролира се от контролни върхове, които не лежат задължително на кривата. Аналогично на кривите, NURBS повърхнините са два вида Point surface и CV surface. NURBS кривите и повърхнините са основата за създаването

на NURBS възли (от англ. nodes) (NURBS обектите в сцената). NURBS възлите се редактират чрез точките или контролните върхове на кривите и повърхнините, които са част от тях. Съществува богат набор от функции за създаване на NURBS възли чрез точки, криви или повърхнини.

Повърхнините могат да бъдат свързани, отделени, подравнени, зашити заедно, разширени, с висока степен на контрол върху параметризацията и непрекъснатостта. Сплайн-модификаторите на системата позволяват деформиране, нормализиране на сплайна, задаване на параметрите на сплайна свързани с рендерирането, завъртане и намиране на сечение. Една част от модификаторите са разработени за директна работа върху подобектни геометрии (например Curve CV, Surface CV, surface).

В 3ds Max се реализира бърза сканираща-линия (scanline) и рендъринг за висококачествена продукция. Тя притежава пълен набор от ефекти, няколко вида светлинни източници (включително фотометрично осветление), атмосферни ефекти, изчисление на сенки, материали, библиотеки от текстури и създаване на системи от частици, които доставят всички нужни компоненти за създаването на реалистични изображения и анимация.

Autodesk 3ds Max разполага с ИПП, който се нарича MaxSDK [112]. MaxSDK е обектно-ориентирана библиотека за създаване на CCM за системата. MaxSDK доставя напълно изчерпателно множество от класове, които разработчиците могат да съчетават и разширяват за да създават разнообразни CCM (плъгини).

Autodesk 3ds Max разполага и със скриптов език, който се нарича MaxScript. Той е специално разработен за да е възможно разширяване на системата [29]. Той може да бъде полезен по много начини: автоматизиране на повтарящи се задачи, създаване на стандартни процедури, импортиране и експортиране на данни, създаване на плъгини. Той може да отговаря и за много важните характеристики на програмата, и освен това да бъде част от потребителския интерфейс. MaxScript е обектно ориентиран език за програмиране и е C++ подобен. MaxScript съдържа обектно ориентиран приложен програмен интерфейс, предоставящ цялата функционалност и мощ на скриптовите приложения.

Autodesk 3ds Max осигурява интерфейс за работа със скриптовия език. Това включва: прозорец за въвеждане на командите, подпрограма за записване на всяко действие под формата на скрипт, подпрограма за въвеждане на командите, извеждане на резултатите от функциите и предоставяща възможност за разглеждане на записаните действия, редактор на скриптове и Visual MaxScript.

Visual MaxScript осигурява лесен начин за създаване на потребителски интерфейс на скрипт, подобно на инструментите във Visual Basic, Visual C++ и други визуални

среди за програмиране. Също като инструментите на споменатите езици, Visual Script е “интерфейс за създаване на интерфейси”.

Чрез MaxScript могат да бъдат създавани NURBS обекти. Диаграма на дървовидната структура на основните класове в NURBS API е представена в MaxScript Reference на 3ds Max. Главният базов клас е NURBSObject, но и класовете NURBSPoint, NURBSCurve, NURBSSurface, NURBSCVCurve и NURBSPointCurve също функционират като базови класове.

Разгледаните до тук особености на 3D графичната система Autodesk 3ds Max определят основните ѝ преимущества, касаещи настоящето изследване:

- предоставя отлични възможности за работа с NURBS примитиви;
- притежава разнообразни средства за моделиране;
- предлага отлични възможности за рендериране;
- позволява реализация на широк кръг от задачи;
- дава възможност за функционално разширяване на системата чрез скриптов език, който в приложния си програмен интерфейс съдържа класове за работа с NURBS обекти.

4.3. Разработване на специализираните модули

NURBS моделите на квадратичните криви и повърхнини трябва бъдат приложени като нови графични примитиви за моделиране в Autodesk 3ds Max. За целта са разработени софтуерни модули, които представляват инструменти за създаване на новите примитиви в системата. Модулите притежават следните основни характеристики:

- всеки един представлява един графичен примитив (една крива или повърхнина);
- притежават графичен потребителски интерфейс, чрез който да се задават параметрите на кривите и повърхнините от параметричното им представяне и параметрите на модела им за дизайн;
- предоставят възможност за интерактивно построяване и моделиране на кривите и повърхнините в реално време;
- могат да построяват кривите и повърхнините във всяка една от координатните равнини и в ориентация в зависимост от избрания изглед.

4.3.1. Избор на софтуерен инструментариум

От раздел 4.2. следва, че 3ds Max притежава две възможности за разработване на специализирани модули MaxSDK и MaxScript. И двата инструмента притежават

нужните средства за да се разработят модулите според изискванията поставени по-горе. Ако модулите бъдат разработени на MaxSDK обаче, за всяка нова версия на програмата те трябва да бъдат коригирани и прекомпилирани за да могат да се изпълняват в системата. Освен това, те отново трябва да бъдат разпространени и инсталирани от потребителите. Ако те бъдат разработени на MAXScript за по-ниска версия на 3ds Max (например версия 4.0), тогава те ще могат да се изпълняват и работят коректно на всяка по-горна версия на програмата (например версия 9.0). Изхождайки от факта, че версиите на програмата излизат сравнително често (всяка година), по-ефективно е специализираните модули да бъдат разработени на MaxScript.

В подкрепа на този избор може да се има предвид също, че разработването на специализирани модули чрез MaxSDK изисква средата за разработка на приложения MS Visual C++, която се предлага сравнително на висока цена. Следователно, от икономическа гледна точка, е по-целесъобразно да се ползва MaxScript, който е част от системата.

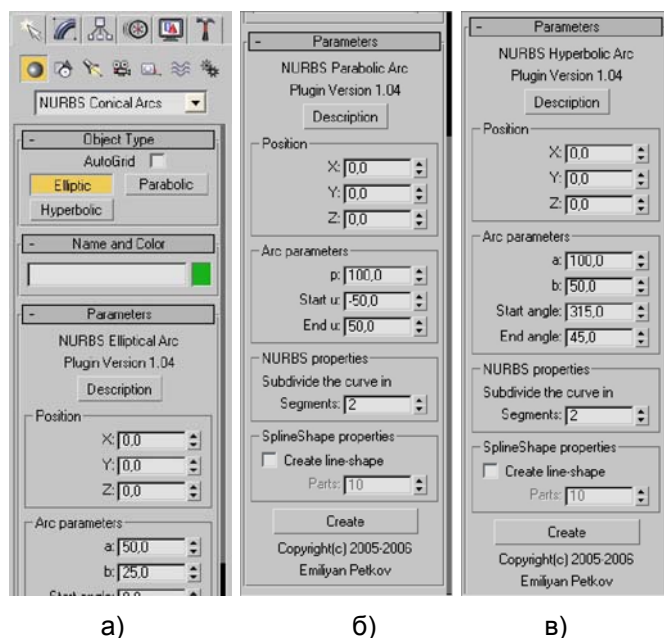
Съществуват няколко вида скрипт-модули. Поради изискването всеки един графичен инструмент да представя един графичен примитив за моделиране, избираме тип плъгин-скрипт (plugin-script). Плъгин-скриптовете се интегрират в потребителския интерфейс и по този начин стават част от вградения набор от групи от графични примитиви. Те се разделят условно на три вида: разширителни – за разширяване на съществуващи плъгини, системни – за създаване на възли и плъгин-скриптове за създаване на нови плъгини, които създават нови графични примитиви.

Прилагането на предложените в Глава II и Глава III NURBS модели на квадратичните криви и повърхнини изисква разработването на нови инструменти, които да представят нови графични примитиви. Следователно разработените плъгин-скриптове ще са от третия вид, а именно плъгини за създаване на нови графични примитиви.

4.3.2. Графичен потребителски интерфейс на CCM

Прилагаме NURBS моделите на кривите от втора степен предложени в Глава II в 3ds Max като CCM, а именно: представянето на дъги от елипси чрез NURBS криви от вида даден с уравнение (2.19) – реализира се чрез Алгоритъм 2.2; представянето на дъги от параболи чрез NURBS криви от втора степен – реализацията следва директно от Твърдение 2.1 и подхода за подразделяне даден с Алгоритъм 2.3; представянето на дъги от хиперболи чрез NURBS криви от втора степен – реализацията следва директно от Твърдение 2.5 и подхода за подразделяне даден с Алгоритъм 2.3.

Тези три NURBS модела се реализират в три отделни CCM: NURBS Elliptical Arc (фиг. 4.2а), NURBS Parabolical Arc (фиг. 4.2б) и NURBS Hyperbolical Arc (фиг. 4.2в). Те съставляват една група наречена NURBS Conical Arcs (фиг. 4.2а).

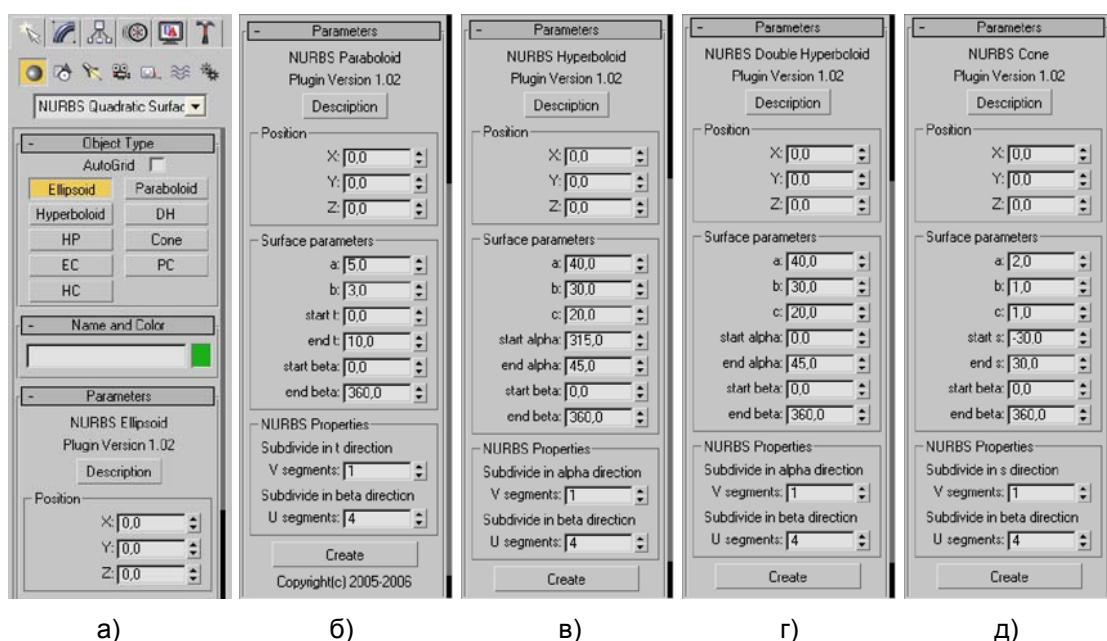


Фиг. 4.2 Графичния потребителски интерфейс на CCM:

а) NURBS Elliptical Arc, б) NURBS Parabolical Arc и в) NURBS Hyperbolical Arc.

Прилагаме NURBS моделите на повърхнините от втора степен предложени в Глава III в 3ds Max като CCM, а именно: представянето на елипсоиди чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.27) – реализира се по подхода даден в точка 3.2.1.2.; представянето на параболоид чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.28) – реализира се по подхода даден в точка 3.2.1.3.; представянето на хиперболоид чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.29) – реализира се по подхода даден в точка 3.2.1.4.; представянето на двоен хиперболоид чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.30) – реализира се по подхода даден в точка 3.2.1.5.; представянето на конус чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.31) – реализира се по подхода даден в точка 3.2.1.6.; представянето на елиптичен цилиндър чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.37) – реализира се по подхода даден в подраздел 3.2.2.; представянето на параболичен цилиндър чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.37) – реализира се по подхода даден в подраздел 3.2.2.; представянето на хиперболичен цилиндър чрез биквадратни NURBS повърхнини от вида даден с уравнение (3.37) – реализира се по подхода даден в подраздел 3.2.2.; представянето на хиперболичен параболоид чрез

биквадратни NURBS повърхнини от вида даден с уравнение (3.40) с контролни върхове (3.41) – реализира се по подхода даден в подраздел 3.2.3.



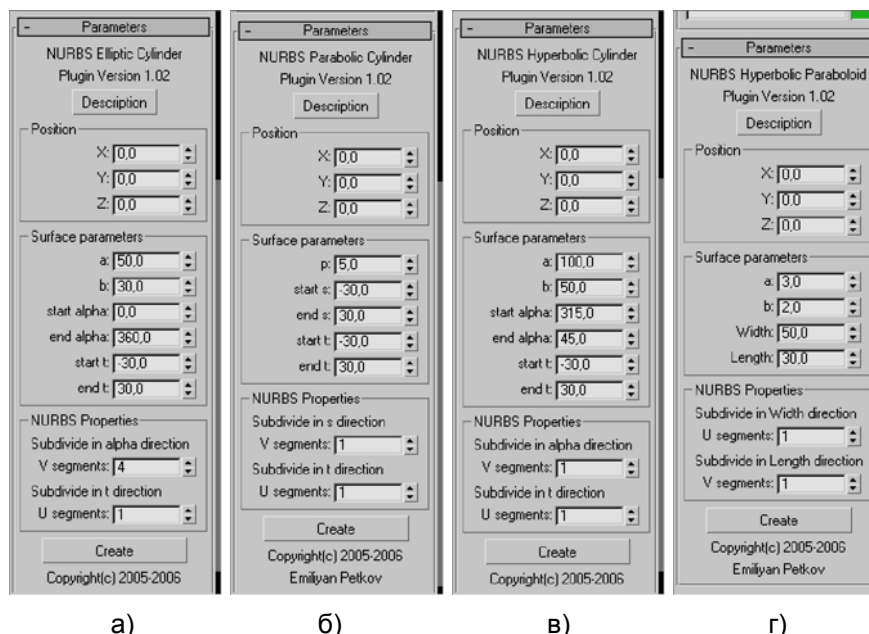
Фиг. 4.3 Графичния потребителски интерфейс на CCM:

а) NURBS Ellipsoid, б) NURBS Paraboloid, в) NURBS Hyperboloid, г) NURBS Double Hyperboloid и д) NURBS Cone.

Тези девет NURBS модела се реализират в девет отделни CCM: NURBS Ellipsoid (фиг. 4.3а), NURBS Paraboloid (фиг. 4.3б), NURBS Hyperboloid (фиг. 4.3в), NURBS Double Hyperboloid (фиг. 4.3г), NURBS Cone (фиг. 4.3д), NURBS Elliptic Cylinder (фиг. 4.4а), NURBS Parabolic Cylinder (фиг. 4.4б), NURBS Hyperbolic Cylinder (фиг. 4.4в) и NURBS Hyperbolic Paraboloid (фиг. 4.4г). Те съставляват една група наречена NURBS Quadratic Surfaces (фиг. 4.3а).

След зареждането на CCM (автоматично или ръчно) в 3ds Max, двете групи от плъгини NURBS Conical Arcs и NURBS Quadratic Surfaces се добавят в полето Create, категорията Geometry (фиг. 4.2а и 4.3а). При избор от страна на потребителя на една от двете групи се зареждат бутоните за стартиране на плъгините от тази група. Всеки CCM е с отделен графичен интерфейс.

Използват следните контроли за създаването на елементите на графичния потребителски интерфейс: button (бутон), spinner (числово поле), checkbox (поле за отмятане), messagebox (прозорец за съобщение), rollout (раздел) и progressbar (прогрес-контрол).



Фиг. 4.4 Графичния потребителски интерфейс на CCM:
а) NURBS Elliptic Cylinder, б) NURBS Parabolic Cylinder,
в) NURBS Hyperbolic Cylinder и г) NURBS Hyperbolic Paraboloid.

Параметрите на всеки отделен NURBS примитив до голяма степен определят структурата на графичния интерфейс на всеки CCM. За всеки обект, който се създава, интерфейсът трябва да има следните раздели от параметри: Position (позиция) – параметри, които определят позицията на обекта спрямо опорната му точка (pivot point) в пространството, Arc parameters или Surface parameters (съответно на дъгите или повърхнините) – параметри (от параметричните представяния), които задават дъгата (при кривите) или повърхнината, която трябва да се построи, NURBS properties (NURBS характеристики) – параметри, чрез които се задава броя на сегментите и бутон Create (създай) – бутон, чрез който се стартират функциите за създаване на NURBS модела в сцената.

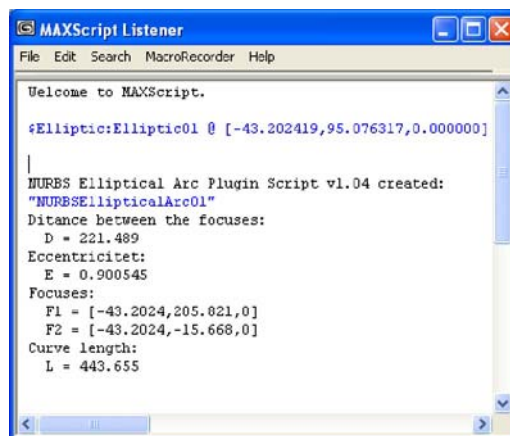
Стойностите на параметрите се задават в числови полета, които могат да приемат цели или реални числа. Тези стойности трябва да бъдат ограничени до допустимите стойности при определянето на съответната крива или повърхнина.

Използването на прозорци за съобщения се налага при възникване на определени събития-изключения в плъгина, когато потребителят трябва да бъде уведомен за нещо или да коригира своите действия.

Плъгините от групата NURBS Conical Arcs разполагат с поле за отмятане в случай, че е необходимо създаването не само на NURBS крива, но и на line-shape форма. Създаването обаче на такава форма изисква повече време, затова в този момент се визуализира прогрес-контрол.

Всеки плъгин разполага с бутон “Create”, чрез който се създава кривата или повърхнината по зададените стойности на параметрите от разделите. Това е алтернативен метод на метода (разгледан по-долу в този раздел) за интерактивно построяване на NURBS обект в пространството.

Стойностите на параметрите (дължина на крива [64, 69, 103, 104], позиции на фокуси, разстояние между фокуси, ексцентрицитет и др.), свързани със специфичните свойства на конкретния обект, се извеждат в прозореца MaxScript Listener (фиг. 4.5).



Фиг. 4.5 Прозореца MaxScript Listener.

4.3.3. Функции създаващи новите NURBS модели в CCM

За създаването на NURBS моделите в CCM са разработени набор от функции. Разделени условно на две групи. Първата се състои от функциите, които определят основните контролни върхове, възловите вектори и всички контролни върхове на кривата или повърхнината. Втората група се състои от функциите, които по получените вече данни за кривата или повърхнината създават NURBS обекта в сцената.

Първа група за NURBS кривите:

ComputeTheUnitCircleArc	Намира NURBS дъга от единичната окръжност. Намира се в плъгина NURBS Elliptical Arc.
ComputeTheEllipticalArc	Намира търсената NURBS елиптична дъга. Намира се в плъгина NURBS Elliptical Arc.
ComputeTheKnotVectorValues	Намира възловия вектор. Намира се в плъгините NURBS Elliptical Arc, NURBS Parabolical Arc и NURBS Hyperbolical Arc.
ComputeTheMainControlPoints	Намира основните контролни върхове на NURBS дъгата. Намира се в плъгините NURBS Parabolical Arc и NURBS Hyperbolical Arc.
ComputeAllCotrolPoints	Намира всички контролни върхове на NURBS дъгата. Намира се в плъгините NURBS Parabolical Arc и NURBS Hyperbolical Arc.

Втора група за NURBS кривите:

SpecifyNURBSNodeName	Определя името на NURBS обекта в сцената.
Create***NURBSCurve	Създава NURBS кривата като възел (node).
CreateNURBS***Arc	Създава NURBS кривата в сцената.
*** - заменя се с Elliptic, Parabolic и Hyperbolic в зависимост от плъгина.	Намират се в плъгините на всички NURBS криви.

Първа група за NURBS повърхнините:

ComputeTheUnitCircleArc	Намира NURBS дъга от единичната окръжност. Намира се в плъгина NURBS Ellipsoid и NURBS Elliptical Cylinder.
ComputeTheGeneratrixKnotVectorValues	Намира възловия вектор на генераторната крива. Намира се в плъгина NURBS Ellipsoid и NURBS Elliptical Cylinder.
ConstructTheUnitSphere	Намира част от единичната сфера в NURBS. Намира се в плъгина NURBS Ellipsoid.
ScaleTheUnitSphereToTheEllipsoid	Мащабира NURBS частта от единичната сфера до търсения NURBS елипсоид. Намира се в плъгина NURBS Ellipsoid.
ComputeTheKnotVectorsOfTheRevolved Surface	Намира възловите вектори на ротационната повърхнина. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid и NURBS Cone.
ComputeTheMainControlPoints	Намира основните контролни върхове на NURBS генераторната крива. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid, NURBS Cone, NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder.
ComputeTheKnotVectorValues	Намира възловия вектор на NURBS генераторната крива. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid и NURBS Cone.
ComputeAllCotrolPoints	Намира всички контролни върхове на NURBS генераторната крива. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid, NURBS Cone, NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder.
ConstructTheRevolved***	Построява ротационната NURBS повърхнина. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid и NURBS Cone.
ScaleTheRevolved***ToTheElliptical***	Мащабира ротационната NURBS повърхнина до търсената. Намира се в плъгините NURBS Paraboloid, NURBS Hyperboloid, NURBS Double Hyperboloid и NURBS Cone.
ComputeTheKnotVectorsOfTheExtruded Surface	Намира възловите вектори на заметената повърхнина. Намира се в плъгините NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder.
ConstructTheExtrudedNURBSSurface	Построява NURBS заметената повърхнина. Намира се в плъгините NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder.
ComputeTheGeneratrixKnotVectorValues	Намира възловия вектор на NURBS генераторната крива. Намира се в плъгините

	NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder.
ComputeTheBaseCVs	Намира основните контролни върхове на NURBS повърхнината. Намира се в плъгина NURBS Hyperbolic Paraboloid.
CalculateTheNumberOfCVsAndKnots	Намира броя на контролните върхове и възли по зададените стойности за подразделяне на NURBS повърхнината. Намира се в плъгина NURBS Hyperbolic Paraboloid.
ComputeTheUKnotVectorValues	Намира възловия вектор за u направление. Намира се в плъгина NURBS Hyperbolic Paraboloid.
ComputeTheVKnotVectorValues	Намира възловия вектор за v направление. Намира се в плъгина NURBS Hyperbolic Paraboloid.
SubdivideTheSurfaceOnU	Подразделя NURBS повърхнината в u направление. Намира се в плъгина NURBS Hyperbolic Paraboloid.
SubdivideTheSurfaceOnV	Подразделя NURBS повърхнината във v направление. Намира се в плъгина NURBS Hyperbolic Paraboloid.
*** - заменя се с Paraboloid, Hyperboloid, DoubleHyperboloid и Cone в зависимост от плъгина.	

Втора група за NURBS повърхнините:

SpecifyNURBSNodeName	Определя името на NURBS обекта в сцената.
CreateTheNURBSSurface	Създава NURBS повърхнината като възел (node).
CreateNURBS***Surface	Създава NURBS повърхнината в сцената.
SetNURBSNodeMaterial	Задава материал на NURBS обекта в сцената.
*** - заменя се с Ellipsoid, Paraboloid, Hyperboloid, DoubleHyperboloid, Cone, Elliptic Cylinder, Parabolic Cylinder, Hyperbolic Cylinder и Hyperbolic Paraboloid в зависимост от плъгина.	Намират се в плъгините на всички NURBS повърхнини.

Функциите CreateNURBS***Arc създават NURBS кривите и функциите CreateNURBS***Surface създават NURBS повърхнините. Следователно тези функции трябва да бъдат извикани когато потребителят въведе стойностите на параметрите от графичния потребителски интерфейс и натисне бутона "Create". Този бутон е свързан с манипулатора на събитие on but_create pressed do, чийто код се изпълнява след натискането на бутона. Именно тук са поставени CreateNURBS***Arc и CreateNURBS***Surface.

4.3.4. Интерактивно създаване на NURBS обектите

Освен чрез въвеждане на стойности на параметрите от графичния интерфейс и натискане на бутона "Create", плъгините разполагат и с механизъм за получаване на

данни от мишката, така че с нейна помощ да се зададат стойности на някои от параметрите и да се създаде NURBS обекта. В кода на скрипта това се прави с помощта на командата `tool` и две основни действия `on mousePoint clickNo do` и `on mouseMove clickNo do`. Тези действия определят как ще се държи инструментът, когато потребителя щракне и влачи с мишката.

Манипулаторите на събитията `on mousePoint clickNo` определят какво ще направи инструментът в ситуациите, когато бъде щракнато с мишката с ляв бутон за първи път и когато се отпусне бутона на мишката за последен път. Манипулаторите на събитията `on mouseMove clickNo` определят какво ще направи инструмента, когато мишката бъде премествана. Например: когато мишката се влачи след щракването с ляв бутон и когато мишката се влачи след като потребителят е отпуснал бутона.

NURBS кривите се построяват чрез мишката с едно натискане на ляв бутон, преместване на мишката и след това отпускане на бутона. Това определя стойностите на два параметъра. При дъгите от елипса това са параметрите a и b (в уравнение(2.4)). При дъгите от парабола това са p и u (в уравнение(2.6)). При дъгите от хипербола това са a и b (в уравнение(2.8)). Останалите параметри трябва да бъдат зададени в графичния интерфейс предварително.

NURBS повърхнините се построяват чрез мишката с едно натискане на ляв бутон, преместване на мишката, отпускане на бутона, отново преместване на мишката и след това натискане отново на ляв бутон. В повечето случаи това определя стойностите на три параметъра. Единствено за конуса имаме допълнението: отново преместване на мишката и след това натискане отново на ляв бутон. Това определя стойностите на четири параметъра.

При построяването на: части от елипсоид параметрите, които се определят чрез мишката са a , b и c (в уравнение(3.6)); при параболоида това са a , b и u (в уравнение(3.17)); при хиперболоида – a , b и c (в уравнение(3.8)); при двойния хиперболоид – a , b и c (в уравнение(3.10)); при конуса – a , b , $start\ u$ и $end\ u$ (в уравнение(3.12)); при елиптичния цилиндър – a , b (в уравнение(3.32)) и $end\ t$; при параболичния цилиндър – p , $start\ s$, $end\ s$ (в уравнение(3.34)) и $end\ t$; при хиперболичния цилиндър – a , b (в уравнение(3.33)) и $end\ t$; при хиперболичния параболоид – a , b (в уравнение(3.19)), $width$ и $length$. Значението на допълнителните параметри е изяснено при представянето на повърхнините чрез NURBS в раздел 3.2.

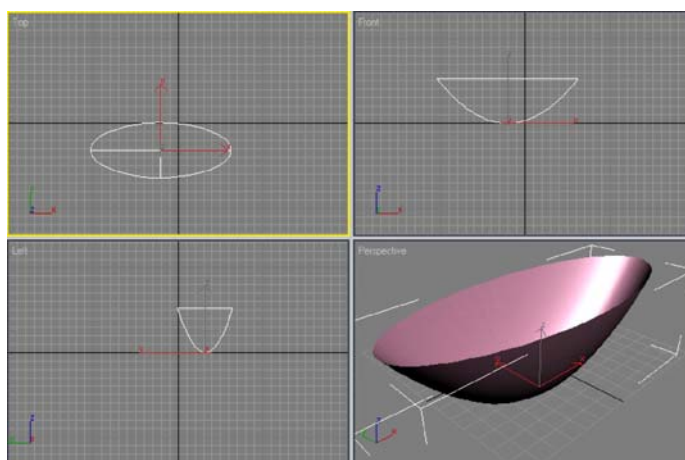
4.3.5. Създаване на NURBS обектите в различните изгледи

NURBS моделите на кривите и повърхнините предложени в Глава II и Глава III точно представят квадратичните криви и повърхнини дадени в техните канонични представяния в координатна система $K = Oxyz$. Моделите на кривите се дават с координата $z = 0$ на контролните им върхове, т.е. кривите лежат в равнината Oxy . Моделите на повърхнините (условно казано) се построяват около оста Oz .

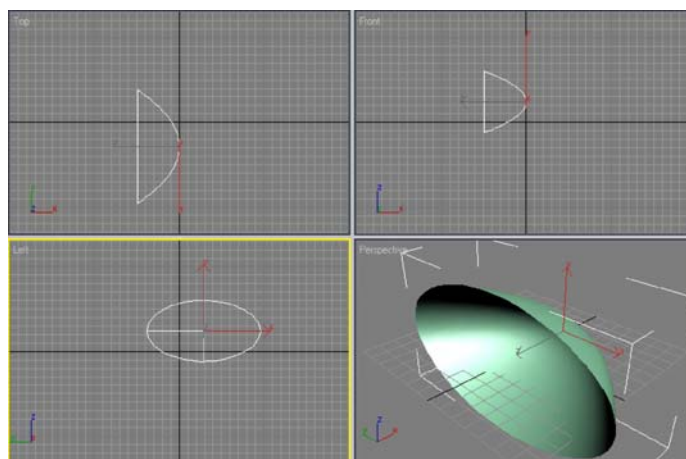
В 3ds Max са дефинирани посоки в пространството (сцената) като т. $O(0,0,0)$ е отправната точка. Оста Oz сочи нагоре (top), а отрицателните стойности са надолу (bottom). Оста Ox сочи надясно (right), а отрицателните стойности са наляво (left). Оста Oy сочи назад (back), а отрицателните стойности са напред (front). Така се определят шестте ортогонални проекции и съответните им изгледи top, bottom, right, left, back и front. Перспективата се осъществява в изгледа perspective.

Твърде ограничаващо ще бъде ако обектите се създават (в координатното пространство) само в една единствена позиция. По-удобно е ако графичния инструмент позволява построяване в шестте направления. Това може да се осъществи като се определи в кой изглед потребителят работи в момента и иска там да бъде създаден обекта. След това плъгинът построява кривата или повърхнината в избрания изглед като координатната система на обектите се взема по правилото: оста x е надясно в изгледа, оста y е нагоре в изгледа и оста z сочи напред от изгледа (сочи към потребителя). Ако обектът се създава в изгледа top или perspective, тогава координатната система на обекта и на пространството съвпадат.

На фигури 4.6 и 4.7 е показано построяването на NURBS Paraboloid в изгледите top и left, съответно.



Фиг. 4.6 Построяване на NURBS Paraboloid в изгледа top (горния ляв изглед).



Фиг. 4.7 Построяване на NURBS Paraboloid
в изгледа left (долния ляв изглед).

Функцията SetDirection се грижи за правилното построяване на обектите от плъгините в зависимост от избрания изглед.

4.3.6. Допустими стойности на параметрите на CCM

Стойностите на параметрите, чрез които се определят дъгите и повърхнините, се ограничават от допустимите стойности дадени със скаларно-параметричните им уравнения. В реализацията на NURBS моделите и при реалното построяване на кривите и повърхнините чрез тях е нужно да се дефинират допустимите стойности, за които CCM работят коректно и обектите се построяват точно. Тези допустими стойности могат да бъдат само подинтервали на интервалите от допустими стойности дадени със скаларно-параметричните уравнения на квадратичните криви и повърхнини.

Например, параметрите a , b и c в СПУ на елипсата трябва да са положителни числа. Възникват следните въпроси:

Какви минимални и максимални стойности могат да приемат те при реалното построяване на кривата? Възможно ли е да могат да приемат всякакви положителни числа?

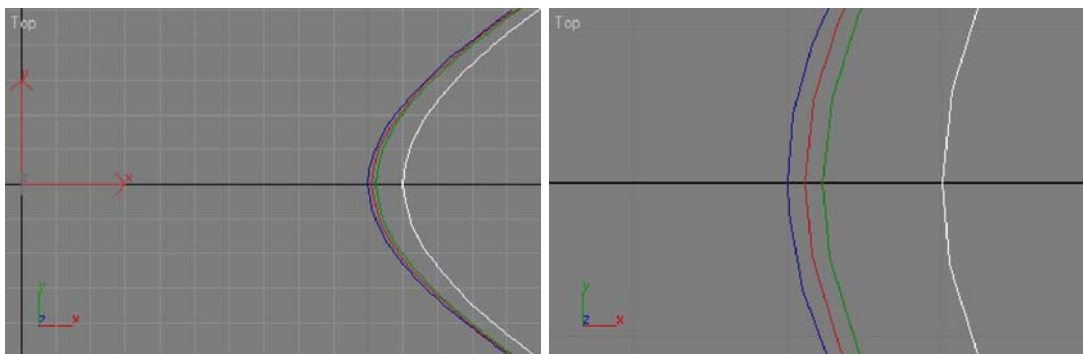
В това отношение, стойностите трябва да бъдат съобразени със системата, в която те ще се въвеждат. В графичната система 3ds Max съществува ограничението (касаещо координати на точки, позиции, размери): за числови полета да се въвеждат стойности в интервала $[0.001; 100,000,000.0]$. Следователно трябва да поставим параметрите a , b и c в този интервал.

Има друг случай обаче, в който не може да се използва ограничението на системата. Например това се вижда при хиперболата, при която параметъра

$\alpha \in [0; 2\pi], \alpha \neq \frac{\pi}{2}, \frac{3\pi}{2}$. В тези случаи можем да действаме по принципа: проверка на работата на модулите в краищата на интервалите, близко до непозволени стойности, в критични стойности (например 0) и в произволни междинни стойности от интервала. Ако за някои стойности резултатите са неточни, тогава се търси най-близката стойност за която те са точни.

Нека разгледаме построяването на дъга от хипербола чрез NURBS крива със стойности $a = 100$, $b = 50$, $segments = 2$, α_s близо до $\frac{\pi}{2}$ и α_e близо до $\frac{3\pi}{2}$. Построяваме кривата в изгледа top. Тогава координатите на върха на хиперболата за тези стойности на a и b трябва да са (100, 0, 0). Изпълняваме плъгина четири пъти с различни стойности за α_s и α_e (в градуси), и следим координатите на върха на NURBS дъгата от хипербола, която получаваме (фиг. 4.8):

- 1) $\alpha_s = 270.1^\circ$ и $\alpha_e = 89.9^\circ \rightarrow (110, 0, 0)$, с размер на NURBS кривата в системата по x : 57293.5 и по y : 57293.5;
- 2) $\alpha_s = 270.5^\circ$ и $\alpha_e = 89.5^\circ \rightarrow (102.3, 0, 0)$, с размер на NURBS кривата в системата по x : 11459.4 и по y : 11459.4;
- 3) $\alpha_s = 271.0^\circ$ и $\alpha_e = 89.0^\circ \rightarrow (101.1, 0, 0)$, с размер на NURBS кривата в системата по x : 5729.9 и по y : 5729.9;
- 4) $\alpha_s = 275.0^\circ$ и $\alpha_e = 85.0^\circ \rightarrow (100.1, 0, 0)$, с размер на NURBS кривата в системата по x : 141.4 и по y : 141.4.



Фиг. 4.8 Върха на дясна част на хипербола представена в NURBS крива чрез плъгина NURBS Hyperbolical Arc за стойности на α : 1) $\alpha_s = 270.1^\circ$ и $\alpha_e = 89.9^\circ$ - в бял цвят; 2) $\alpha_s = 270.5^\circ$ и $\alpha_e = 89.5^\circ$ - в зелен цвят; 3) $\alpha_s = 271.0^\circ$ и $\alpha_e = 89.0^\circ$ - в червен цвят; 4) $\alpha_s = 275.0^\circ$ и $\alpha_e = 85.0^\circ$ - в син цвят.

Резултатите показват, че при много близки стойности на α_s до $\frac{\pi}{2}$ и α_e до $\frac{3\pi}{2}$ се получават не съвсем точни стойности за позицията на върха на получената NURBS дъга от хипербола. Но тези неточности са съпоставими с размера на кривата. Оказва се, че неточността расте право пропорционално на увеличаването на размера на NURBS кривата. Например, за случай 1) грешката се увеличава 5 пъти спрямо тази в случай 2), но и размера на кривата се увеличават 5 пъти.

Резултатите от направеното изследване за откриване на интервалите от допустими стойности за параметрите на CCM са поместени в таблици от 4.1 до 4.4, в които се използват константите: $\text{eps} = 0.001$, $\text{buf} = 1.0$, $\text{par_max} = 1,000,000.0$, $\text{pos_max} = 100,000,000.0$.

Таблица 4.1 Допустими стойности на параметрите на плъгините от групата NURBS Conical Arcs

Пара- метри	NURBS елиптична дъга	NURBS параболична дъга	NURBS хиперболична дъга
a	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
b	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
α_s	[-par_max; par_max]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
α_e	[-par_max; par_max]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
p	-	[0 + eps; par_max]	-
u_s	[-90; 90]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
u_e	[-90; 90]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
X	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Y	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Z	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
<i>Segments</i>	[1; 100]	[1; 100]	[1; 100]
<i>Parts</i>	[1; 100]	[1; 100]	[1; 100]

Таблица 4.2 Допустими стойности на плъгините NURBS Ellipsoid, NURBS Paraboloid и NURBS Hyperboloid от групата NURBS Quadratic surfaces

Пара- метри	NURBS елипсоид- повърхнина	NURBS параболоид- повърхнина	NURBS хиперboloид- повърхнина
a	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
b	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
c	[0 + eps; par_max]	-	[0 + eps; par_max]

u_s	[-90; 90]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
u_e	[-90; 90]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
t_s	-	[0; par_max]	-
t_e	-	[0; par_max]	-
v_s	[-par_max; par_max]	[-par_max; par_max]	[-par_max; par_max]
v_e	[-par_max; par_max]	[-par_max; par_max]	[-par_max; par_max]
X	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Y	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Z	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
V	[1; 100]	[1; 100]	[1; 100]
U	[1; 100]	[1; 100]	[1; 100]

Таблица 4.3 Допустими стойности на пългините NURBS Double Hyperboloid, NURBS Cone и NURBS Hyperbolic Paraboloid от групата NURBS Quadratic surfaces

Пара- метри	NURBS двоен- хиперболоид- повърхнина	NURBS конус- повърхнина	NURBS хиперболичен- параболоид-повърхнина
a	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
b	[0 + eps; par_max]	[0 + eps; par_max]	[0 + eps; par_max]
c	[0 + eps; par_max]	[0 + eps; par_max]	-
u_s	[0; 90 - buf]	-	-
u_e	[0; 90 - buf]	-	-
$Width$	-	-	[0 + eps; par_max]
$Length$	-	-	[0 + eps; par_max]
v_s	[-par_max; par_max]	-	-
v_e	[-par_max; par_max]	-	-
s_s	-	[-par_max; par_max]	-
s_e	-	[-par_max; par_max]	-
X	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Y	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Z	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
V	[1; 100]	[1; 100]	[1; 100]
U	[1; 100]	[1; 100]	[1; 100]

Таблица 4.4 Допустими стойности на пългините NURBS Elliptic Cylinder, NURBS Parabolic Cylinder и NURBS Hyperbolic Cylinder от групата NURBS Quadratic surfaces

Пара- метри	NURBS елиптичен- цилиндър-повърхнина	NURBS параболичен- цилиндър-повърхнина	NURBS хиперболичен- цилиндър-повърхнина
a	[0 + eps; par_max]	-	[0 + eps; par_max]
b	[0 + eps; par_max]	-	[0 + eps; par_max]
n	-	[0 + eps; par_max]	-
α_s	[-par_max; par_max]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]

α_e	[-par_max; par_max]	-	[270 + buf; 360][0; 90 - buf] or [90 + buf; 270 - buf]
S_s	-	[-par_max; par_max]	-
S_e	-	[-par_max; par_max]	-
t_s	[-par_max; par_max]	[-par_max; par_max]	[-par_max; par_max]
t_e	[-par_max; par_max]	[-par_max; par_max]	[-par_max; par_max]
X	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Y	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
Z	[-pos_max; pos_max]	[-pos_max; pos_max]	[-pos_max; pos_max]
V	[1; 100]	[1; 100]	[1; 100]
U	[1; 100]	[1; 100]	[1; 100]

4.4 Повдигане на степента на NURBS кривите и повърхнините

Разнообразните приложения на NURBS кривите и повърхнините налагат използването им с различни степени на базисните им функции. В графичните системи за моделиране най-често се работи с NURBS криви и повърхнини от степените 1-ва, 2-ра, 3-та, 5-та и 7-ма, като от тях най-универсалната е 3-та [16, 35, 97].

NURBS моделите, които се предлагат в Глава II и Глава III са криви от втора степен и биквадратни повърхнини. Те могат да бъдат повдигнати до по-горна степен без това да ги промени геометрично или параметрично [91, 189 с.]. В [91, 188-212 с.] се разглеждат няколко метода за повдигане на степента на NURBS криви и повърхнини. При необходимост, всеки един от тях би могъл да бъде приложен върху NURBS моделите, които се предлагат в настоящата разработка.

4.5. Сравнителен анализ и демонстрация на работата на CCM

В Глава I, раздел 1.5 бяха дадени съществуващи методи за построяване на квадратичните криви и повърхнини чрез NURBS криви и повърхнини. Прилагането на повечето от тези техники в 3ds Max е възможно. Ще бъде направен сравнителен анализ на работата на тези методи и работата на разработените нови графични инструменти. Това ще се изразява както в анализ на работата им, така и в анализ на резултата, който се получава. Основните характеристики, които се следят са:

- брой на стъпките при конструиране;
- време за конструиране;
- възможност за локална модификация;
- брой на контролните върхове;
- ниво на визуализация (само за повърхнините).

За всяка от разглежданите криви и повърхнини ще бъдат разгледани възможните подходи за построяване с наличните средства в 3ds Max. След което ще се

демонстрира построяването на NURBS кривата или повърхнината от плъгина, работещ със съответния NURBS модел.

Характеристиките на основните компоненти на компютърната система, на която са извършени сравнителният анализ и демонстрацията са поместени в таблица 4.5.

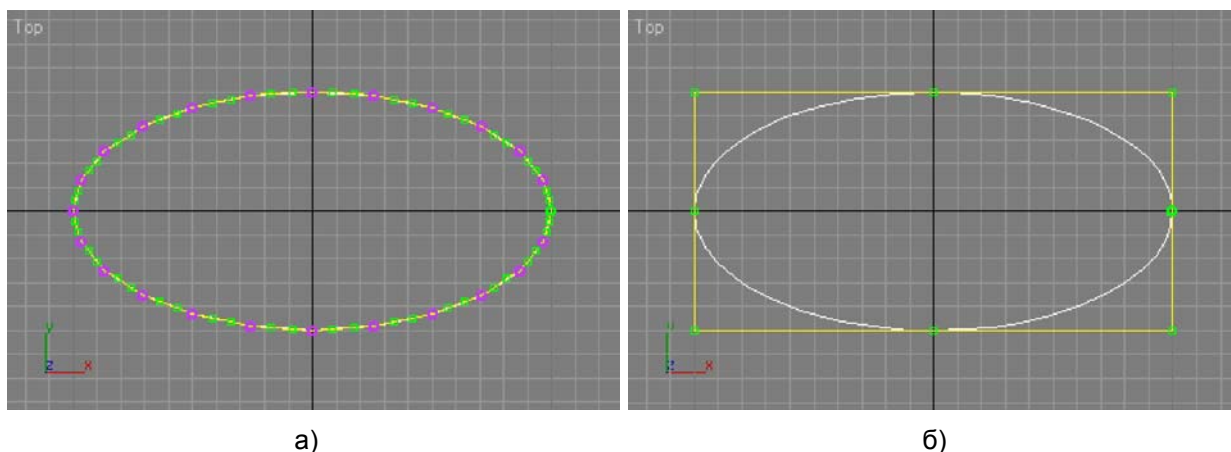
Таблица 4.5 Характеристики на основните компоненти на компютърната система за анализ и демонстрация.

Компонент	Търговска марка и параметри
Производител на системата	VIA Technologies, Inc.
Модел на системата	VT8363
Тип на системата	X86-based PC
Процесор	AMD Duron 1.29 GHz
Графична карта (ГК)	NVIDIA GeForce4 MX 440
RAM на ГК	64,00 MB
BIOS версия	ASI, Inc. 6.00 PG, 30.11.2001 г.
RAM	256,00 MB
Виртуална памет	2,00 GB
Операционна система (ОС)	Microsoft Windows XP Professional
Версия на ОС	5.1.2600 Service Pack 2 Build 2600

4.5.1. Построяване на NURBS дъга от елипса

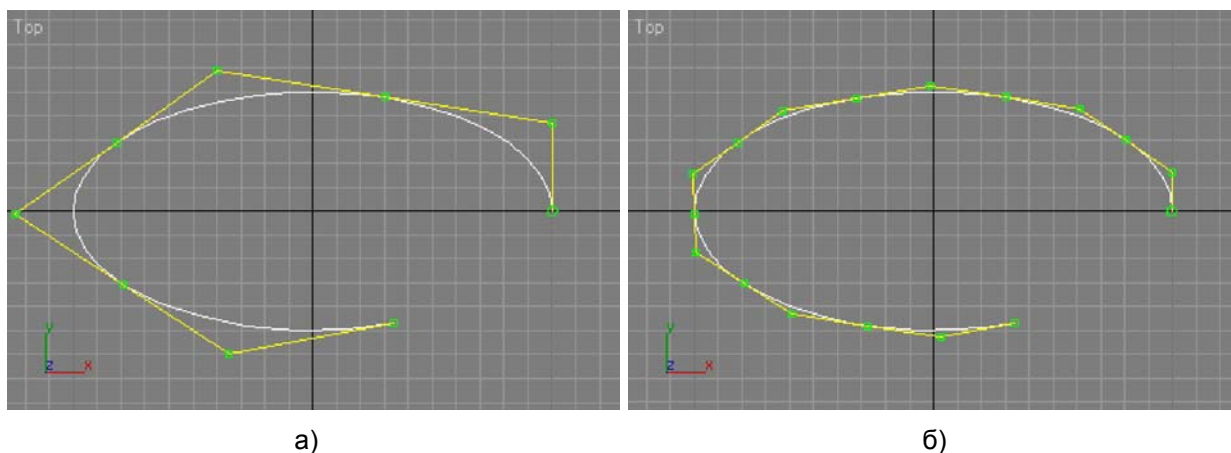
Елипса, представена в NURBS, може да бъде построена в 3ds Max чрез намиране на сечението между кръгов конус и равнина (за получаването на окръжност като сплайн). След това сечението да се дилатира по x (или по y) за да се получи желания размер на елипсата. След което тази сплайн крива да се преобразува до NURBS крива. По този подход не е възможно построяването на отворени дъги, а само на пълна елипса. Реалният резултат (фиг. 4.9а) се получава за 3 стъпки при конструирането и за време около 60 секунди. Тя е NURBS крива със 73 контролни върхове, което означава, че локална модификация е възможна, но моделирането с кривата ще е трудно.

Еквивалентна крива се построява от плъгина NURBS Elliptical Arc за 1 стъпка, за време 5 секунди (12 пъти по-бързо от посочения по-горе метод) и е представена с 9 контролни върха (около 8 пъти по-малко) (фиг. 4.9б). Броят на контролните върхове зависи от избрания брой на сегментите на кривата, т.е. при необходимост този брой може да бъде увеличен.



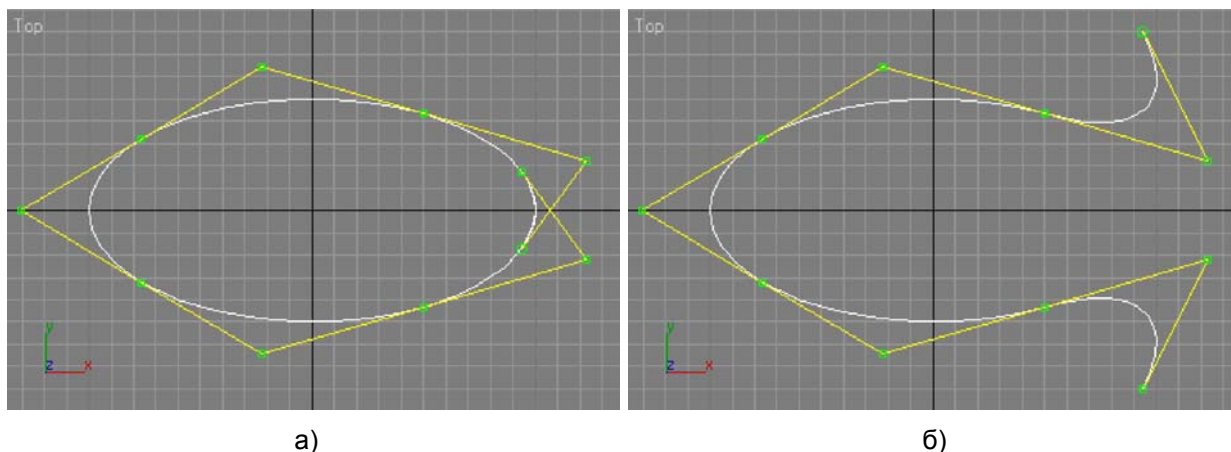
Фиг. 4.9 а) Елипса получена чрез сечение и преобразувана в NURBS;
 б) NURBS елипса построена с плъгина NURBS Elliptical Arc за стойностите: $a = 100$,
 $b = 50$, $\alpha_s = 0^\circ$, $\alpha_e = 360^\circ$, $segments = 4$.

Един сравнително ефективен подход в 3ds Max е да се използват примитивите Circle, Ellipse и Arc. Получените криви могат да бъдат преобразувани до NURBS криви. Недостатък на Circle и Ellipse е, че не е възможно построяването на отворени дъги, което пък е възможно с Arc (фиг. 4.10а). Недостатък при използването на Arc е, че не е възможно построяването на дъги при вариране на ъгъла за построяване за стойности по-големи от 360° .



Фиг. 4.10 а) Елипса построена чрез примитива на 3ds Max “Arc” за стойностите:
 $Radius = 50$, $\alpha_s = 0^\circ$, $\alpha_e = 290^\circ$ и преобразувана в NURBS;
 б) NURBS елипса построена с плъгина NURBS Elliptical Arc за стойностите: $a = 100$,
 $b = 50$, $\alpha_s = 0^\circ$, $\alpha_e = 290^\circ$, $segments = 6$.

На фигура 4.10б е показана еквивалентна крива, построена от плъгина NURBS Elliptical Arc, подразделена на 6 сегмента. Освен това плъгинът може да построява дъги при вариране на ъгъла за построяване за стойности по-големи от 360° (фиг. 4.11а), а също така да бъде локално модифицирана с лекота (фиг. 4.11б).

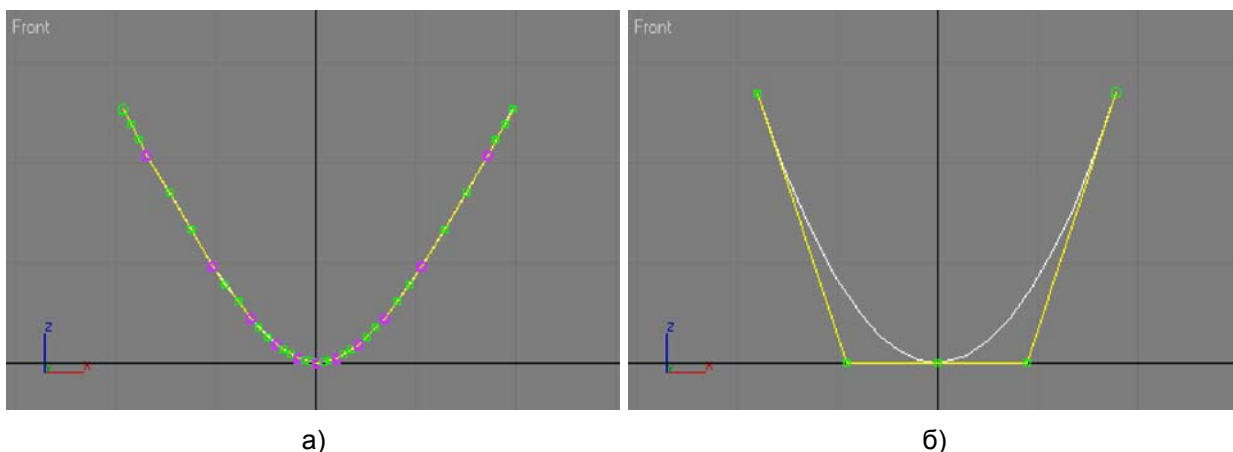


Фиг. 4.11 а) NURBS елипса построена с плъгина NURBS Elliptical Arc за стойностите:
 $a = 100$, $b = 50$, $\alpha_s = -20^\circ$, $\alpha_e = 380^\circ$, $segments = 5$;
 б) NURBS кривата е локално модифицирана.

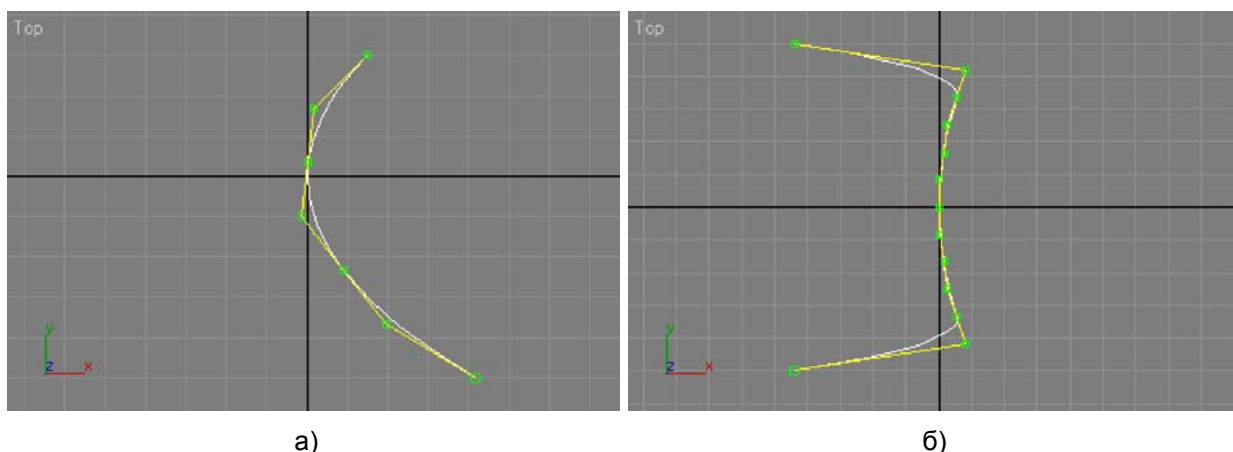
4.5.2. Построяване на NURBS дъга от парабола

За получаване на дъга от парабола трябва да се следва подхода за намиране на конично сечение с тази форма. Получената сплайн крива може да се преобразува до NURBS крива. Реалният резултат (фиг. 4.12а) се получава за 3 стъпки при конструирането и за време около 75 секунди. Тя е NURBS крива с 37 контролни върхове.

На фигура 4.12б е показана еквивалентна крива, построена от плъгина NURBS Parabolical Arc за 1 стъпка, за време 5 секунди (15 пъти по-бързо от посочения по-горе метод) и подразделена на 2 сегмента, т.е. 5 контролни върха (около 7 пъти по-малко). Освен това плъгинът може да построява дъги за различни стойности на u_s и u_e (фиг. 4.13а), а също така те могат да бъдат локално модифицирани с лекота (фиг. 4.13б).



Фиг. 4.12 а) Парабола получена чрез сечение и преобразувана в NURBS;
 б) NURBS парабола построена с пългана NURBS Parabolical Arc за стойностите: $p = 6$,
 $u_s = -18$, $u_e = 18$, $segments = 2$.



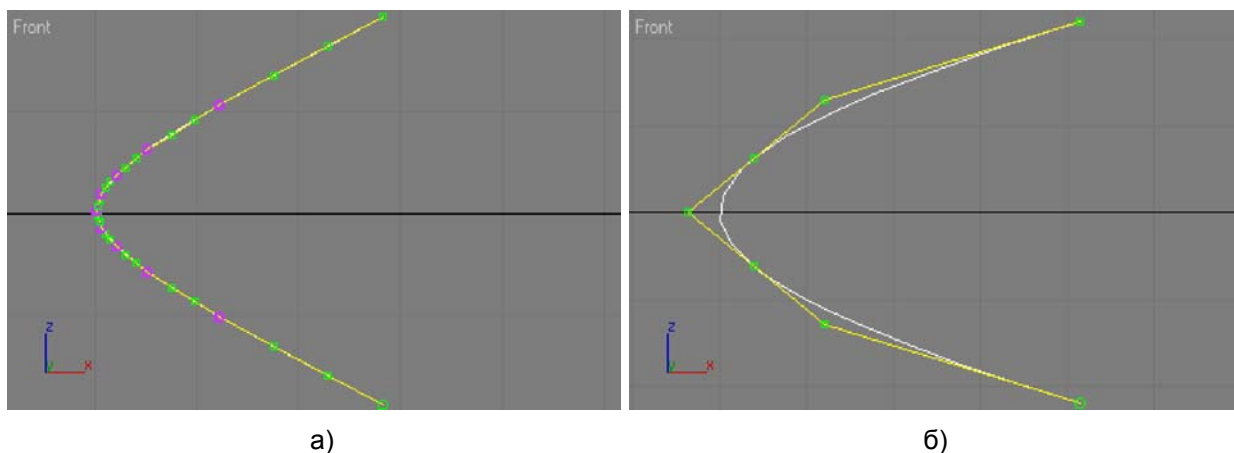
Фиг. 4.13 NURBS парабола построена с пългана NURBS Parabolical Arc за стойностите:
 а) $p = 30$, $u_s = -50$, $u_e = 30$, $segments = 3$; б) $p = 100$, $u_s = -50$, $u_e = 50$,
 $segments = 6$ и е локално модифицирана.

4.5.3. Построяване на NURBS дъга от хипербола

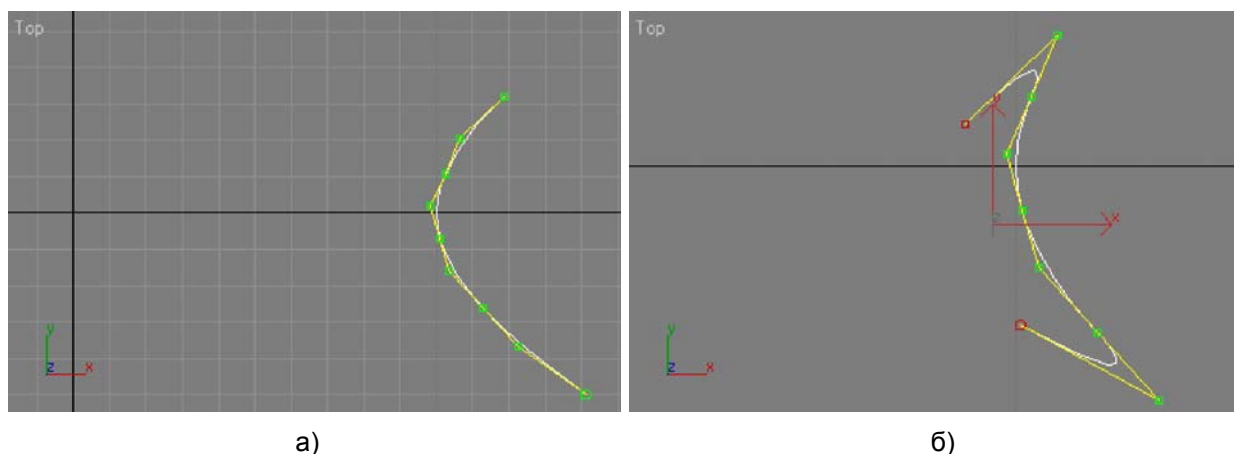
За получаване на дъга от хипербола трябва да се следва подхода за намиране на конично сечение с тази форма. Получената сплайн крива може да се преобразува до NURBS крива. Реалният резултат (фиг. 4.14а) се получава за 3 стъпки при конструирането и за време около 75 секунди. Тя е NURBS крива с 31 контролни върхове.

На фигура 4.14б е показана еквивалентна крива, построена от пългана NURBS Hyperbolic Arc за 1 стъпка, за време 5 секунди (15 пъти по-бързо от посочения по-горе метод) и подразделена на 3 сегмента, т.е. 7 контролни върха. Освен това пългинът

може да построява дъги за различни стойности на α_s и α_e (фиг. 4.15а), а също така да бъдат локално модифицирани с лекота (фиг. 4.15б).



Фиг. 4.14 а) Хипербола получена чрез сечение и преобразувана в NURBS;
 б) NURBS хипербола построена с пългана NURBS Hyperbolic Arc за стойностите:
 $a = 100$, $b = 22$, $\alpha_s = 315^\circ$, $\alpha_e = 45^\circ$, $segments = 3$.



Фиг. 4.15 а) NURBS хипербола построена с пългана NURBS Hyperbolic Arc за стойностите: $a = 100$, $b = 50$, $\alpha_s = 315^\circ$, $\alpha_e = 32.5^\circ$, $segments = 4$; б) NURBS хиперболата локално модифицирана.

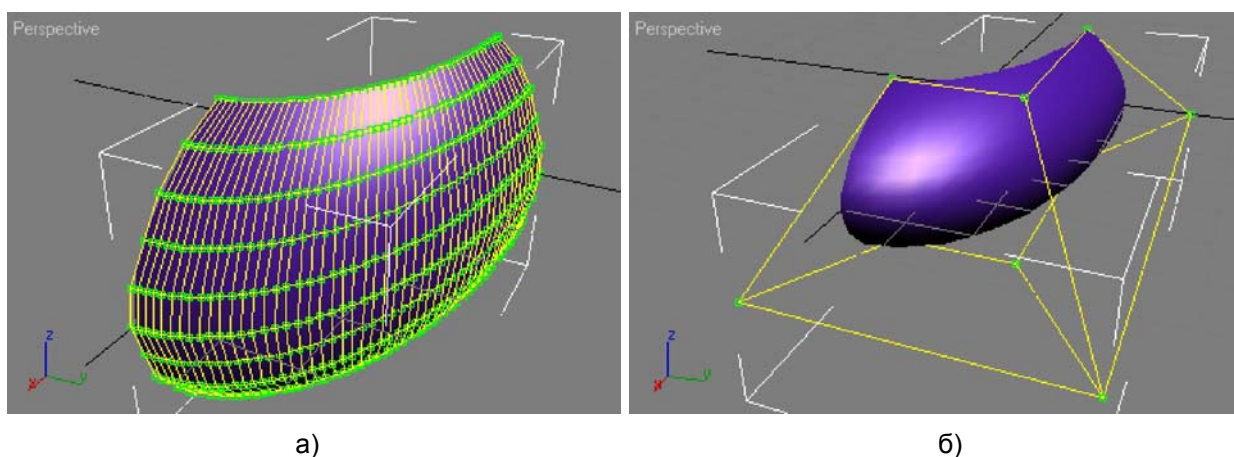
4.5.4. Построяване на NURBS елипсоид

В 3ds Max елипсоид преобразуван в NURBS може да се получи по няколко начина. Може би най-интуитивният е построяването на сфера, след това преобразуването ѝ в NURBS повърхнина и след това мащабирането ѝ до желания елипсоид. Основният недостатък тук е невъзможността да се построяват отворени елипсоид-повърхнини.

Друг подход е построяването на полуокръжност, която да се преобразува до NURBS крива, след което да се построи ротационна NURBS повърхнина. Но в този случай повърхнината е зависима⁸.

Най-ефективният и със сравнително добри резултати е подходът, при който се създава дъга от окръжност, преобразува се до сплайн, след което се изгражда ротационна повърхнина (зависима patch повърхнина). Тази повърхнина се мащабира до желанния елипсоид и накрая се преобразува до NURBS повърхнина. Реалният резултат (фиг. 4.16а) се получава за 5 стъпки при конструирането и за време около 45 секунди. Тя е NURBS повърхнина с 1280 контролни върхове с много добра визуализация в рендерираното изображение. Локална модификация е възможна, но поради огромния брой на контролните върхове моделирането с тази повърхнина е много трудно.

На фигура 4.16б е показана еквивалентна NURBS повърхнина създадена от плъгина NURBS Ellipsoid за 1 стъпка и за време 5 секунди (9 пъти по-бързо от най-ефективния посочен по-горе метод). Тя се определя от 9 контролни върха (около 142 пъти по-малко) като при нужда може да бъде подразделена, което ще доведе до увеличаване на броя на контролните върхове.



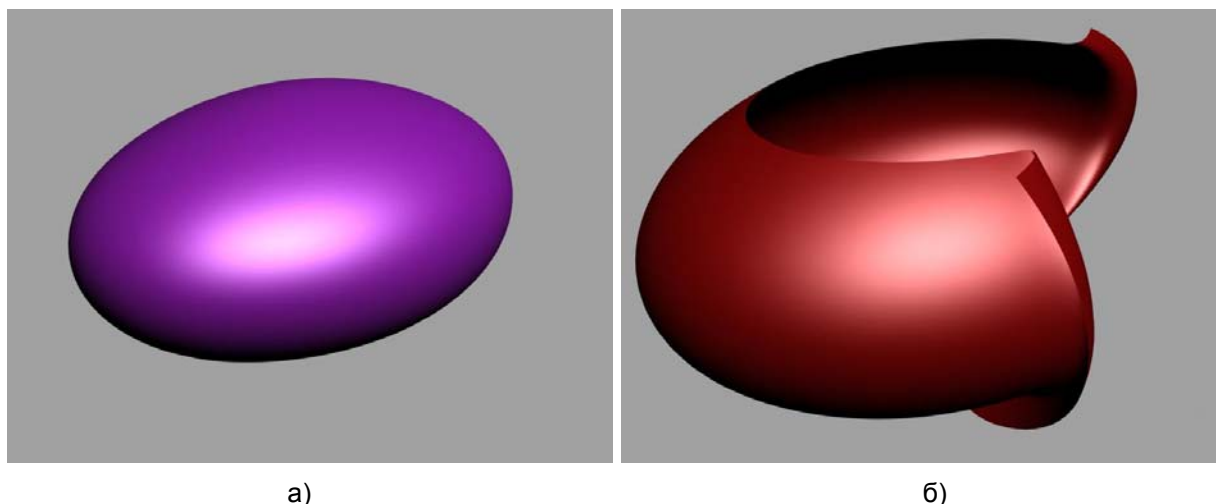
Фиг. 4.16 а) Елипсоид получен (чрез средствата на 3ds Max) посредством ротационно движение на дъга, мащабиране и след това преобразувана в NURBS;

б) NURBS елипсоид създаден с плъгина NURBS Ellipsoid за стойностите:

$$a = 60, b = 30, c = 20, \alpha_s = -45^\circ, \alpha_e = 45^\circ, \beta_s = 0^\circ, \beta_e = 90^\circ, \text{segments } (U \text{ и } V) = 1.$$

⁸ Повърхнина, която е построена с помощта на генератор (права или крива) и промяната на формата ѝ може да се извърши само чрез промяна на формата на генератора, т.е. повърхнината е “зависима” от генератора.

На фигура 4.17 са показани рендерирани изображения на два NURBS елипсоида създадени от плъгина NURBS Ellipsoid като единият е локално модифициран. От изображенията се вижда, че повърхнините са гладки и с отлична визуализация.



Фиг. 4.17 а) Рендерирано изображение на пълен NURBS елипсоид;

б) Рендерирано изображение на локално модифициран NURBS елипсоид създаден за стойностите: $a = 40$, $b = 30$, $c = 20$, $\alpha_s = -90^\circ$, $\alpha_e = 45^\circ$, $\beta_s = 0^\circ$, $\beta_e = 270^\circ$, $segments(U \text{ и } V) = 3$.

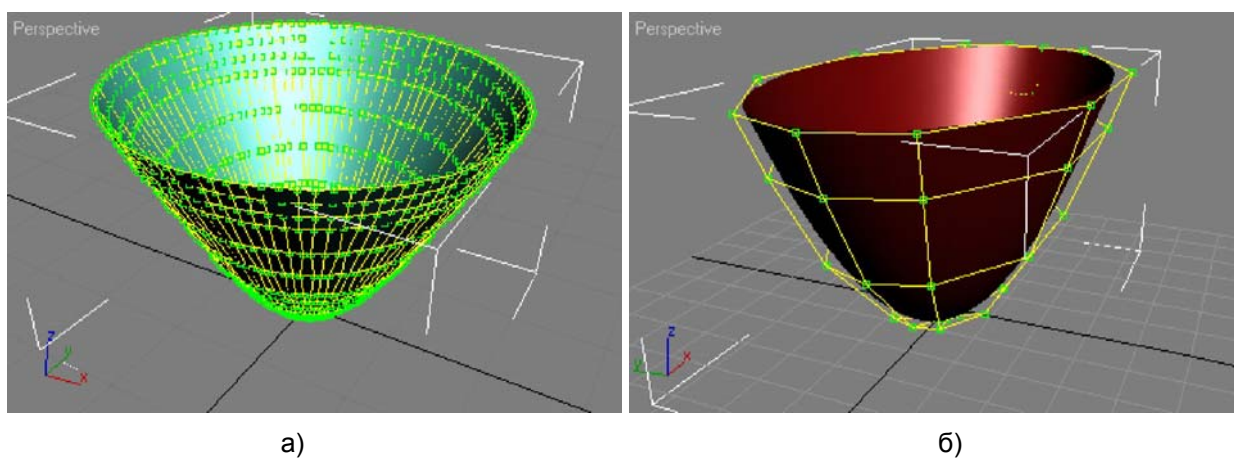
4.5.5. Построяване на NURBS параболоид

В 3ds Max построяването на параболоид в NURBS може да се извърши след като бъде получена сплайн дъга от парабола (по начина споменат в подраздел 4.4.2.) и тази крива се ползва като генератор. Ако изключим като възможност получаването на подчинена NURBS повърхнина, като подход остава сплайн кривата да се преобразува до NURBS крива, след това да се построи ротационна повърхнина, която да се преобразува до NURBS повърхнина и мащабира до желания елиптичен параболоид.

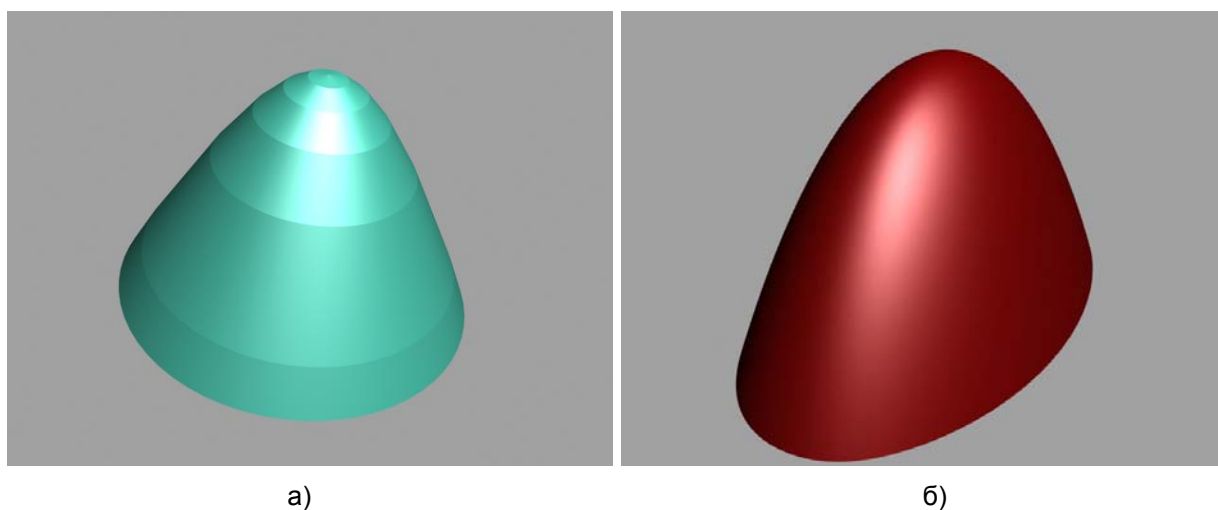
Резултатът (фиг. 4.18а) следвайки този подход се получава за 6 стъпки при конструирането и за време около 180 секунди. NURBS повърхнината е бикубична, с 2560 контролни върхове. На рендерираното изображение (фиг. 4.19а) се вижда, че тя не е достатъчно гладка (има видими ръбове). Поради огромния брой на контролните върхове моделирането с тази повърхнина е много трудно.

На фигура 4.18б е показана еквивалентна бикубична NURBS повърхнина, създадена от плъгина NURBS Paraboloid за 1 стъпка и за време 5 секунди (36 пъти по-бързо от метода посочен по-горе), подразделена на 4 сегмента и в двете направления. Ако повърхнината се създаде като бикубична без допълнително подразделяне, контролните върхове, които я определят, ще са 52 (около 49 пъти по-малко). На фигура

4.196 се вижда, че повърхнината е достатъчно гладка, следствие на което визуализацията ѝ в рендерираното изображение е отлична.

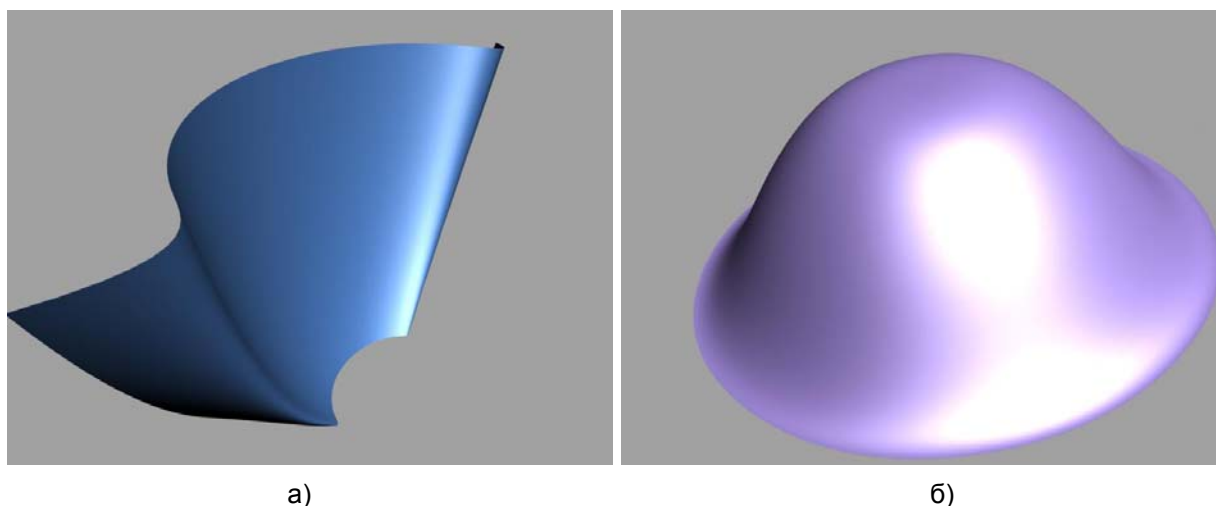


Фиг. 4.18 а) Параболоид получен (чрез средствата на 3ds Max) посредством сечение, ротационно построяване и преобразуване в NURBS; б) NURBS параболоид създаден с пългина NURBS Paraboloid за стойностите: $a = 5$, $b = 3$, $t_s = 0$, $t_e = 10$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $segments (U \text{ и } V) = 4$.



Фиг. 4.19 Рендерирани изображения на параболоидите от:
а) фигура 4.18а; б) фигура 4.18б.

На фигура 4.20 са показани рендерирани изображения на бикубични NURBS параболоиди, които са локално модифицирани.



Фиг. 4.20 NURBS параболоиди създадени с плъгина NURBS Paraboloid за стойностите:

а) $a = 5$, $b = 3$, $t_s = 3$, $t_e = 10$, $\beta_s = 180^\circ$, $\beta_e = 360^\circ$, $segments (U \text{ и } V) = 2$;

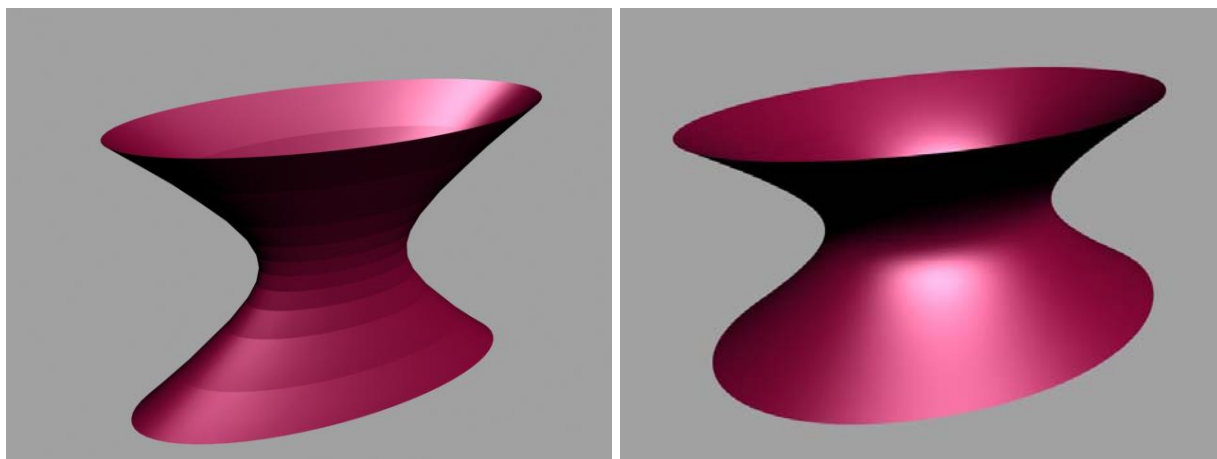
б) $a = 5$, $b = 5$, $t_s = 0$, $t_e = 10$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $segments (U \text{ и } V) = 4$.

4.5.6. Построяване на NURBS хиперболоид

В 3ds Max построяването на хиперболоид като NURBS повърхнина може да се извърши след като бъде получена сплайн дъга от хипербола (по начина споменат в подраздел 4.4.3.) и се действа аналогично на подхода за построяването на поараболоида от подраздел 4.4.5.

Резултатът се получава за 6 стъпки при конструирането и за време около 240 секунди. NURBS повърхнината е бикубична, с 4864 контролни върхове. На рендерираното изображение (фиг. 4.21а) се вижда, че тя не е достатъчно гладка. Поради огромния брой на контролните върхове моделирането с тази повърхнина е много трудно.

На фигура 4.21б е показана еквивалентна бикубична NURBS повърхнина създадена от плъгина NURBS Hyperboloid за 1 стъпка и за време 5 секунди (48 пъти по-бързо от метода посочен по-горе). Ако повърхнината се създаде като биквадратна без допълнително подразделяне, контролните върхове, които я определят, ще са 27 (около 180 пъти по-малко). На фигура 4.21б се вижда, че повърхнината е достатъчно гладка, т.е. визуализацията ѝ в рендерираното изображение е отлична.



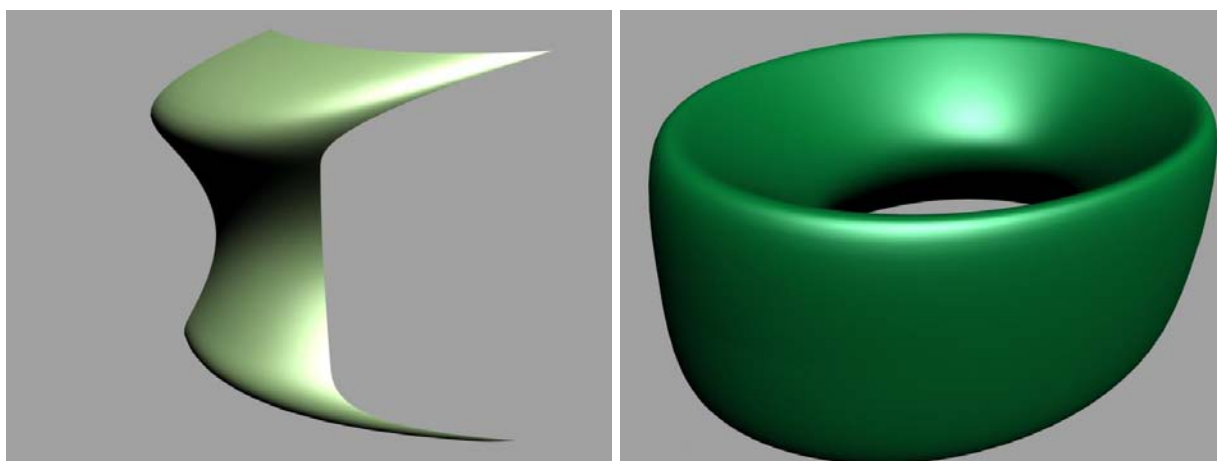
а)

б)

Фиг. 4.21 а) Хиперболоид получен (чрез средствата на 3ds Max) посредством сечение, ротационно построяване и преобразуване в NURBS; б) NURBS хиперболоид създаден с плъгина NURBS Hyperboloid за стойностите: $a = 40$, $b = 30$, $c = 20$, $\alpha_s = 300^\circ$,

$$\alpha_e = 60^\circ, \beta_s = 0^\circ, \beta_e = 360^\circ, \text{segments } (U \text{ u } V) = 4.$$

На фигура 4.22 са показани две рендерирани изображения на а) биквадратен и б) бикубичен NURBS хиперболоид, които са локално модифицирани.



а)

б)

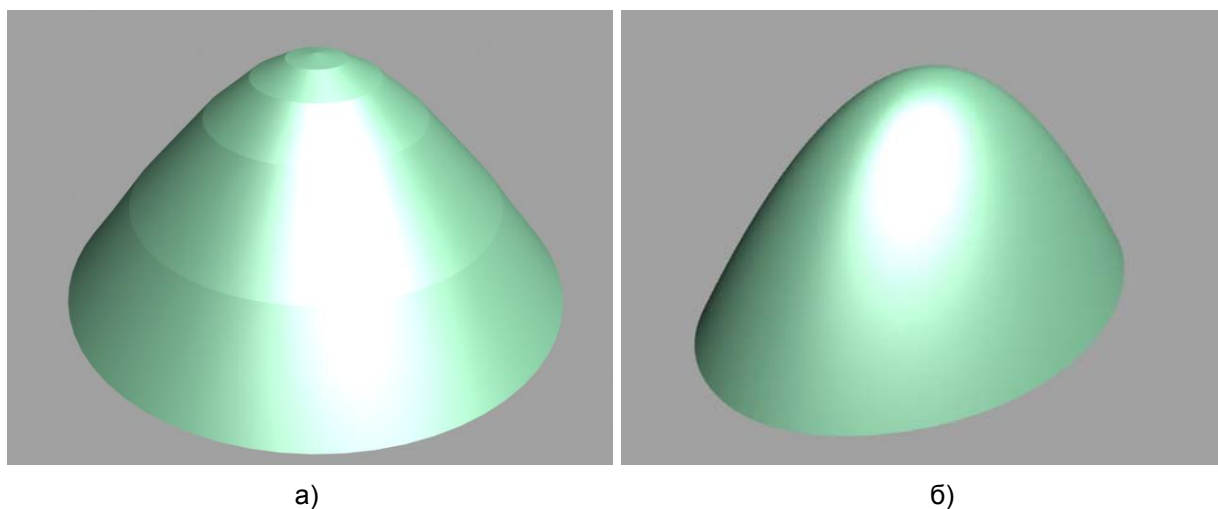
Фиг. 4.22 NURBS хиперболоиди създадени с плъгина NURBS Hyperboloid за стойностите: а) $a = 40$, $b = 40$, $c = 40$, $\alpha_s = 300^\circ$, $\alpha_e = 60^\circ$, $\beta_s = 0^\circ$, $\beta_e = 75^\circ$, $\text{segments } (U \text{ u } V) = 4$; б) $a = 40$, $b = 30$, $c = 20$, $\alpha_s = 295^\circ$, $\alpha_e = 65^\circ$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $\text{segments } (U \text{ u } V) = 8$.

4.5.7. Построяване на NURBS двоен хиперболоид

Построяването на двоен хиперболоид като NURBS повърхнина се извършва след като бъде получена сплайн дъга от хипербола (по начина споменат в подраздел 4.4.3.) и след това се действа аналогично на подхода за построяването на параболоида от подраздел 4.4.5. Този подход се прилага при построяването и на двете части на двойния хиперболоид.

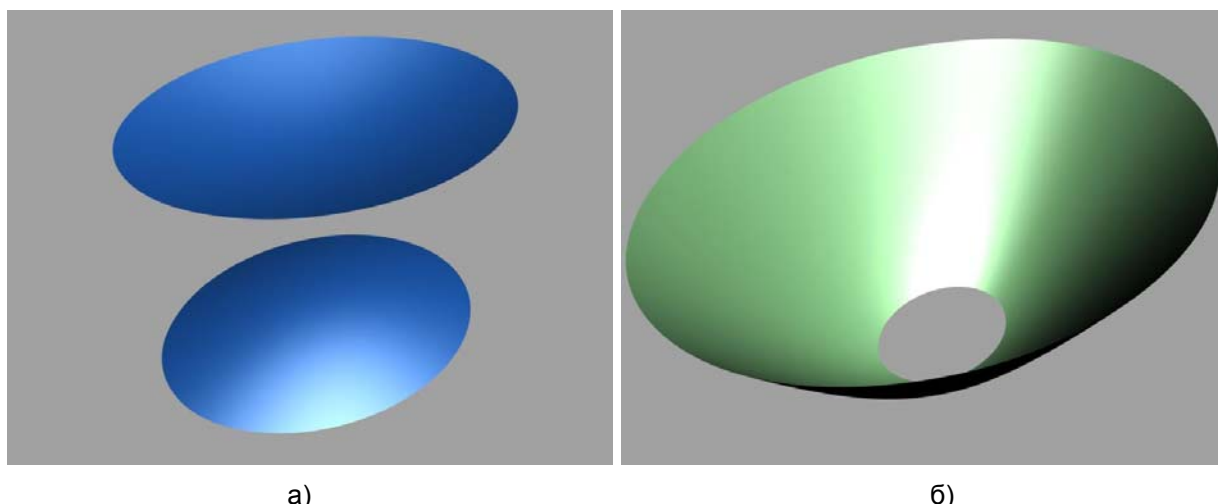
Резултатът се получава за 6 стъпки при конструирането на една част и за време около 180 секунди. NURBS повърхнината е бикубична, с 2560 контролни върхове. На рендерираното изображение (фиг. 4.23а) се вижда, че тя не е достатъчно гладка. Поради огромния брой на контролните върхове моделирането с тази повърхнина е много трудно.

На фигура 4.23б е показана еквивалентна бикубична NURBS повърхнина създадена от плъгина NURBS Double Hyperboloid за 1 стъпка, за време 5 секунди (36 пъти по-бързо от метода посочен по-горе) и подразделена на 4 сегмента и в двете направления. Ако повърхнината се създаде като биквадратна без допълнително подразделяне, контролните върхове, които я определят, ще са 27 (около 94 пъти по-малко). На фигура 4.23б се вижда, че визуализацията на повърхнината в рендерираното изображение е отлична.



Фиг. 4.23 а) Двоен хиперболоид получен (чрез средствата на 3ds Max) посредством сечение, ротационно построяване и преобразуване в NURBS; б) Едно крило на NURBS двоен хиперболоид създадено с плъгина NURBS Double Hyperboloid за стойностите: $a = 40$, $b = 30$, $c = 70$, $\alpha_s = 0^\circ$, $\alpha_e = 45^\circ$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $segments (U \text{ и } V) = 4$.

На фигура 4.24 са показани две рендерирани изображения на а) биквадратен и б) бикубичен NURBS двоен хиперболоид.



а) б)

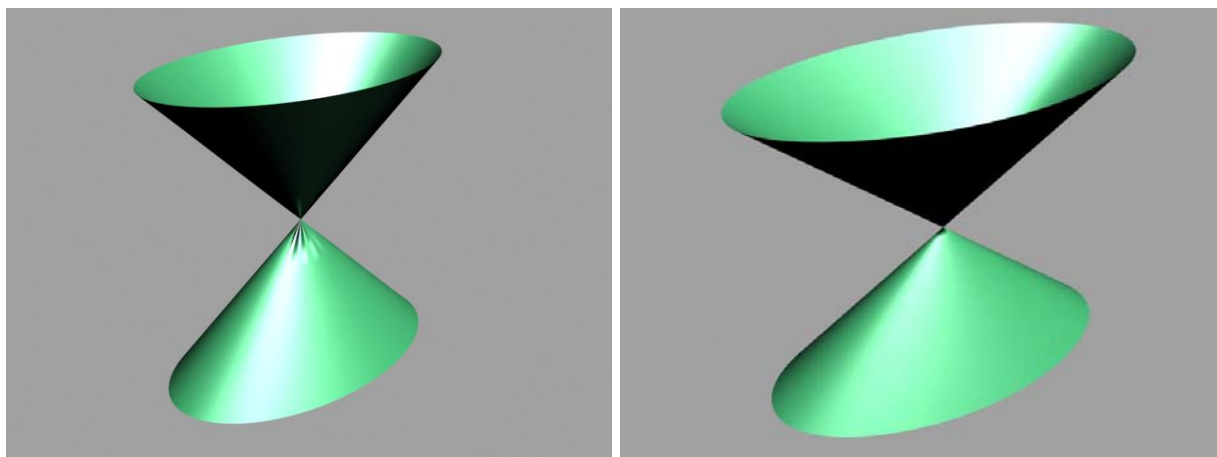
Фиг. 4.24 NURBS двойни хиперboloиди създадени с плъгина NURBS Hyperboloid за стойностите: а) $a = 40$, $b = 30$, $c = 20$, ляво крило: $\alpha_s = 0^\circ$, $\alpha_e = 45^\circ$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$ и дясно крило: $\alpha_s = 180^\circ$, $\alpha_e = 225^\circ$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $segments (U \cup V) = 4$; б) $a = 40$, $b = 30$, $c = 68$, $\alpha_s = 20^\circ$, $\alpha_e = 53^\circ$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $segments (U \cup V) = 4$.

4.5.8. Построяване на NURBS конус

В 3ds Max построяването на конус като NURBS повърхнина може да се извърши след като бъде направена права (сплайн) линия и се действа аналогично на подхода за построяването на параболоида от подраздел 4.4.5.

Резултатът се получава за 5 стъпки при конструирането и за време около 45 секунди NURBS повърхнината е бикубична, с 256 контролни върхове. На рендерираното изображение (фиг. 4.25а) се вижда, че тя е сравнително гладка, но не достатъчно, особено във върха си. Поради сравнително големия брой на контролните върхове моделирането с тази повърхнина е трудно.

На фигура 4.25б е показана еквивалентна бикубична NURBS повърхнина създадена от плъгина NURBS Cone за 1 стъпка и за време 5 секунди (9 пъти по-бързо от метода посочен по-горе). Ако повърхнината се създаде като биквадратна без допълнително подразделяне, контролните върхове, които я определят, ще са 27 (около 9 пъти по-малко). На фигура 4.25б се вижда, че визуализацията на повърхнината в рендерираното изображение е отлична.

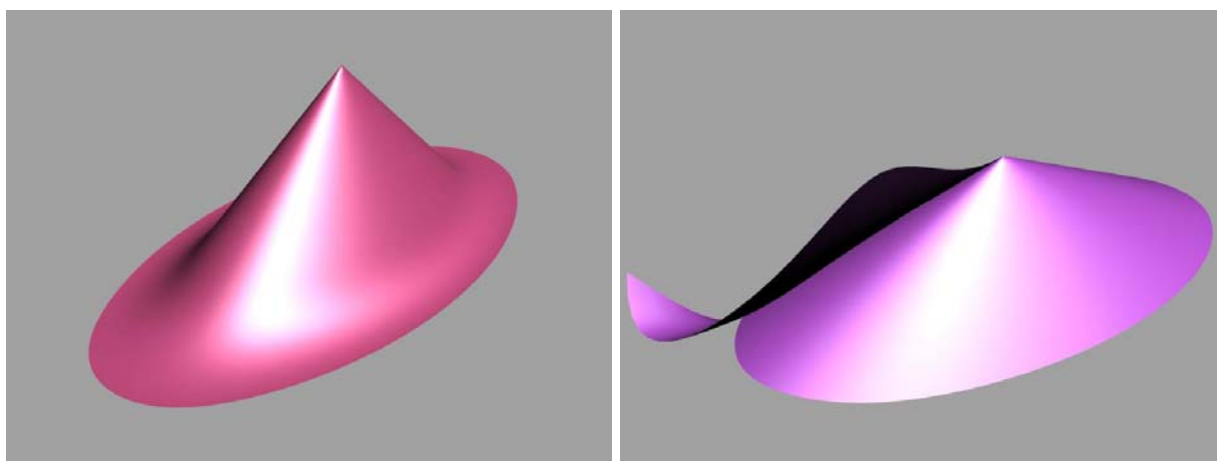


а)

б)

Фиг. 4.25 а) Конус получен (чрез средствата на 3ds Max) посредством ротационно построяване и преобразуване в NURBS; б) NURBS конус създаден с плъгина NURBS Cone за стойностите: $a = 1.2$, $b = 0.6$, $c = 1$, $s_s = -30$, $S_e = 30$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $U \text{ segments} = 4$, $V \text{ segments} = 1$.

На фигура 4.26 са показани две рендерирани изображения на бикубични NURBS конуси, които са локално модифицирани.



а)

б)

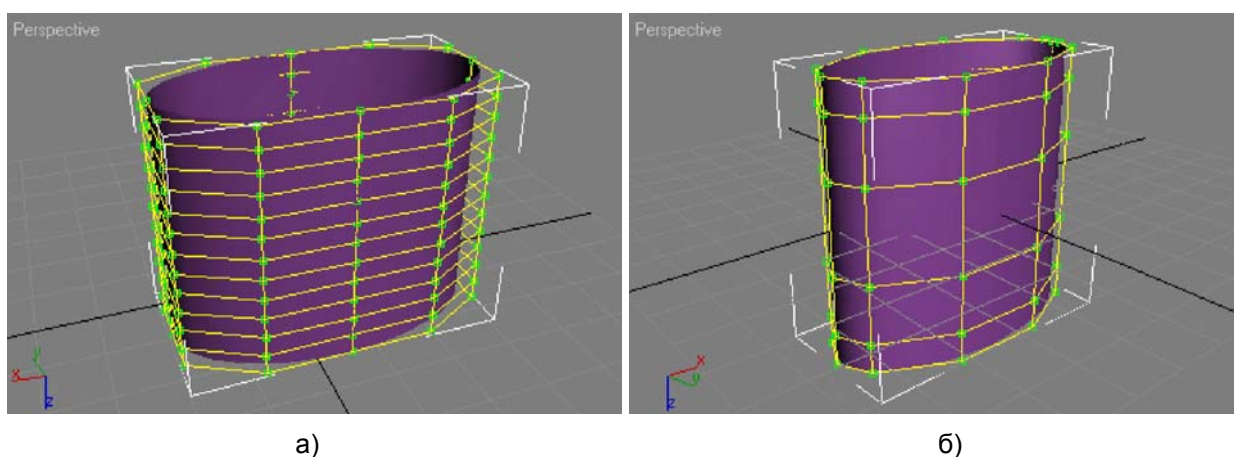
Фиг. 4.26 NURBS конуси създадени с плъгина NURBS Cone за стойностите: а) $a = 2$, $b = 1$, $c = 1$, $s_s = 0$, $s_e = 30$, $\beta_s = 0^\circ$, $\beta_e = 360^\circ$, $U \text{ segments} = 4$, $V \text{ segments} = 1$; б) $a = 2$, $b = 1$, $c = 1$, $s_s = 0$, $s_e = 30$, $\beta_s = 0^\circ$, $\beta_e = 385^\circ$, $U \text{ segments} = 5$, $V \text{ segments} = 1$.

4.5.9. Построяване на NURBS елиптичен цилиндър

В 3ds Max могат да бъдат приложени не малък брой подходи за построяване на NURBS елиптичен цилиндър. Тази повърхнина може да бъде изградена чрез

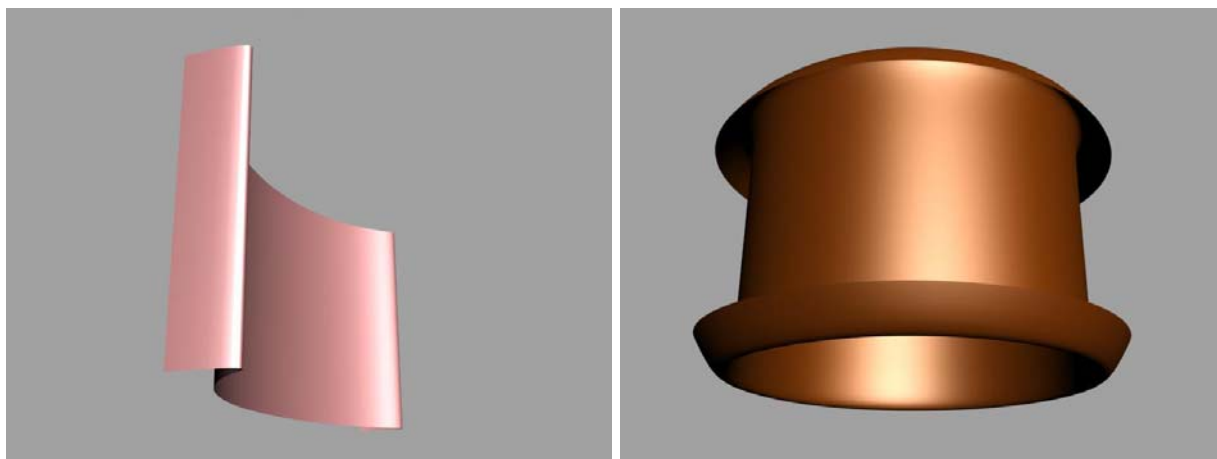
ротационно движение на права около ос, след което получената повърхнина да се мащабира до желаня елиптичен цилиндър. След това повърхнината да бъде преобразувана до NURBS. Елиптичен цилиндър може да бъде изграден и чрез праволинейно движение на елипса и след това получената повърхнина да се преобразува до NURBS. Има и други подходи, които няма да бъдат споменати тук. Който и подход, обаче, да се избере, винаги участва преобразуването до NURBS повърхнина. А както се вижда при до сега разгледаните повърхнини, това не дава много добри резултати.

На фигура 4.27а е даден един преобразуван до NURBS елиптичен цилиндър. Тази повърхнина е с 256 контролни върхове. На фигура 4.27б е показана еквивалентна бикубична NURBS повърхнина създадена от плъгина NURBS Elliptic Cylinder.



Фиг. 4.27 а) Елиптичен цилиндър получен (чрез средствата на 3ds Max) посредством замитане на елипса и преобразуван до NURBS; б) NURBS елиптичен цилиндър създаден с плъгина NURBS Elliptic Cylinder за стойностите: $a = 30$, $b = 15$, $\alpha_s = 0^\circ$, $\alpha_e = 360^\circ$, $t_s = -30$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 4$.

На фигура 4.28 са показани две рендерирани изображения на бикубични NURBS елиптични цилиндри, които са локално модифицирани. Вижда се, че визуализацията им в рендерираното изображение е отлична.



а)

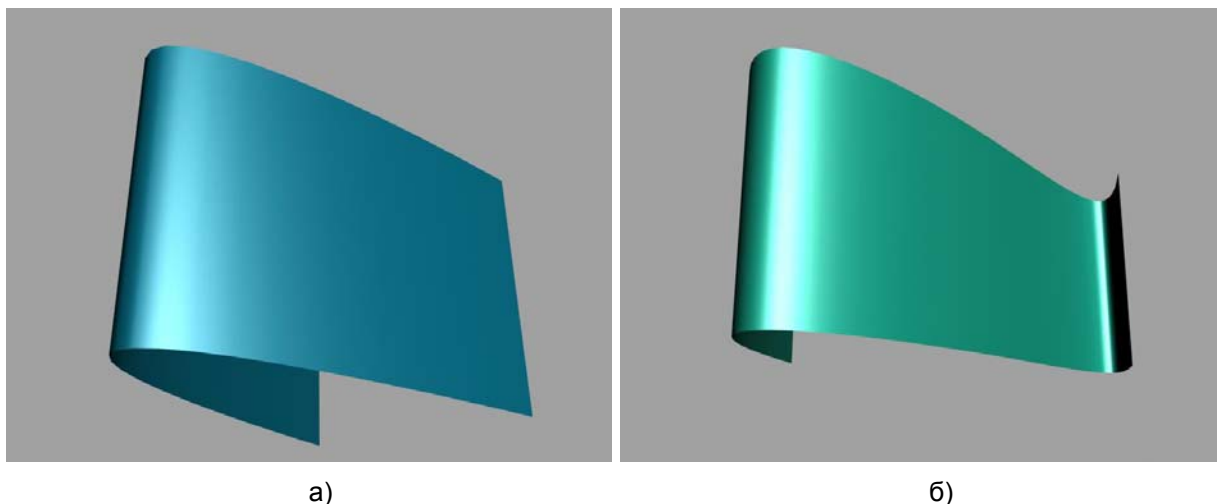
б)

Фиг. 4.28 NURBS елиптически цилиндри създадени с плъгина NURBS Elliptic Cylinder за стойностите: а) $a = 28$, $b = 17$, $\alpha_s = 0^\circ$, $\alpha_e = 178^\circ$, $t_s = 0$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 2$; б) $a = 25$, $b = 25$, $\alpha_s = 0^\circ$, $\alpha_e = 360^\circ$, $t_s = -30$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 4$.

4.5.10. Построяване на NURBS параболичен цилиндър

Построяването на параболичен цилиндър като NURBS повърхнина се извършва след като бъде получена сплайн дъга от парабола (по начина споменат в подраздел 4.4.2.) и след това се действа аналогично на подходите за построяването на елиптически цилиндър (посочени в подраздел 4.4.9). Резултатите се получават подобни на тези при елиптическия цилиндър.

На фигура 4.29 са показани две рендерирани изображения на един биквадратен и един бикубичен NURBS параболичен цилиндър. Вижда се, че визуализацията им в рендерираното изображение е отлична.



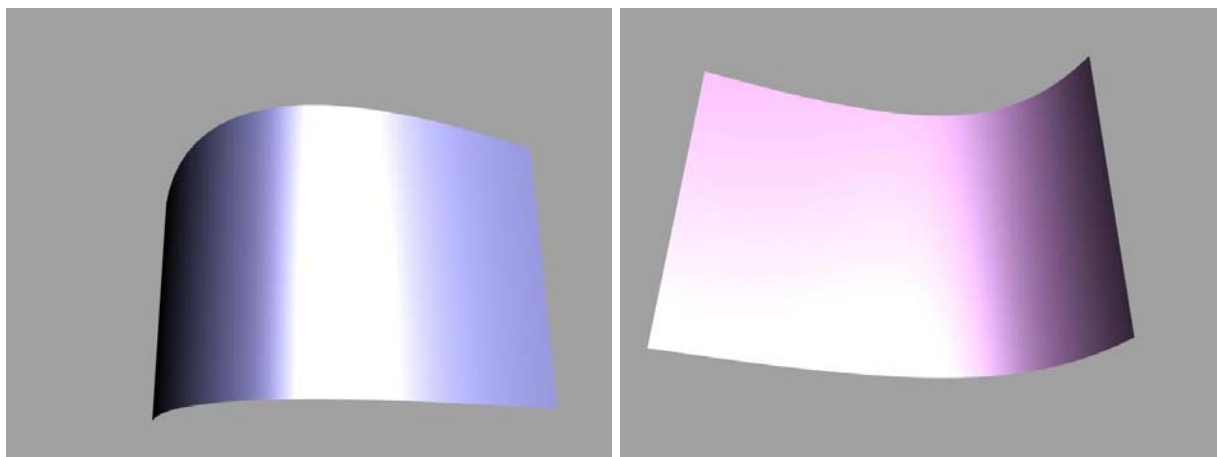
Фиг. 4.29 NURBS параболични цилиндри създадени с плъгина NURBS Parabolic Cylinder за стойностите:

- а) $p = 5$, $s_s = -30$, $s_e = 30$, $t_s = -30$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 1$;
 б) $p = 3.5$, $s_s = -10$, $s_e = 30$, $t_s = 0$, $t_e = 30$, $U \text{ segments} = 4$, $V \text{ segments} = 4$.

4.5.11. Построяване на NURBS хиперболичен цилиндър

Построяването на хиперболичен цилиндър като NURBS повърхнина се извършва след като бъде получена сплайн дъга от хипербола (по начина споменат в подраздел 4.4.3.) и след това се действа аналогично на подходите за построяването на елиптичен цилиндър (посочени в подраздел 4.4.9). Резултатите се получават подобни на тези при елиптичния цилиндър.

На фигура 4.30 са показани две рендерирани изображения на един биквадратен и един бикубичен NURBS хиперболичен цилиндър. Визуализацията им в рендерираното изображение е отлична.



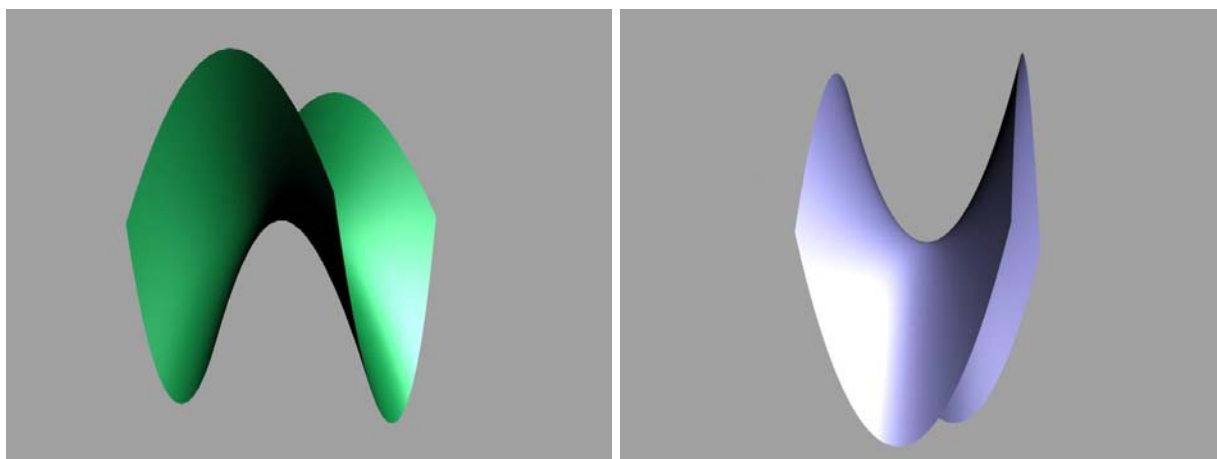
а)

б)

Фиг. 4.30 NURBS хиперболични цилиндри построени с плъгина NURBS Hyperbolic Cylinder за стойностите: а) $a = 100$, $b = 50$, $\alpha_s = 315^\circ$, $\alpha_e = 45^\circ$, $t_s = -30$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 1$; б) $a = 96$, $b = 51$, $\alpha_s = 135^\circ$, $\alpha_e = 200^\circ$, $t_s = -30$, $t_e = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 1$.

4.5.12. Построяване на NURBS хиперболичен параболоид

За построяването на хиперболичен параболоид в 3ds Max не беше открит инструмент или доказан подход.



а)

б)

Фиг. 4.31 NURBS хиперболични параболоиди построени с плъгина NURBS Hyperbolic Paraboloid за стойностите: а) $a = 3$, $b = 2$, $Width = 50$, $Length = 30$, $U \text{ segments} = 1$, $V \text{ segments} = 1$; б) $a = 2$, $b = 2$, $Width = 30$, $Length = 30$, $U \text{ segments} = 4$, $V \text{ segments} = 4$.

На фигура 4.31 са показани две рендерирани изображения на един биквадратен и един бикубичен NURBS хиперболичен параболоид. Визуализацията им в рендерираното изображение е отлична.

4.6. Изводи

От направеното прилагане на NURBS моделите (предложени в Глава II и Глава III) на кривите и повърхнините от втора степен в 3D графичната система за моделиране 3ds Max, могат да бъдат направени следните изводи:

1. Изследвани са възможностите, с които 3D графичните системи за моделиране разполагат за функционално разширяване. Следствие от това е избрана една от тях (Autodesk 3ds Max) за прилагането на NURBS моделите на квадратичните криви и повърхнини. Тя отговаря на всички изисквания поставени към системата в началото на раздел 4.3. Следователно, това е подходяща 3D графична система за прилагане на предложените в Глава II и Глава III NURBS модели на криви и повърхнини.

2. Разработени са 12 специализирани софтуерни модула за 3ds Max. Основните етапи в този процес са: разработването на графичния потребителски интерфейс на CCM, написване на функциите създаващи новите NURBS модели и прилагане на техники за интерактивно създаване. Това позволява реалното тестване на NURBS моделите в системата и запознаване с резултатите, които се получават.

3. Предвидени са: възможността за създаване в различните изгледи на системата, нуждата от ограничаване на допустимите стойности в параметричното задаване на кривите и повърхнините и възможността за повдигане на степента на кривите и повърхнините. Това прави CCM да работят в 3D графичната система по същия начин както и вградените примитиви. След инсталирането им в графичното приложение те се интегрират и работят по същия начин, по който и вградените графични инструменти. А поради това, че са написани на MaxScript, те стават съвместими с всички версии на програмата от четвърта и по-нови.

4. Направен е сравнителен анализ на работата на CCM в системата и на резултатите, които се получават, по предварително зададени критерии, подобни на посочените в началото на раздел 4.5. Анализът между досега познати техники и предложените в този труд модели и подходи, показва много добри резултати в полза на новите модели в създаването, визуализацията и моделирането на обектите.

Представените в тази дисертация модели, алгоритми и специализирани софтуерни модули позволяват кривите и повърхнините да се създават:

- с брой на стъпките – 1, а при другите подходи – 3 - за кривите, и 5 и 6 - за повърхнините;

- в реално време – за около 5 секунди, което е от 9 до 48 пъти по-бързо от другите подходи;
- с отлична възможност за локална модификация;
- с минимален брой на контролните върхове: 3 - за кривите и 9 - за повърхнините, а при другите подходи от 31 до 73 - за кривите и от 256 до 4864 - за повърхнините;
- с отлична визуализация при рендериране (за повърхнините).

ЗАКЛЮЧЕНИЕ

В настоящата дисертация направихме представяне на кривите и повърхнините от втора степен чрез NURBS апарата за целите на ГМ.

Теоретичните разработки бяха разделени на две части. В първата част е включено представянето на кривите (Глава II), а във втората част – представянето на повърхнините (Глава III). В тези две части е разработен NURBS модела на всяка от квадратичните криви и повърхнини. Моделите са разработени така, че NURBS кривата или повърхнината да се построява след като квадратичната крива или повърхнина е зададена със стойности за параметрите от скаларно-параметричните ѝ уравнения. Предложените NURBS модели на криви и повърхнини предоставят отлични възможности за локална модификация на формата.

От изключително значение бе да приложим (Глава IV) получените NURBS модели в една 3D графична система за моделиране (например, Autodesk 3ds Max) като специализирани софтуерни модули, които представляват нови графични примитиви в системата, разширявайки нейния графичен инструментариум. Това позволи да се направи сравнителен анализ между съществуващи подходи (за построяване на разглежданите криви и повърхнини) в системата и предложените тук инструменти.

Поставените в първа глава задачи са решени, в резултат на което са постигнати **следните резултати:**

1. Изследвана е методологията за създаване на специализирани софтуерни модули-добавки в триизмерните графични системи и е определен единен подход за алгоритмична реализация на плъгини за моделиране.

2. Разработени са NURBS модели на криви от втора степен, с чиято помощ се построяват дъги от елипса, парабола и хипербола в положителна и отрицателна посока.

- Дъгите от елипса се построяват за стойности по-малки, равни и по-големи от $|360^\circ|$, което разширява приложните области на моделите.
- Представянето на дъги от параболи и хиперболи следва директно от доказани твърдения. Стойностите на координатите на контролните върхове и техните тегла на NURBS кривата, в зависимост от стойностите на параметрите от скаларно-параметричното задаване на дъгата, се намират по изведени формули.

Разработен е алгоритъм за подразделяне на NURBS кривите. NURBS моделите на тези криви могат да бъдат използвани самостоятелно, но те също участват и в представянето на квадратичните повърхнини чрез NURBS. Реализацията позволява построяване на кривите с произволен брой сегменти. Освен това, представената по този начин NURBS крива, позволява модифицирането ѝ в отделни сегменти да става с изключителна лекота. Представянето на трите типа криви в един и същ модел прави моделирането с тях еднотипно и интуитивно. Това повишава лекотата при работа с тях, следствие от което се повишава и производителността.

3. Разработени са NURBS модели на повърхнините елипсоид, елиптичен параболоид, прост хиперболоид, двоен хиперболоид и конус. За тази цел бе разработен и внедрен алгоритъм за построяване на ротационна биквадратна NURBS повърхнина ползвайки генераторна NURBS крива от втора степен. Повърхнините могат да бъдат подразделени и по двете си направления, което дава възможност за локална модификация.

4. Разработени са NURBS модели на повърхнините на правите цилиндри. За построяването на повърхнините се използва генераторна NURBS крива, която се замиа по зададен вектор (направление). Получава се биквадратна NURBS повърхнина, която може да бъде подразделена.

5. Разработен е NURBS модел на хиперболичен параболоид, довел до получаване на едно-сегментна, биквадратна NURBS-повърхнината. Разработен е алгоритъм за подразделяне на биквадратна NURBS повърхнина на зададен брой сегменти. Прилагайки този алгоритъм за NURBS модела на хиперболичния параболоид се получава подразделена повърхнина.

6. На база на създадените NURBS модели на криви и повърхнини са разработени дванадесет специализирани софтуерни модули за 3D графичната система за моделиране Autodesk 3ds Max. Те представляват софтуерни инструменти за построяване на нови графични примитиви в системата. Направен е сравнителен анализ (с познати в системата техники), който показва много добри резултати в полза на новите инструменти и примитиви в създаването, визуализацията и моделирането на обектите.

Дванадесетте специализирани софтуерни модула, които бяха разработени за 3D графичната система за моделиране 3ds Max, са приложени към дисертацията на DVD носител заедно с инсталационна програма на графичното приложение с 30 дневен лиценз за свободно ползване с нетърговски цели. Дванадесетте плъгина са одобрени и приети от Интернет портала за високи 3D технологии и приложения *HighEnd3D* и са

достъпни за изтегляне както следва: *NURBS Conical Arcs 1.4* от <http://www.highend3d.com/f/4368.html>, *NURBS Quadratic Surfaces 1.2* от <http://www.highend3d.com/f/4369.html>.

На базата на предложените NURBS модели могат да бъдат извършени допълнителни изследвания и разработки. Ето някои актуални бъдещи задачи:

- Да бъдат разработени подмодули, които в процеса на моделиране с NURBS обекти служат за изчисляване и моделиране на скаларните им инварианти и асоциираните с тях геометрични обекти (кривина, торзия, гаусова и средна кривина на повърхнините, дължина или лице, да бъдат построени еволюти и мрежите от основни линии върху повърхнината).
- Да се създадат графични примитиви, осъществяващи затваряне на кривите и повърхнините по определени критерии.
- На базата на досегашните разработки във всяка конкретна 3D система да се разработят и добавят алгоритми за избор на оптимални модели.
- Усъвършенстване на конкретните методики в 3D системите за да се развият адаптивни качества на съществуващите и новоразработваните модели.

Тези и още много други подобни задачи могат да бъдат естествено продължение на това изследване и приложение на получените резултати.

ЛИТЕРАТУРА

1. Аммерал, Л. *Машинная графика на персональных компьютерах*. Сол Систем, Москва, 1992.
2. Бабишков, Д. Л. *Моделиране на криви и повърхнини чрез рационални B-сплайни*. Докторска дисертация, ТУ, София, 1991.
3. Бордмън, Т. *3ds Max 6 – практически учебен курс*. Софтпрес, София, 2004.
4. Боянов, Б. *Лекции по числени методи*. Дарба, София, 1995.
5. Бърнев, П. *Компютърна графика*. Просвета, София, 1994.
6. Георгиев, И. К., Г. Герасимов и др. *Графично програмиране*. Графика с компютър, София, 1993.
7. Георгиев, И., И. Сиклунов. *Графично моделиране в проектирането и производството*. Техника, София, 1985.
8. Георгиев, И., Л. Бончев. *Автоматизирано проектиране, производство и тестване в цифровата електроника, I част*. Графика с компютър, София, 1995.
9. Гочев, Г. *Компютърно зрение и невронни мрежи*. Технически университет, София, 1998.
10. Грийнбърг, А. Д., С. Грийнбърг. *Adobe Photoshop 5 – пълно ръководство*. Софтпрес, София, 1999.
11. Де Бор, К. *Практическо ръководство по сплайнам*. Радио и связь, Москва, 1985.
12. Ендерле, Г. *Програмни средства машинной графики: Междунар. Стандарт GKS*. Радио и связь, Москва, 1988.
13. Каратопраклиева, И. *Дефурирциална геометрия*. СУ "Св. Климент Охридски", София, 1994.
14. Круглински, Д. Дж. *Visual C++ - поглед отвътре*. Софтпрес, София, 1998.
15. ЛаМот, Ратклиф и др. *Компютърни игри – програмиране и дизайн*. NiSoft, София, 1996.
16. Ламърс, Д., Л. Гудинг. *Maya 4 – учебен курс*. Софтпрес, 2002.
17. Лукипудис, Е. *Компютърна графика и геометрично моделиране*. Пазарджик, 1996.
18. Любенова, Е., П. Недевски и др. *Ръководство по математически анализ – част I*. СУ, София, 1994.
19. Мортиър, Ш. *3ds Max 5*. АлексСофт, София, 2003.
20. Мърдок, К. Л. *3ds max 4 – Библия*. АлексСофт, С., 2001.
21. Нисофт. *Съвремена компютърна графика – част I*. София, 1996.

22. Павлидис. *Алгоритмы машинной графики и обработки изображений*. Радио и связь, Москва, 1986.
23. Паул, М. Дж. Д. *Теория и методи за апроксимация*. Студии на БИАП – Математически науки, Том 5, София, 2002.
24. Райан, Д. *Инженерная графика в САПР*. Мир, Москва, 1987.
25. Рейнбоу, В. *Компютърна графика*. Питер, Москва, 2003.
26. Сендов, Бл., В. Попов. *Числени методи, Част 1*. СУ “Св. Климент Охридски”, София, 1996.
27. Станилов, Г. *Аналитична геометрия*. Наука и изкуство, София, 1974.
28. Стоянова, Р. *Компютърна графика*. Електронна книга: <http://www.geocities.com/graphicspub/>, ТУ София, 2004.
29. Удс, К., А. Бикало, Кр. Мъри. *3ds max 4 – Mastering*. СофтПрес ООД, София, 2001.
30. Харингтън, Ст. *Компютърна графика*. Техника, София, 1989.
31. Холзнер, Ст. *Microsoft Visual C++*. Alexsoft, София, 1995.
32. Хорстман, К. *Принципи на програмирането със C++*. ИК “Софтек”, София, 2000.
33. Христов, М. *Аналитична геометрия*. Астарта, Велико Търново, 2003.
34. Христов, М., Г. Накова. *Ръководство по геометрия – елементи на математическите основи на геометричното моделиране и компютърната графика*. Астарта, Велико Търново, 2005.
35. *3ds Max 7 Fundamentals and Beyond Courseware Manual*. Elsevier Inc., USA, 2005.
36. *Adobe Illustrator CS Classroom in a book*. Pearson Education Inc., USA, 2004.
37. *Adobe InDesign CS2 classroo in a book*. Pearson Education Inc., USA, 2004.
38. *Autodesk Maya API - White Paper*. Autodesk Maya Publisher, USA, 2007.
39. Bain St., N. Wilkinson. *CorelDRAW 11: The Official Guide*. The McGraw-Hill Componies, USA, 2002.
40. Bezier, P. E. *Numerical Control: Mathematics and Applications*. New York: John Wiley, 1972.
41. Bezier, P. E. *The mathematical basis of the UNISURF CAD System*. Butteworth, London, 1986.
42. Bloomenthal, J. *Skeletal Design of Natural Forms*. PhD Dissertation, University of Calgary, Dept. of Computer Science, Calgary, 1995.
43. Bloomenthal, M. D. *Error bounded approximate reparametrization of non-uniform B-spline curves*. University of Utah, USA, 1999.
44. Boehm, W. *Generating the Bezier Points of B-Spline Curves and Surfaces*. Computer Aided Design, Vol.13, November, 1981. pp. 365-366.

45. Botsch, Mario, L. Kobbelt. *An Intuitive Framework for Real-Time Freeform Modeling*. SM'04, ACM, 2004.
46. Buss, S. R. *3-D Computer Graphics - A Mathematical Introduction with OpenGL*. Cambridge University Press, UK, 2003.
47. Carlson, W. *A Critical History of Computer Graphics and Animation*. Ohio State University, USA, 2003.
48. Casciola G., S. Morigi. *Reparametrization of NURBS curves*. International Journal of Shape Modelling, Vol 2, Number 2&3, 1996, pp. 103-116.
49. Cox, M. G. *The numerical evaluation of B-splines*. Jour. Inst. Math. Applic., Vol. 10, 1972, pp.134-149.
50. Criado, C., N. Alamo. *Optical Properties of the Conics Derived from an Elementary Kinematic Consideration*. Universidad de Malaga, Spain. International Journal "The Physics Teacher", Volume 42, Issue 5, 2004. pp. 272-273.
51. DeLoura, M. A., *An In-Depth Look at Bicubic Bezier Surfaces*. Game Developers Conference, GamaNetwork, CMP Media LLC, 1999.
52. Drucker, D. *Reflection Properties of Curves and Surfaces*. Mathematics Magazine, Vol. 65, No. 3, Jun. 1992, pp. 147-157.
53. Duguet, J. C. *Scientific APL2 Computing: Bezier, BSplines, NURBS (Curves & Surfaces)*. APL Conference, Volume 32, Number 1, 2001.
54. English, J. *Macromedia Flash 8 – Training from the Source*. Pearson Education Inc., USA, 2006. 416 p.
55. Fellner, D. W., C. Helmberg. *Robust Rendering of General Ellipses and Elliptical Arcs*. ACM Transactions on Graphics, Vol. 12, No. 3, July 1993. pp. 251-276.
56. Forrest, A. R., *Interactive interpolation and approximation by Bezier polynomials*, The Computer Journal, Vol. 15, No 1, pp. 71-79, 1972. Corrected and updated version in CAD, Vol. 22, No. 9, , 1990. pp. 527-537.
57. Fosner, R. *OpenGL Programming for Windows 95 and Windows NT*. Addison Wesley Longman, Menlo Park, California, 1997.
58. Franklin, D., Br. Patton. *Macromedia Flash 5! Creative Web Animation*. Peachpit Press, USA, 2000.
59. Fujiwara, S. *Optical properties of conic surfaces*. I. Reflecting cone. Journal of the Optical Society of America, vol. 52, issue 3, 1962. pp. 287.
60. Gallier, J. *Curves and Surfaces in Geometric modelling – theory and algorithms*. Morgan Kaufmann Publishers, USA, 2000.
61. Gerald, F. *A History of Curves and Surfaces in CAGD*. Arizona State University, USA, 1998.

62. Gerald, F. *Curvature Continuity and Offsets for Piecewise Conics*. ACM Transactions on Graphics, Vol. 8, No. 2, April 1989. pp. 89-99.
63. Gerald, F. *From Conies to NURBS: A Tutorial and Survey*. IEEE Computer Graphics & Applications, September 1992. pp. 78-86.
64. Gil J., D. Keren. *New approach to the arc length parametrization problem*. Haifa University, Israel, 1997.
65. Glassner, A. *Around and around*. Published by the IEEE Computer Society, 2004.
66. Goldenthal, R., M. Bercovier. *Design of Curves and Surfaces by Multi Objective Optimization*. School of Computer Science and Engineering, Hebrew University of Jerusalem, 2005.
67. Goldman, R. N., J. R. Miller. *Combining Algebraic Rigor with Geometric Robustness For the Detection and Calculation of Conic Sections in the Intersection of Two Natural Quadric Surfaces*. ACM 089791-427-9/91/0006/02211991, 1991.
68. Grahn, H., T. Volk, H. J. Wolters. *NURBS in VRML*. ACM: VRML 2000, Monterey, CA USA, 2000.
69. Guenter, B., R. Parent. *Computing the Arc Length of Parametric Curves*. IEEE Computer Graphics & Applications, 1990.
70. Harvey, J. E., P. L. Thompson, A. Krywonos. *Hyperboloid-hyperboloid grazing incidence x-ray telescope designs for wide-field imaging applications*. The International Society for Optical Engineering, CREOL/Univ. of Central Florida, USA, 2003.
71. Jerry, V. A., M. Novak. *Curve-Drawing Algorithms for Raster Displays*. ACM Transactions on Graphics, Vol. 4, No. 2, April 1985, pp. 147-169.
72. Jianying, H., Th. Pavlidis. *Function Plotting Using Conic Splines*. IEEE Computer Graphics & Applications, January 1991, pp. 89-94.
73. Juan, V., M. Zorraquino, F. G. Rodríguez, J. L. C. Martín. *Applications of a New Property of Conics to Architecture*. Nexus Network Lurnal: Architecture and Mathematics Online, vol. 3, no. 2, <http://www.emis.de/journals/NNJ/Mar-Gra-Can.html>, 2001.
74. Judelman, Gr. *Aesthetics and Inspiration for Visualization Design: Bridging the Gap between Art and Science*. International School of New Media, Universitat zu Lubeck, Germany, 2005.
75. Kase, K., Y. Teshima, S. Usami, H. Ohmori, C. Teodosiu, A. Makinouchi. *Volume CAD*. The Eurographics Association, 2003.
76. Kasik, D. J., W. Buxton, D. R. Ferguson. *Ten CAD Challenges*. IEEE Computer Graphics and Applications, 2005.

77. Kramer, J., J. Magee, M. Sloman. *Configuration Support for System Description, Construction and Evolution*. Publications Department of ACM O-89791 -305-I /89/0500/0028, 1989.
78. Lowther, J. L., Ching-Kuang Shene. *Computing with Geometry as an Undergraduate Course: A Three-Year Experience*. ACM: SIGCSE 2001 2/01 Charlotte, NC, USA, 2001.
79. Luh, H. S. *Equivalent hyperboloid (ellipsoid) and its application*. Antennas and Propagation, IEEE Transactions on Volume 48, Issue 4, Apr 2000. pp. 581-584.
80. Luken, W. L., F. Cheng. *Comparison of Surface and Derivative Evaluation Methods for the Rendering of NURB Surfaces*. ACM Transactions on Graphics, Vol. 15, No. 2, April 1996, pp. 153–178.
81. Mauzit, V. *Hyperbolic paraboloid shelters and retractable structures*. Proceedings of the 5th International Bamboo Congress and the 6th International Bamboo Workshop, San José, Costa Rica, 2-6 November 1998, 2002.
82. *Maya API White Paper*. Alias|Wavefront, Toronto, ON, Canada M5A 1J7, 2007.
83. McIlroy, M. D. *Getting Raster Ellipses Right*. ACM Transactions on Graphics, Vol. 11, No. 3, July 1992. pp. 259-275.
84. Meade, T., S. Arima. *Maya 6: The Complete Reference*. The McGraw-Hill Companies, USA, 2005.
85. Morin, G. *Analytic Functions in Computer Aided Geometric Design*, Doctoral dissertation, Rice University, Houston, Texas, 2001.
86. Mueller, T. I. *Geometric Modelling With Multivariate B-Splines*. A dissertation from the University of Utah, USA, 1986.
87. Nelson, J. *Modern Optical Design*. NATO Summer School: Optics in Astrophysics, 2002.
88. *OpenGL Super Bible*. Terms & Conditions, EarthWeb Inc., 2000.
89. Page, K. A. *Macromedia Dreamweaver MX: Training from the Source*. Pearson Education Inc., USA, 2006.
90. Pavlidis, Th. *Curve Fitting with Conic Splines*. ACM Transactions on Graphics, Vol. 2, No. 1, January 1983. pp. 1-31.
91. Piegl, L., W. Tiller. *The NURBS book, 2nd edition*. Springer, New York, USA, 1997.
92. Piegl, L. *On NURBS: A Survey*. Published by the IEEE Computer Society, 1991.
93. Pottmann, H. *Locally Controllable Conic Splines with Curvature Continuity*. ACM Transactions on Graphics, Vol. 10, No. 4, October 1991. pp. 366-377.
94. Prosser, C. J., Alistair C. Kilgour. *An integer method for the graphical output of conic sections*. ACM Transactions on Graphics, Vol. 2, No. 3, July 1983, Pages 182-191.

95. Razdan, Ansh. *Knot Placement for B-Spline Curve Approximation*. Arizona State University, Tempe AZ 85287-5106, USA, 1999.
96. Rogers, D. F. *Procedural elements for Computer Graphics*. McGraw-Hill, Inc. New York, NY, USA, 1984.
97. Rogers, D. F., J. A. Adams. *Mathematical elements for computer graphics, second edition*. McGraw-Hill, USA, 1990.
98. Rogers, D. F. *An Introduction to NURBS*. David F. Rogers and Morgan Kaufmann Publishers, 2000.
99. Sánchez-Reyes, J. A *A Simple Technique for NURBS Shape Modification*. IEEE Computer Graphics and Applications, January-February 1997. pp. 52-59.
100. Stuart, M. *Mac 3D: Three-Dimensional Modelling and Rendering in the Macintosh*. Intellect, Oxford, England, 1994.
101. Terzopoulos, D. *Dynamic NURBS with Geometric Constraints for Interactive Sculpting*. ACM Transactions on Graphics, Vol 13, No. 12, April 1994. pp. 103-136.
102. Vaughan, Pr. *Techniques for Conic Splines*. San Francisco, July 22-26 Volume 19, Number 3, 1985.
103. Walter, M., A. Fournier. *Approximate Arc Length Parametrization*. University of British Columbia, 1996.
104. Wang, H., J. Kearney, K. Atkinson. *Arc-Length Parameterized Spline Curves for Real-Time Simulation*. Curve and Surface Design: Saint-Malo 2002. Nashboro Press, Brentwood, TN, 2002, pp. 387–396.
105. Wang, H., J. Kearney, K. Atkinson. *Curve and Surface Design*. Saint-Malo 2002, Nashboro Press, Brentwood, TN, 2002, pp. 387–396.
106. Warren, J. *Subdivision methods for geometric design*. Department of Computer Science, Rice University, 1995.
107. Woodward, Ch. D. *Methods for Cross-Sectional Design of B-Spline Surfaces*. Eurographics'86, European Computer Graphics Conference and Exhibition, Lisbon, Portugal, 1986.
108. Yang, X. N. *Approximating NURBS curves by arc splines*. Proc. of Geometric Modeling and Processing'2000, Hongkong, IEEE Computer Society, 2000. pp.10-12.
109. Zirbel, J. H., St. B. Combs. *AutoCAD Release 13 for windows*. Que Corporation, USA, 1996.

110. *3D computer graphics software*. Wikipedia.
http://en.wikipedia.org/wiki/3D_computer_graphics_software. 2008.
111. *3ds Max 8 Documentation*. Autodesk.
<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=9861621&linkID=5604642>, 2008.
112. *3ds Max SDK Documentation*. Autodesk,
<http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=7481368>. 2008.
113. *A Medical Application of the Ellipse*. <http://mathcentral.uregina.ca>. 2007.
114. *Applications of Conics in Acoustics*. Mega Essays.
<http://www.megaessays.com/viewpaper/3187.html>. 2007.
115. *Applications of Hyperbolas*.
<http://www.pen.k12.va.us/Div/Winchester/jhhs/math/lessons/calc2004/apphyper.html>. 2004.
116. *Autodesk 3ds Max Detailed Features*. Autodesk.
<http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=8108755>. 2008.
117. *Autodesk Unveils 3D Studio MAX for Windows NT; Technology Innovations Mark Next Generation of Leading Professional 3D Animation Tool*.
http://findarticles.com/p/articles/mi_m0EIN/is_1995_August_8/ai_17118597. Business Wire, 1995.
118. B-spline Surfaces: Construction.
<http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/surface/bspline-construct.html>.
119. Boebert E. *Computing the Arc Length of Cubic Bezier Curves*.
<http://comp.graphics.algorithms>. 1993.
120. Bogomolny A. *The Parabola*. <http://www.cut-the-knot.org/ctk/Parabola.shtml>. 2004.
121. Calvert J. B. *The Hyperboloid and its Applications to Engineering*. Created 6 March 2007, <http://www.du.edu/~etuttle/tech/hyperbo.htm>. 2007.
122. *Comparison of 3d tools*. CG Society of digital artists.
http://wiki.cgsociety.org/index.php/Comparison_of_3d_tools. 2007.
123. *Comparison of OpenGL and Direct3D*. Wikipedia.
http://en.wikipedia.org/wiki/Comparison_of_OpenGL_and_Direct3D. 2008.
124. *Computer-aided design*. Wikipedia. http://en.wikipedia.org/wiki/Computer_aided_design. 2008.
125. *Conic sections*. Classic Encyclopedia: http://www.1911encyclopedia.org/Conic_section. 2007.

126. Damer B. *World Building Tools*.
<http://www.digitalspace.com/AVATARS/book/fullbook/chow/chow2.htm>. 1997.
127. *DirectX Graphics C/C++ Tutorials*. http://msdn.microsoft.com/archive/en-us/dx8_c/directx_cpp/graphics_tut_1icz.asp. 2008.
128. *Ellipse*. <http://www.du.edu/~jcalvert/math/ellipse.htm>. 2008.
129. *European CG network "TDT 3D"*. http://www.tdt3d.be/articles_viewer.php?art_id=99. 2008.
130. *Famous Curves*. <http://curvebank.calstatela.edu/famouscurves/famous.htm>. 2007.
131. *Graphic art software*. Wikipedia. http://en.wikipedia.org/wiki/Graphic_art_software. 2008.
132. *Interactive Gallery of Quadric Surfaces*. <http://mathdl.maa.org/>. 2007.
133. *Key 3ds Max Features for Design Visualisation Professionals*.
<http://www.autodesk.co.uk/adsk/servlet/item?siteID=452932&id=9898959>. 2008.
134. *List of vector graphics editors*. Wikipedia.
http://en.wikipedia.org/wiki/List_of_vector_graphics_editors. 2008.
135. *Maplesoft Documentation Center*. http://www.maplesoft.com/documentation_center/. 2008.
136. *Occurrence of the conics*. <http://britton.disted.camosun.bc.ca/jbconics.htm>. 2007.
137. *OpenGL FAQ and Troubleshooting Guide v1.2001.11.01*.
<http://www.opengl.org/resources/faq/technical/>. 2001.
138. *Parabola*. <http://www.du.edu/~jcalvert/math/parabola.htm>,
<http://www.sacredscience.com/archive/Parabola.htm>. 2008.
139. *Properties of conics*. Math Education and Technology, IES.
<http://www.ies.co.jp/math/java/conics/>. 2007.
140. Schmarge K. *Conic Sections in Ancient Greece*. History of Mathematics Term Paper.
<http://www.math.rutgers.edu/~cherlin/History/Papers1999/schmarge.html>. 1999.
141. *Special Relativity and Conic Sections*. <http://mathdl.maa.org/>. 2007.
142. *The Paraboloid and Elliptical Paraboloid*.
<http://curvebank.calstatela.edu/paraboloid/paraboloid.htm>. 2007.
143. *Vector graphics editor*. Wikipedia. http://en.wikipedia.org/wiki/Vector_graphics_editor. 2008.
144. *Winchester Public Schools*.
<http://www.pen.k12.va.us/Div/Winchester/jhhs/math/lessons/calc2004/>. 2004.