

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Počítačová podpora výuky geometrie

Anotace

Bakalářská práce se zabývá vytvořením softwarového nástroje, který umožní zobrazovat vybrané geometrické objekty a jejich průniky. Mezi tyto objekty patří kvadriky, kuželosečky, roviny, přímky a body. Každý objekt může být zadán různými typy rovnic, které tento objekt jednoznačně určují. Nástroj umožní měnit koeficienty v jednotlivých rovnicích a aplikovat transformace rotace a posunutí na jednotlivé objekty. Pro praktickou využitelnost budou podporovány standardní operace jako jsou uložení a načtení obsahu, úpravy vpřed a zpět, nápověda a nastavení různých možností zobrazení.

Děkuji Mgr. Tomáši Kührovi, Ph.D. za jeho rady a vedení mé bakalářské práce.
Dále chci poděkovat Mgr. Romanu Haškovi, Ph.D. za jeho rady, které mi pomohly
pochopit, jak nalézt transformační matici kvadriky.

Obsah

1. Úvod	8
1.1. Cíle práce	8
1.2. Existující řešení	8
1.3. Volba geometrických konvencí	9
2. Použité algoritmy	10
2.1. Klasifikace kvadrik	10
2.2. Klasifikace kuželoseček	13
2.3. Vzájemná poloha dvou geometrických objektů	15
2.3.1. Vzájemná poloha dvou přímek	15
2.3.2. Vzájemná poloha roviny a přímky	15
2.3.3. Vzájemná poloha dvou rovin	16
2.3.4. Vzájemná poloha kuželosečky a přímky	17
2.3.5. Vzájemná poloha kuželosečky a roviny	19
2.3.6. Vzájemná poloha kvadriky a přímky	19
2.3.7. Vzájemná poloha kvadriky a roviny	20
2.3.8. Vzájemná poloha kvadriky a kuželosečky	21
2.3.9. Vzájemná poloha dvou kvadrik	21
2.3.10. Vzájemná poloha dvou kuželoseček	25
2.4. Vykreslování geometrických objektů	27
2.5. Ořezávání grafických objektů	29
2.6. Ořezávání křivky průniku dvou kvadrik	31
3. Programátorská příručka	33
3.1. Matematika	33
3.1.1. Vector	33
3.1.2. Matrix	33
3.1.3. Interval	33
3.1.4. Algebraic Equations	33
3.1.5. EquationSystem	34
3.1.6. CommonEquation	34
3.1.7. ParametricEquation	34
3.2. Grafické objekty	35
3.2.1. GraphicObject	35
3.2.2. WithParametricEquationObject	36
3.2.3. WithCommonEquationObject	37
3.2.4. WithCanonicEquationObject	37
3.2.5. ConicSection	38
3.3. Manipulace s objekty	38
3.3.1. ManipulationManager	38
3.3.2. Editační dialogy	38

3.4.	Vytvoření grafické reprezentace rovnice	39
3.4.1.	Formátovací řetězec	39
3.4.2.	MemberControl	39
3.4.3.	LabelControl	39
3.4.4.	FracControl	39
3.4.5.	SqrtControl	40
3.4.6.	CombinedControl	40
4.	Uživatelská příručka	41
4.1.	Hlavní okno programu	41
4.2.	Manipulace se scénou	41
4.3.	Kontextové menu	42
4.4.	Přidání geometrického objektu	42
4.5.	Editace geometrických objektů	43
4.6.	Historie	44
4.7.	Uložení	44
4.8.	Načtení	44
4.9.	Nová scéna	44
4.10.	Export do obrázku	45
4.11.	Nastavení	45
4.12.	Uživatelská nápověda	45
4.13.	Ukončení programu	45
	Závěr	46
	Reference	47
	A. Obsah přiloženého CD	48

Seznam obrázků

1.	Hlavní okno programu	41
2.	Detail grafického objektu	43

Seznam tabulek

1.	Přehled středových singulárních kvadrik	11
2.	Přehled středových regulárních kvadrik	11
3.	Přehled nestředových singulárních kvadrik	12
4.	Přehled nestředových regulárních kvadrik	12
5.	Přehled středových kuželoseček	14
6.	Přehled nestředových kuželoseček	14
7.	Přehled simple ruled kvadrik	22
8.	Přehled simple ruled kuželoseček	26

1. Úvod

1.1. Cíle práce

Cílem práce je vytvoření softwarového nástroje umožňujícího zobrazování vybraných geometrických objektů, mezi které patří kvadriky, kuželosečky, roviny, přímky a body. Nástroj umožní editaci jednotlivých koeficientů v rovnicích objektu a tím i sledování změn tvaru a pozice objektu v závislosti na zadaných koeficientech. Dále nástroj umožní spočítání průniku libovolných dvou objektů a tento průnik graficky zobrazí. Všem objektům bude dále umožněno měnit jejich vlastnosti jako jsou viditelnost, zobrazení os, barva a jméno objektu. Nástroj bude podporovat základní operace pro uložení a načtení souboru, uložení aktuálního obrázku scény do souboru a úpravy vpřed a zpět. Pro seznámení s ovládáním, bude nástroj obsahovat stručnou uživatelskou nápovědu.

Výsledný program může pomoci studentům středních a vysokých škol k získání základních představ o vybraných objektech.

1.2. Existující řešení

Existuje řada softwarových nástrojů zobrazujících geometrické objekty. Mezi jejich autorem známé představitele patří následující výčet:

1. GNUplot je platformově nezávislý nástroj obsahující mnoho funkcionality. Umožňuje například vykreslování objektů zadaných explicitní rovnicí, přičemž tyto rovnice mohou obsahovat různé matematické funkce jako jsou sin, cos, log, statistické funkce a další speciální funkce. GNUplot však nepodporuje současné zobrazení více objektů v jedné scéně a tedy ani vyšetřování vzájemné polohy zadaných objektů.
2. Archimedes Geo3D je nástroj nabízející podobnou funkcionalitu jako vytvořený program. Jeho GUI je velmi bohaté a přehledné. Umožňuje zadání vybraných geometrických objektů, určení a vykreslení jejich průniku. Na rozdíl od vytvořeného programu, Archimedes Geo3D nepodporuje zadání libovolné kvadriky nebo kuželosečky. Tento program preferuje zadávání objektů specifikováním kontrolních bodů. S objekty se potom dá manipulovat pouze myší.
3. Kuželosečky je nástroj zaměřený pouze na kuželosečky, přímky a body. Tento nástroj umožňuje demonstraci postupů při řešení úloh týkajících se kuželoseček. Vzájemnou polohu dvou kuželoseček však vyšetřit neumí. Na rozdíl od předchozích nástrojů má tento velmi dobře vyřešenu možnost editace zadaných objektů.

1.3. Volba geometrických konvencí

1. V celém textu budeme předpokládat základní znalosti vektorových a afinních prostorů. Není-li uvedeno jinak budeme pod označením V chápat afinní prostor, který vznikl zavedením kanonické struktury afinního prostoru na vektorovém prostoru dimenze 3. Vektory z V zapisujeme do kulatých závorek, body z V zapisujeme do hranatých závorek.
2. Existují dvě konvence reprezentace transformační matice. Budeme respektovat konvenci OpenGL a textů [2, 3, 4]. Podle této konvence se řádkový vektor souřadnic násobí transformační maticí zprava.
3. Program umožňuje uživateli měnit rotaci objektu vzhledem k souřadným osám. Pro specifikování této rotace budeme používat trojici úhlů α, β, γ , reprezentující dílčí rotace objektu o úhel α okolo souřadné osy x , o úhel β okolo souřadné osy y a o úhel γ okolo souřadné osy z .

2. Použité algoritmy

2.1. Klasifikace kvadrik

Při klasifikaci kvadrik vycházíme z textů [1, 2, 3]. Definice kvadriky v textu [2] říká, že kvadrika je libovolná nenulová kvadratická forma na trojrozměrném vektorovém prostoru. Z toho vychází známější definice pomocí obecné rovnice. Podle této definice za kvadriku považujeme množinu bodů, splňující obecnou rovnici kvadriky

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + Gx + Hy + Kz + L = 0. \quad (2.1)$$

Tato rovnice lze přepsat do matice kvadriky

$$\begin{pmatrix} A & D & E & G \\ D & B & F & H \\ E & F & C & K \\ G & H & K & L \end{pmatrix}$$

Každý typ kvadriky lze rozpoznat z tzv. *kanonického tvaru* její rovnice. Pro nalezení tohoto tvaru je nutno nejdříve z její obecné rovnice eliminovat členy D, E, F . Toho docílíme vhodnou rotací kvadriky a to tak, že její osy budou splývat se souřadnými osami. Z [2, 3] plyne, že tuto rotaci obdržíme vypočítáním tzv. *vlastních vektorů* kvadriky. Postup pro jejich získání je následující:

1. Vypočítáme tzv. *vlastní čísla* kvadriky. Dá se dokázat, že tato čísla jsou vždy právě 3 včetně násobností. Vlastní čísla získáme vypočítáním kořenů rovnice

$$\begin{vmatrix} A - \lambda & D & E \\ D & B - \lambda & F \\ E & F & C - \lambda \end{vmatrix} = 0.$$

2. Dosadíme vlastní čísla do následující soustavy rovnic a tuto vyřešíme

$$\begin{aligned} (A - \lambda_i)u_1 + Du_2 + Eu_3 &= 0, \\ Du_1 + (B - \lambda_i)u_2 + Fu_3 &= 0, \\ Eu_1 + Fu_2 + (C - \lambda_i)u_n &= 0. \end{aligned}$$

3. Z řešení těchto soustav sestavíme bázi vektorového prostoru V . Opět se dá dokázat, že pro n -násobné vlastní číslo λ_i , získáme řešení soustavy závislé na n parametrech. N -násobnému vlastnímu číslu λ_i tedy připadá n -rozměrný vektorový prostor vlastních vektorů. Protože hledáme bázi V , zvolíme za vlastní vektory příslušné vlastnímu číslu λ_i libovolnou bázi jemu připadajícího prostoru vlastních vektorů. To, že výsledná trojice vektorů tvoří bázi celého V , se dá opět dokázat.

V dalším kroku aplikujeme na matici kvadriky takovou matici posunutí, která v matici kvadriky eliminuje koeficienty G a H , případně i K . Nalezení tohoto posunutí, odpovídá nalezení souřadnic středu kvadriky. Přičemž bod $S = [m, n, p]$ je středem kvadriky právě tehdy, když jeho souřadnice splňují soustavu rovnic

$$\begin{aligned}Am + Dn + Ep + G &= 0 \\ Dm + Bn + Fp + H &= 0 \\ Em + Fn + Cp + K &= 0\end{aligned}$$

Poznámka 1. Informace o tomto kroku lze najít v textech [2, 3].

Tento postup funguje i pokud má kvadrika přímku nebo rovinu středů. Pokud klasifikujeme kvadriku s vrcholem nebo přímkou vrcholů, pak tato soustava nemá řešení. Proto se nabízí druhá možnost nalezení tohoto posunutí, kterou je doplnění na součet čtverců v nové rovnici kvadriky. Nyní bychom měli vždy obdržet kanonický tvar rovnice kvadriky.

Typ kvadriky	Kanonický tvar	Parametrizace
Reálná kuželová plocha	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$	$x(u, v) = av \cdot \cos u$ $y(u, v) = bv \cdot \sin u$ $z(u, v) = cu$
Imaginární kuželová plocha	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 0$	neexistuje

Tabulka 1. Přehled středových singulárních kvadrik

Typ kvadriky	Kanonický tvar	Parametrizace
Jednodílný hyperboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$	$x(u, v) = a \cdot \cosh v \cos u$ $y(u, v) = b \cdot \cosh v \sin u$ $z(u, v) = c \cdot \sinh v$
Dvoudílný hyperboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = -1$	$x(u, v) = a \cdot \sinh v \cos u$ $y(u, v) = b \cdot \sinh v \sin u$ $z(u, v) = c \cdot \cosh v$
Reálný elipsoid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$	$x(u, v) = a \cdot \cos v \cdot \sin u$ $y(u, v) = b \cdot \sin v \sin u$ $z(u, v) = c \cdot \cos u$
Imaginární elipsoid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = -1$	neexistuje

Tabulka 2. Přehled středových regulárních kvadrik

Typ kvadriky	Kanonický tvar	Parametrizace
Reálná eliptická válcová plocha	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	$x(u, v) = a \cdot \cos u$ $y(u, v) = b \cdot \sin u$ $z(u, v) = v$
Imaginární eliptická válcová plocha	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = -1$	neexistuje
Hyperbolická válcová plocha	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$x(u, v) = \pm a \cdot \cosh u$ $y(u, v) = b \cdot \sinh u$ $z(u, v) = v$
Parabolická válcová plocha	$\frac{x^2}{a^2} + 2py = 0$	$x(u, v) = u$ $y(u, v) = -\frac{u^2}{a^2 2p}$ $z(u, v) = v$
Reálné různoběžné roviny	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 0$	$x(u, v) = \pm \frac{ub}{a}$ $y(u, v) = u$ $z(u, v) = v$
Imaginární různoběžné roviny	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 0$	$x(u, v) = m$ $y(u, v) = n$ $z(u, v) = u$
Reálné rovnoběžné roviny	$\frac{x^2}{a^2} = 1$	$x(u, v) = \pm m$ $y(u, v) = u$ $z(u, v) = v$
Imaginární rovnoběžné roviny	$\frac{x^2}{a^2} = -1$	neexistuje
Dvojnásobná rovina	$\frac{x^2}{a^2} = 0$	$x(u, v) = m$ $y(u, v) = u$ $z(u, v) = v$

Tabulka 3. Přehled nestředových singulárních kvadrik

Typ kvadriky	Kanonický tvar	Parametrizace
Eliptický paraboloid	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + 2kz = 0$	$x(u, v) = u$ $y(u, v) = v$ $z(u, v) = (\frac{u^2}{2a^2k} + \frac{v^2}{2b^2k})$
Hyperbolický paraboloid	$\frac{x^2}{a^2} - \frac{y^2}{b^2} + 2kz = 0$	$x(u, v) = u$ $y(u, v) = v$ $z(u, v) = (\frac{u^2}{2a^2k} - \frac{v^2}{2b^2k})$

Tabulka 4. Přehled nestředových regulárních kvadrik

2.2. Klasifikace kuželoseček

Při klasifikaci kuželoseček vycházíme z textů [1, 2, 4]. Definice kuželosečky v textu [2] říká, že kuželosečka je libovolná nenulová kvadratická forma na dvou-rozměrném vektorovém prostoru. Z toho vychází známější definice pomocí obecné rovnice. Podle této definice za kuželosečku považujeme množinu bodů, splňující obecnou rovnici kuželosečky

$$Ax^2 + By^2 + 2Dxy + Gx + Hy + L = 0. \quad (2.2)$$

Tato rovnice lze přepsat do matice kuželosečky

$$\begin{pmatrix} A & D & G \\ D & B & H \\ G & H & L \end{pmatrix}$$

Každý typ kuželosečky lze rozpoznat z tzv. *kanonického tvaru* její rovnice. Pro nalezení tohoto tvaru je nutno nejdříve z její obecné rovnice eliminovat člen D . Toho docílíme vhodnou rotací kuželosečky a to tak, že její osy budou splývat se souřadnými osami. Z [2, 4] plyne, že tuto rotaci obdržíme vypočítáním tzv. *vlastních vektorů* kuželosečky. Postup pro jejich získání je následující:

1. Vypočítáme tzv. *vlastní čísla* kuželosečky. Dá se dokázat, že tato čísla jsou vždy právě 2 včetně násobností. Vlastní čísla získáme vypočítáním kořenů rovnice

$$\begin{vmatrix} A - \lambda & D \\ D & B - \lambda \end{vmatrix} = 0.$$

2. Dosadíme vlastní čísla do následující soustavy rovnic a tuto vyřešíme

$$\begin{aligned} (A - \lambda_i)u_1 + Du_2 &= 0, \\ Du_1 + (B - \lambda_i)u_2 &= 0, \end{aligned}$$

3. Z řešení těchto soustav sestavíme bázi vektorového prostoru V . Opět se dá dokázat, že pro n -násobné vlastní číslo λ_i , získáme řešení soustavy závislé na n parametrech. N -násobnému vlastnímu číslu λ_i tedy připadá n -rozměrný vektorový prostor vlastních vektorů. Protože hledáme bázi V , zvolíme za vlastní vektory příslušné vlastnímu číslu λ_i libovolnou bázi jemu připadajícího prostoru vlastních vektorů. Dá se dokázat, že výsledná dvojice vektorů je ortogonální.

V dalším kroku aplikujeme na matici kuželosečky takovou matici posunutí, která v matici kuželosečky eliminuje koeficienty G , případně i H . Nalezení tohoto posunutí, odpovídá nalezení souřadnic středu kuželosečky. Přičemž bod

$S = [m, n]$ je středem kuželosečky právě tehdy, když jeho souřadnice splňují soustavu rovnic

$$\begin{aligned} Am + Dn + G &= 0 \\ Dm + Bn + H &= 0 \end{aligned}$$

Poznámka 2. Informace o tomto kroku lze najít v textech [2] a [4].

Tento postup funguje i pokud má kuželosečka přímku středů. Pokud klasifikujeme kuželosečku s vrcholem, pak tato soustava nemá řešení. Proto se nabízí druhá možnost nalezení tohoto posunutí, kterým je doplnění na součet čtverců v nové rovnici kuželosečky. Nyní bychom měli obdržet kanonický tvar rovnice kuželosečky.

Typ kuželosečky	Kanonický tvar	Parametrizace
Reálná elipsa	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	$x(u) = a \cdot \cos u$ $y(u) = b \cdot \sin u$
Imaginární elipsa	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = -1$	neexistuje
Hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$x(u) = \pm a \cdot \cosh u$ $y(u) = b \cdot \sinh u$
Různoběžné přímky	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 0$	$x(u) = u$ $y(u) = \pm u \frac{b}{a}$
Imaginární různoběžné přímky	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 0$	$x(u) = 0$ $y(u) = 0$

Tabulka 5. Přehled středových kuželoseček

Typ kuželosečky	Kanonický tvar	Parametrizace
Parabola	$\frac{x^2}{a^2} + 2ky = 0$	$x(u) = u$ $y(u) = -\frac{u^2}{2ka^2}$
Dvojnásobná přímka	$\frac{x^2}{a^2} = 0$	$x(u) = 0$ $y(u) = u$
Rovnoběžné přímky	$\frac{x^2}{a^2} = 1$	$x(u) = \pm a$ $y(u) = u$
Imaginární rovnoběžné přímky	$\frac{x^2}{a^2} = -1$	neexistuje

Tabulka 6. Přehled nestředových kuželoseček

2.3. Vzájemná poloha dvou geometrických objektů

2.3.1. Vzájemná poloha dvou přímek

Mějme přímky p a q na V a jejich parametrické rovnice

$$\begin{aligned}p &: X = A + tu, \\q &: X = B + sv,\end{aligned}$$

kde $A = [a_1, a_2, a_3]$ a $B = [b_1, b_2, b_3]$ jsou počátečními body přímek p a q a $u = (u_1, u_2, u_3)$ a $v = (v_1, v_2, v_3)$ jsou směrovými vektory přímek p a q .

Při vyšetřování vzájemné polohy dvou přímek, mohou nastat čtyři případy:

1. Přímky p a q jsou *totožné*, pokud každý bod přímky p náleží i přímce q . V praxi je to tehdy, jsou-li jejich směrové vektory lineárně závislé a existuje bod M náležející oběma přímkám. Jelikož víme, že bod A náleží přímce p , položíme $M = A$.
2. Přímky p a q jsou *rovnoběžné*, pokud jsou jejich směrové vektory lineárně závislé, ale neexistuje bod náležející oběma přímkám.
3. Přímky p a q jsou *různoběžné*, pokud nejsou jejich směrové vektory lineárně závislé a existuje bod M , který náleží oběma přímkám. Stanovit zda existuje bod M potom znamená vyřešit soustavu rovnic

$$\begin{aligned}a_1 + tu_1 &= b_1 + sv_1, \\a_2 + tu_2 &= b_2 + sv_2, \\a_3 + tu_3 &= b_3 + sv_3.\end{aligned}$$

4. Přímky p a q jsou *mimoběžné*, pokud nejsou jejich směrové vektory lineárně závislé a neexistuje bod náležející oběma přímkám.

2.3.2. Vzájemná poloha roviny a přímky

Mějme danu rovinu $\alpha : ax + by + cz + d = 0$ a přímku $p : X = A + tu$. Při vyšetřování vzájemné polohy roviny a přímky, mohou nastat tři případy:

1. Přímka p leží v rovině α , pokud jsou normálový vektor roviny a směrový vektor přímky ortogonální a zároveň existuje bod M náležející přímce i rovině. Za bod M stačí zvolit počáteční bod A přímky p .
2. Přímka p a rovina α jsou *mimoběžné*, pokud jsou normálový vektor roviny a směrový vektor přímky ortogonální a zároveň neexistuje bod M náležející přímce i rovině. Za bod M zvolíme počáteční bod A přímky p .

3. Přímka p a rovina α jsou *různoběžné*, pokud nejsou normálový vektor roviny a směrový vektor přímky ortogonální. Potom přímka protíná rovinu v bodě M . Tento bod nalezneme dosazením parametrických rovnic přímky do obecné rovnice roviny. Obdržíme lineární rovnici o jedné neznámé

$$a(A_1 + tu_1) + b(A_2 + tu_2) + c(A_3 + tu_3) + d = 0.$$

Po vyjádření neznámé t obdržíme

$$t = -\frac{(aA_1 + bA_2 + cA_3 + d)}{(au_1 + bu_2 + cu_3)}.$$

Tento výraz má smysl, protože ve jmenovateli obsahuje skalární součin normálového vektoru roviny a směrového vektoru přímky. Protože tyto vektory nejsou ortogonální, tento skalární součin není nulový.

Vypočítané t následně dosadíme do parametrických rovnic přímky a obdržíme bod průniku.

2.3.3. Vzájemná poloha dvou rovin

Mějme roviny α a β na V a jejich obecné rovnice

$$\alpha : a_1x + b_1y + c_1z + d_1 = 0, \quad (2.3)$$

$$\beta : a_2x + b_2y + c_2z + d_2 = 0. \quad (2.4)$$

Normálový vektor roviny α označíme n_1 , normálový vektor roviny β označíme n_2 . Při vyšetřování vzájemné polohy rovin α a β , mohou nastat tři případy:

1. Roviny α a β jsou *totožné*, pokud každý bod náležející rovině α náleží i rovině β . V praxi je to tehdy jsou-li normálové vektory rovin α a β lineárně závislé a existuje bod M náležející oběma rovinám.

Jelikož normálový vektor roviny nesmí být nulový, pak bod M roviny α nalezneme vyjádřením jedné z proměnných x, y, z , která není násobena nulovým koeficientem. Předpokládejme, že je to proměnná x , potom libovolně zvolíme y -novou a z -tovou souřadnici bodu M a dopočítáme x -ovou souřadnici pomocí vztahu

$$x = -\frac{b_1y + c_1z + d_1}{a_1}. \quad (2.5)$$

2. Roviny α a β jsou *rovnoběžné*, pokud jsou jejich normálové vektory lineárně závislé, ale neexistuje bod náležející oběma rovinám.

3. Roviny α a β jsou *různoběžné*, pokud jsou jejich normálové vektory lineárně nezávislé. V tom případě se roviny protínají v přímkce. Parametrické rovnice této přímky stanovíme nalezením dvou bodů, jež na této přímce leží. Z obecné rovnice (2.3) první roviny vyjádříme proměnnou, která není násobena nulovým koeficientem. Předpokládejme, že je to proměnná x . Po vyjádření x z rovnice první roviny a dosazení do rovnice druhé roviny, obdržíme jednu rovnici dvou neznámých y a z , která má tvar

$$b_3y + c_3z + d_3 = 0.$$

Jelikož nejsou normálové vektory rovin α a β lineárně závislé, pak tato rovnice obsahuje alespoň jednu proměnnou, která není násobena nulovým koeficientem. Nechť je to proměnná y . Tuto proměnnou vyjádříme a obdržíme výraz ve tvaru

$$y = -\frac{c_3z + d_3}{b_3}.$$

Dosazením libovolných dvou hodnot z_1, z_2 za proměnnou z , obdržíme odpovídající hodnoty y_1, y_2 proměnné y . Finálním dosazením dvojic hodnot y_1, z_1 a y_2, z_2 do vyjádření (2.5), obdržíme hodnoty x_1, x_2 proměnné x .

Parametrické rovnice přímky, jež leží v průniku rovin α a β , již z bodů $M_1 = (x_1, y_1, z_1)$ a $M_2 = (x_2, y_2, z_2)$ sestavíme triviálně.

2.3.4. Vzájemná poloha kuželosečky a přímky

Mějme kuželosečku C ležící v libovolné rovině α a přímku p ,

$$\begin{aligned} C : a_{11}x_1^2 + a_{22}x_2^2 + a_{12}x_1x_2 + a_{13}x_1 + a_{23}x_2 + a_{33} &= 0, \\ p : X &= A + tu. \end{aligned}$$

Při vyšetřování vzájemné polohy kuželosečky a přímky nejprve stanovíme vzájemnou polohu roviny α a přímky p .

Pokud přímka protíná rovinu v bodě M , potom musíme nejdříve získat vyjádření kuželosečky C a bodu M vzhledem ke stejné bázi. Na kuželosečku i bod tedy aplikujeme transformaci T , jež převádí kuželosečku do kanonického tvaru. Nyní leží jak kuželosečka tak bod v rovině β o rovnici $\beta : z = 0$ a můžeme otestovat, zda nové souřadnice M splňují novou obecnou rovnici kuželosečky. Pokud souřadnice bodu splňují obecnou rovnici kuželosečky, pak přímka kuželosečku v tomto bodě *protíná*, jinak jsou přímka a kuželosečka *mimoběžné*.

Pokud jsou přímka p a rovina α *mimoběžné*, pak jsou *mimoběžné* i kuželosečka C a přímka p .

Leží-li přímka p v rovině α , potom musíme nejdříve získat vyjádření kuželosečky C a přímky p vzhledem ke stejné bázi. Na kuželosečku i přímku tedy aplikujeme transformaci T , jež převádí kuželosečku do kanonického tvaru. Nyní leží jak kuželosečka tak přímka v rovině β o rovnici $\beta : z = 0$ a můžeme dosadit parametrické rovnice přímky p do obecné rovnice kuželosečky C :

$$a_{11}(m+tu)^2 + a_{22}(n+tv)^2 + a_{12}(m+tu)(n+tv) + a_{31}(m+t) + a_{32}(n+tv) + a_{33} = 0.$$

Obdržíme maximálně kvadratickou rovnici parametru t

$$At^2 + 2Bt + C = 0. \quad (2.6)$$

Pro koeficienty A, B, C rovnice (2.6) platí:

$$\begin{aligned} A &= a_{11}u_1^2 + a_{22}u_2^2 + 2a_{12}u_1u_2 \\ B &= u_1(a_{11}m + a_{12}n + a_{13}) + u_2(a_{21}m + a_{22}n + a_{24}) \\ C &= a_{11}m^2 + a_{22}n^2 + 2a_{12}mn + 2a_{31}m + 2a_{32}n + a_{33} \end{aligned}$$

Při řešení rovnice (2.6) mohou nastat tyto případy:

1. $A = 0, B \neq 0$. Pak se rovnice (2.6) redukuje na lineární rovnici, která má jediné řešení t_1 . Přímka p a kuželosečka C se *protínají v jednom bodě*, jehož souřadnice získáme dosazením t_1 za t do parametrických rovnic přímky p . Tento bod převedeme pomocí transformace T do původní soustavy souřadnic.
2. $A = 0, B = 0, C \neq 0$. Pak rovnice (2.6) nemá reálné řešení z čehož plyne, že kuželosečka a přímka jsou *mimoběžné*.
3. $A = 0, B = 0, C = 0$. Pak každé reálné číslo je řešením rovnice (2.6) a každý bod přímky náleží kuželosečce.
4. $A \neq 0, B^2 - AC > 0$. Pak má rovnice (2.6) dva různé reálné kořeny t_1, t_2 . Přímka p a kuželosečka C se *protínají ve dvou bodech*, jejichž souřadnice získáme dosazením t_1 a t_2 za t do parametrických rovnic přímky p . Tyto body převedeme pomocí transformace T do původní soustavy souřadnic.
5. $A \neq 0, B^2 - AC = 0$. Pak má rovnice (2.6) jeden reálný kořen t_1 . *Přímka p je tečnou* kuželosečky C a dotýká se jí v bodě, jehož souřadnice získáme dosazením t_1 za t do parametrických rovnic přímky p . Tento bod převedeme pomocí transformace T do původní soustavy souřadnic.
6. $A \neq 0, B^2 - AC < 0$. Pak rovnice (2.6) nemá reálné řešení z čehož plyne, že kuželosečka a přímka jsou *mimoběžné*.

2.3.5. Vzájemná poloha kuželosečky a roviny

Mějme kuželosečku C v rovině α a rovinu β ,

$$\begin{aligned} C : a_{11}x_1^2 + a_{22}x_2^2 + a_{12}x_1x_2 + a_{13}x_1 + a_{23}x_2 + a_{33} &= 0, \\ \beta : ax + by + cz + d &= 0. \end{aligned}$$

Nejprve určíme vzájemnou polohu roviny α a roviny β . Potom rozlišujeme následující případy:

1. Roviny jsou různoběžné a protínají se v přímce p . Potom určíme vzájemnou polohu kuželosečky C a roviny β jako vzájemnou polohu kuželosečky C a přímky p . Pokud je však přímka p tečnou kuželosečky C , pak říkáme že se rovina β dotýká kuželosečky v bodě dotyku přímky p s kuželosečkou C .
2. Kuželosečka C leží v rovině β , pokud jsou roviny α a β totožné.
3. Pokud jsou roviny α a β mimoběžné, pak jsou *mimoběžné* i kuželosečka C a rovina β .

2.3.6. Vzájemná poloha kvadriky a přímky

Mějme kvadriku K a přímku p ,

$$\begin{aligned} K : a_{11}x_1^2 + a_{22}x_2^2 + a_{33}x_3^2 + 2a_{12}x_1x_2 + 2a_{13}x_1x_3 + 2a_{23}x_2x_3 + \\ 2a_{14}x_1 + 2a_{24}x_2 + 2a_{34}x_3 + a_{44} &= 0. \\ p : A + tu & \end{aligned} \tag{2.7}$$

Provedeme dosazení parametrických rovnic přímky p do obecné rovnice kvadriky K . Tímto obdržíme maximálně kvadratickou rovnici parametru t :

$$At^2 + 2Bt + C = 0, \tag{2.8}$$

přičemž koeficienty A, B, C této rovnice mají tvar:

$$\begin{aligned} A &= a_{11}u_1^2 + a_{22}u_2^2 + a_{33}u_3^2 + 2a_{12}u_1u_2 + 2a_{13}u_1u_3 + 2a_{23}u_2u_3, \\ B &= u_1(a_{11}m + a_{12}n + a_{13}p + a_{14}) + u_2(a_{21}m + a_{22}n + a_{23}p + a_{24}) + \\ &\quad u_3(a_{31}m + a_{32}n + a_{33}p + a_{34}), \\ C &= a_{11}m^2 + a_{22}n^2 + a_{33}p^2 + 2a_{12}mn + 2a_{13}mp + 2a_{23}np + \\ &\quad 2a_{14}m + 2a_{24}n + 2a_{34}p + a_{44}. \end{aligned}$$

Podle hodnot koeficientů A, B, C a počtu řešení rovnice (2.8) rozlišujeme následující případy.

1. $A \neq 0, B^2 - 4AC > 0$. Pak má rovnice (2.8) dva různé reálné kořeny t_1, t_2 a po dosazení těchto kořenů za t do parametrických rovnic (2.7) přímky p obdržíme *dva body průniku* kvadriky a přímky.

2. $A \neq 0, B^2 - 4AC = 0$. Pak má rovnice (2.8) jeden dvojnásobný reálný kořen t_1 a přímka je *tečnou* kvadriky. *Bod dotyku* získáme dosazením kořene t_1 za t do parametrických rovnic (2.7) přímky.
3. $A \neq 0, B^2 - 4AC < 0$. Pak rovnice (2.8) nemá řešení a průnik kvadriky a přímky je *prázdný*.
4. $A = 0, B \neq 0$. Pak má rovnice (2.8) jeden reálný kořen t_1 a po dosazení kořene t_i za t do parametrických rovnic (2.7) přímky p obdržíme *bod průniku* kvadriky a přímky.
5. $A = 0, B = 0, C \neq 0$. Pak rovnice (2.8) nemá řešení a průnik kvadriky a přímky je *prázdný*.
6. $A = 0, B = 0, C = 0$. Pak má rovnice (2.8) nekonečně mnoho řešení a přímka leží na kvadrice.

2.3.7. Vzájemná poloha kvadriky a roviny

Mějme kvadriku K a rovinu α . Prvním krokem je aplikace takové rotace T na rovinu α , jež tuto rovinu ztotožní s rovinou β danou rovnicí

$$\beta : z = 0. \quad (2.9)$$

Následně tuto rotaci aplikujeme i na kvadriku K a dosadíme za z -tovou souřadnici v obecné rovnici orotované kvadriky K hodnotu nula. Tím obdržíme rovnici jež má tvar

$$a_{11}x^2 + a_{22}y^2 + a_{12}xy + a_{41}x + a_{42}y + a_{44} = 0. \quad (2.10)$$

Podle tvaru rovnice (2.10) rozlišujeme následující čtyři případy.

1. Pokud je alespoň jeden z koeficientů a_{11}, a_{22}, a_{12} různý od nuly, pak je rovnice (2.10) rovnicí kuželosečky. V tomto případě je tedy *průnikem* kvadriky a roviny *kuželosečka*.
2. Pokud jsou všechny koeficienty a_{11}, a_{22} i a_{12} nulové, ale alespoň jeden z koeficientů a_{41} nebo a_{42} není nulový, pak je *průnikem* kvadriky a roviny *přímka*.
3. Jsou-li všechny koeficienty v rovnici (2.10) nulové, pak je *průnikem* kvadriky a roviny *rovina* α .
4. Pokud je v rovnici (2.10) nenulový pouze koeficient a_{44} , pak je *průnik* kvadriky a roviny *prázdný*.

2.3.8. Vzájemná poloha kvadriky a kuželosečky

Označme vyšetřovanou kvadriku Q , vyšetřovanou kuželosečku C a rovinu ρ v níž kuželosečka leží. Nejprve učíme vzájemnou polohu kvadriky Q a roviny ρ . Pro vzájemnou polohu kvadriky a kuželosečky pak mohou nastat čtyři případy

1. Kvadrika Q a rovina ρ se protínají v rovině ρ . Potom se kvadrika Q a kuželosečka C protínají v kuželosečce C .
2. Kvadrika a rovina se protínají v přímce p . Potom učíme vzájemnou polohu kuželosečky C a přímky p . Pokud je přímka tečnou kuželosečky v bodě t_1 , pak se kuželosečka a kvadrika v tomto bodě dotýkají. Jinak určíme vzájemnou polohu kuželosečky a kvadriky jako vzájemnou polohu kuželosečky a přímky.
3. Kvadrika a rovina se protínají v kuželosečce C_2 . Potom učíme vzájemnou polohu kvadriky a kuželosečky jako vzájemnou polohu kuželosečky C a kuželosečky C_2 .
4. Kvadrika a rovina jsou mimoběžné, pak jsou mimoběžné i kuželosečka a kvadrika.

2.3.9. Vzájemná poloha dvou kvadrik

V této kapitole se objevují pojmy kvadratická forma a matice kvadratické formy. Jejich definice lze nalézt v textu [2].

Pro určení vzájemné polohy dvou kvadrik zvolíme metodu, která je univerzální a citovaná v mnoha pracích jako například [5, 6, 7]. Základy této metody prvně publikoval ve svých článcích Joshua Zev Levin. Uvedené postupy jsem čerpal ze článku [5]. Levinova metoda se opírá o dokázané tvrzení, které říká, že v pencilu dvou různých kvadrik, viz. definice 2.1, existuje alespoň jedna tzv. *simple ruled* kvadrika. Dále se dá dokázat, že průnik dvou různých kvadrik P a Q je totožný s průnikem *simple ruled* kvadriky R z pencilu kvadrik P a Q a jednou z kvadrik P nebo Q , která je různá od R .

Mějme dány kvadriky P a Q . V případě, že je obecná rovnice kvadriky P nenulovým násobkem obecné rovnice kvadriky Q , pak jsou kvadriky *totožné*.

V případě, že nejsou kvadriky P a Q totožné budeme postupovat podle následujících kroků, které budou později detailněji popsány:

1. Nalezneme *simple ruled* kvadriku R v pencilu vyšetřovaných kvadrik P a Q .

2. Označíme si W tu z kvadrik P a Q , která není totožná s R . Kvadriku R převedeme na kanonický tvar a transformaci T , převádějící R na kanonický tvar, aplikujeme i na kvadriku W .
3. Každou parametrizaci kvadriky R dosadíme do obecné rovnice kvadriky W .
4. Stanovíme definiční obor získané parametrizace $X(u, v)$ průniku kvadrik P a Q .
5. Aplikujeme transformaci T na parametrizaci $X(u, v)$ pro získání parametrizace průniku vzhledem ke kartézské soustavě souřadnic.

Typ kvadriky	Kanonický tvar	Parametrizace
Imaginární různoběžné roviny	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 0$	$x(u, v) = 0$ $y(u, v) = 0$ $z(u, v) = u$
Dvojnásobná rovina	$\frac{x^2}{a^2} = 0$	$x(u, v) = 0$ $y(u, v) = u$ $z(u, v) = v$
Rovnoběžné roviny	$\frac{x^2}{a^2} = 1$	$x(u, v) = \pm a$ $y(u, v) = u$ $z(u, v) = v$
Různoběžné roviny	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 0$	$x(u, v) = au$ $y(u, v) = \pm bu$ $z(u, v) = v$
Parabolická válcová plocha	$\frac{x^2}{a^2} - 2kz = 0$	$x(u, v) = u$ $y(u, v) = \frac{u^2}{a^2 2k}$ $z(u, v) = v$
Hyperbolická válcová plocha	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$x(u, v) = \frac{a}{2}(u + \frac{1}{u})$ $y(u, v) = \frac{b}{2}(u - \frac{1}{u})$ $z(u, v) = v$
Hyperbolický paraboloid	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - 2kz = 0$	$x(u, v) = \frac{\sqrt{a^2 2k(u+v)}}{2}$ $y(u, v) = \frac{\sqrt{b^2 2k(u-v)}}{2}$ $z(u, v) = uv$

Tabulka 7. Přehled simple ruled kvadrik

Simple ruled kvadriky jsou pro nás výhodné, protože jsme pro ně schopni nalézt parametrizaci, jež je maximálně lineární vzhledem k prvnímu parametru, v našem značení vzhledem k v a maximálně kvadratickou vzhledem ke druhému parametru, v našem značení vzhledem k u . Definice simple ruled kvadratických forem je následující.

Definice 2.1. Množina všech kvadratických forem na V tvoří vektorový prostor V_Q . Libovolný podprostor prostoru V_Q se nazývá *pencil*. Pencilem R_{PQ} generovaným kvadratickými formami P a Q rozumíme množinu všech kvadratických forem daných rovnicí

$$R_{PQ} = \lambda P + \mu Q, \quad kde \lambda, \mu \in \mathbb{R} \quad (2.11)$$

Nyní vystává otázka jak určit λ a μ v rovnici (2.11). V tomto nám pomůže důležitá vlastnost simple ruled kvadrik. Všechny simple ruled kvadriky mají totiž nulový determinant jejich matice kvadratické formy. Navíc podle [5] stačí uvažovat $\mu = 1$. Na základě toho, že je determinant matice kvadratické formy simple ruled kvadriky R vždy nulový, platí pro λ v rovnici (2.11) následující rovnice

$$\begin{vmatrix} p_{11} - \lambda q_{11} & p_{12} - \lambda q_{12} & p_{13} - \lambda q_{13} \\ p_{21} - \lambda q_{21} & p_{22} - \lambda q_{22} & p_{23} - \lambda q_{23} \\ p_{31} - \lambda q_{31} & p_{32} - \lambda q_{32} & p_{33} - \lambda q_{33} \end{vmatrix} = 0. \quad (2.12)$$

Roznásobením rovnice (2.12) získáme kubickou rovnici o neznámé λ . Po jejím vyřešení a dosazení kořene λ_i do rovnice (2.11), získáme matici kvadriky R_i . Tato kvadrika nemusí být jednou z kvadrik z tabulky (7.), jelikož simple ruled kvadriky nejsou jediné, které mají nulový determinant matice kvadratické formy. Nicméně alespoň jedna simple ruled kvadrika se mezi kvadrikami R_i vyskytuje. Pokud je *prázdná množina* jedinou simple ruled kvadrikou mezi kvadrikami R_i , pak mají kvadriky P a Q *prázdný průnik*.

V dalším kroku převedeme nalezenou simple ruled kvadriku R na kanonický tvar pomocí metod uvedených v kapitole 2.1. Kvadrice R v kanonickém tvaru potom přísluší jedna z parametrizací z tabulky (7.). Označme si parametrizaci R písmenem X a transformaci jež převádí R na kanonický tvar písmenem T .

Nyní z kvadrik P a Q vybereme tu jež je různá od R . Jedna z nich taková jistě je, protože předpokládáme že $P \neq Q$. Nechť je touto kvadrikou kvadrika P . Na kvadriku P nyní aplikujeme transformační matici T , jež převádí simple ruled kvadriku R do kanonického tvaru. Označme matici transformované kvadriky P písmenem A . Nyní máme obě kvadriky R i P ve stejné bázi a můžeme provést dosazení parametrizace X do obecné rovnice P . Toto můžeme zapsat maticově

$$XAX^T = 0.$$

Tato rovnice je maximálně kvadratická vzhledem k v . Přepsáním na kvadratickou rovnici získáme tvar

$$a(u)v^2 + b(u)v + c(u) = 0. \quad (2.13)$$

Vzhledem k parametrizacím simple ruled kvadrik z tabulky (7.) může tato rovnice obsahovat i členy, které mají nad u záporný exponent. Nicméně i po vynásobení celé rovnice potřebnou mocninou u , má tato rovnice následující vlastnosti.

- 1) $a(u)$ je maximálně kvadratická funkce vzhledem k u .
- 2) $b(u)$ je maximálně kvadratická funkce vzhledem k u .
- 3) $c(u)$ je maximálně kvartická funkce vzhledem k u .
- 4) $b(u)^2 - 4a(u)c(u)$ je maximálně kvartická funkce vzhledem k u .

Nyní může nastat několik případů.

1. Je-li tato rovnice triviálně splněna, pak je průnikem kvadrika jejíž parametrizace je X .
2. Není-li tato rovnice nikdy plněna, je průnik prázdný.
3. Je-li tato rovnice ve tvaru $uv = 0$, pak jsou průnikem dvě křivky, jejichž parametrizace získáme dosazením nuly za u a pak za v do X . Pokud je simple ruled kvadrikou hyperbolická válcová plocha, pak přeskóčíme dosazení hodnoty nula za u do parametrizace X , protože tato parametrizace není v tomto bodě definovaná.
4. Je-li v této rovnici $a(u) = 0, b(u) = 0$, pak rovnici $c(u) = 0$ vyřešíme a dosazením získaných hodnot u do X , obdržíme parametrizace křivek průniku. Opět, pokud je simple ruled kvadrikou hyperbolická válcová plocha, pak přeskóčíme dosazení hodnoty nula za u do parametrizace X .
5. Jinak z této rovnice vyjádříme v a jeho dosazením do X , obdržíme parametrizaci průniku.

Definiční obor parametrizace průniku stanovíme jako množinu všech možných hodnot parametru u a to na základě vyjádření parametru v .

1. Je-li toto vyjádření ve tvaru řešení kvadratické rovnice

$$v = \frac{-b(u) \pm \sqrt{b(u)^2 - 4a(u)c(u)}}{2a(u)},$$

pak vyřešíme rovnici $b(u)^2 - 4a(u)c(u) = 0$ a určíme intervaly hodnot parametru u , pro které je tato rovnice nezáporná. Od těchto intervalů odebereme řešení rovnice $a(u) = 0$.

2. Je-li toto vyjádření ve tvaru řešení lineární rovnice

$$v = -\frac{c(u)}{b(u)},$$

pak za definiční obor křivky položíme interval $(-\infty, \infty)$, od kterého odebereme řešení rovnice $b(u) = 0$.

3. Ve všech ostatních případech za definiční obor křivky průniku položíme interval $(-\infty, \infty)$.

Nakonec, je-li simple ruled kvadrikou hyperbolická válcová plocha, je třeba od definičního oboru odebrat bod nula, protože její parametrizace není v tomto bodě definovaná.

Na každou získanou parametrizaci aplikujeme transformaci T , čímž obdržíme její vyjádření vzhledem ke kartézské soustavě souřadnic.

2.3.10. Vzájemná poloha dvou kuželoseček

V této kapitole se objevují pojmy kvadratická forma a matice kvadratické formy. Jejich definice, lze nalézt v textu [2].

Mějme danu kuželosečku P ležící v rovině α a kuželosečku Q ležící v rovině β .

Kuželosečky ležící ve stejné rovině

Pro určení vzájemné polohy dvou kuželoseček používáme stejný postup jako je uveden v kapitole 2.3.9.

V případě, že nejsou kuželosečky P a Q totožné budeme postupovat podle následujících kroků:

1. Nalezneme simple ruled kuželosečku R v pencilu vyšetřovaných kuželoseček P a Q .
2. Označíme si W tu z kuželoseček P a Q , která není totožná s R . Kuželosečku R převedeme na kanonický tvar a transformaci T , převádějící R na kanonický tvar, aplikujeme i na kuželosečku W .
3. Dosadíme parametrizace kuželosečky R do obecné rovnice transformované kuželosečky W .
4. Aplikujeme transformaci T na parametrizaci $X(u)$ průniku kuželoseček P a Q pro získání parametrizace průniku vzhledem ke kartézské soustavě souřadnic.

Nechť máme nalezenou simple ruled kuželosečku R z pencilu kuželoseček P a Q . Dále nechť T je transformační matice převádějící simple ruled kuželosečku R na kanonický tvar a A je matice transformované kuželosečky P , jež byla zvolena v druhém kroku. Označme X parametrizaci simple ruled kuželosečky R , vzhledem k jejímu kanonickému tvaru.

Dosazení parametrizace X do obecné rovnice lze zapsat maticově ve tvaru

$$XAX^T = 0. \quad (2.14)$$

Typ kuželosečky	Kanonický tvar	Parametrizace
Hyperbola	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$	$x(u) = \frac{a}{2}(u + \frac{1}{u})$ $y(u) = \frac{b}{2}(u - \frac{1}{u})$
Parabola	$\frac{x^2}{a^2} + 2ky = 0$	$x(u) = u$ $y(u) = -\frac{u^2}{2ka^2}$
Dvojnásobná přímka	$\frac{x^2}{a^2} = 0$	$x(u) = 0$ $y(u) = u$
Různoběžné přímky	$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 0$	$x(u) = u$ $y(u) = \pm u \frac{b}{a}$
Rovnoběžné přímky	$\frac{x^2}{a^2} = 1$	$x(u) = \pm a$ $y(u) = u$

Tabulka 8. Přehled simple ruled kuželoseček

Tato rovnice je maximálně kvartická vzhledem k parametru u . Vyřešením této rovnice obdržíme až čtyři hodnoty u_1, \dots, u_n parametru u , které následně dosadíme do parametrizace X a obdržíme body X_1, \dots, X_n . Pokud je tato rovnice triviálně splněna pro každé u , potom se kuželosečky P a Q protínají v kuželosečce, jejíž parametrizace je X .

Nakonec na parametrizaci X , případně body X_1, \dots, X_n získané vyřešením rovnice (2.14) a dosazením do X , aplikujeme transformaci T , pro získání jejich vyjádření vzhledem ke kartézské soustavě souřadnic.

Kuželosečky ležící v různých rovinách

Pokud nejsou roviny α a β kuželoseček P a Q totožné, potom nejdříve určíme jejich vzájemnou polohu.

Jsou-li roviny α a β mimoběžné, pak jsou *mimoběžné* i kuželosečky P a Q .

Pokud jsou roviny různoběžné, pak se protínají v přímce p . Potom určíme vzájemnou polohu přímky p a kuželosečky P a vzájemnou polohu přímky p a kuželosečky Q . Nyní mohou nastat čtyři situace:

1. Přímka p leží na obou kuželosečkách. Potom se kuželosečky *protínají* v přímce p .
2. Přímka leží na jedné z kuželoseček a s druhou se protíná v bodech X_1, \dots, X_n . Potom každý bod X_i náležející oběma kuželosečkám přidáme do množiny I . Pokud je množina bodů I neprázdná, pak říkáme že se kuželosečky *protínají* v bodech $X_j \in I$. Jinak jsou kuželosečky P a Q *mimoběžné*.

3. Přímka protíná kuželosečku P v bodech X_1, \dots, X_n a kuželosečku Q v bodech Y_1, \dots, Y_k . Potom položíme $I = \{X_1, \dots, X_n\} \cap \{Y_1, \dots, Y_k\}$. Pokud je množina bodů I neprázdná, pak říkáme že se kuželosečky *protínají* v bodech $X_j \in I$. Jinak jsou kuželosečky *mimoběžné*.
4. Přímka je mimoběžná s jednou z kuželoseček. Potom jsou kuželosečky P a Q *mimoběžné*.

2.4. Vykreslování geometrických objektů

Pro vykreslování geometrických objektů programem využíváme grafickou knihovnu OpenGL. Tato knihovna obsahuje osm základních grafických primitiv, pomocí nichž lze s dávkou trpělivosti vytvořit libovolný trojrozměrný útvar.

Nyní uvedu výčet primitiv s jejich stručným popisem.

1. Vertex (bod)
Nejzákladnější prvek, který je specifikován pomocí svých souřadnic.
2. Lines (úsečky)
Každé dva po sobě jdoucí vertexy definují úsečku.
3. Tringles (trojúhelníky)
Každé tři po sobě jdoucí vertexy specifikují trojúhelník.
4. Quads (čtyřúhelníky)
Každé čtyři po sobě jdoucí vertexy specifikují čtyřúhelník. Je potřeba aby vertexy leželi v jedné rovině.
5. LineStrip (pás úseček)
První dva vertexy definují úsečku. Každý další definuje úsečku danou posledním vertexem předchozí úsečky a novým vertexem.
6. TriangleStrip (pás trojúhelníků)
První tři vertexy definují trojúhelník. Každý další vertex definuje trojúhelník daný posledními dvěma vertexy předchozího trojúhelníku a novým vertexem.
7. QuadStrip (pás čtyřúhelníků)
První čtyři vertexy definují čtyřúhelník. Každé dva další vertexy definují čtyřúhelník daný posledními dvěma vertexy předchozího čtyřúhelníku a dvěma novými vertexy. Opět je potřeba, aby vertexy každého čtyřúhelníku ležely v jedné rovině.
8. TriangleFan (trs trojúhelníků)
První vertex nazveme vrchol. První tři vertexy specifikují trojúhelník. Každý další vertex definuje trojúhelník daný vrcholem, posledním vertexem předchozího trojúhelníku a novým vertexem.

Všechny programem podporované geometrické objekty se vykreslují dosazováním parametrů do jejich parametrických rovnic. Pro většinu objektů jsme schopni nalézt parametrické rovnice, které jsou pro potřeby vykreslování mnohem výhodnější, než je například dosazování do obecné rovnice. Dosazování za parametry v parametrických rovnicích je velmi podobné pro všechny objekty.

Pro každý parametr u a v , v parametrických rovnicích objektu, zvolíme maximální a minimální hodnoty $uMin, uMax$ a $vMin, vMax$ tohoto parametru. Potom za tyto parametry postupně dosazujeme následujícími způsoby.

1. U kvadrik a rovin dosazujeme tak, abychom získali posloupnost bodů, jež po vložení do jednoho bloku, případně více bloků `TriangleStrip`, vytvoří sekvenci pásů trojúhelníků, jež dohromady vytváří trojúhelníkovou síť. Tato trojúhelníková síť představuje aproximaci povrchu daného objektu. Jednotlivé trojúhelníky trojúhelníkové sítě s využitím možností OpenGL vyplníme. Zjednodušený kód pro vygenerování povrchu roviny vypadá takto:

```
for (double v = vMin; v < vMax; v += vStep)
{
    Gl.glBegin(Gl.GL_TRIANGLE_STRIP);
    for (double u = uMin; u <= uMax; u += uStep)
    {
        SetVertex(u, v);
        SetVertex(u, v+vStep);
    }
    Gl.glEnd();
}
```

Zjednodušení tohoto kódu spočívá v tom, že ve skutečnosti je potřeba zajistit, aby oba for cykly opravdu končily v maximálních hodnotách daných parametrů. Což se v tomto případě nestane pokud $uStep$ a $vStep$ nejsou celočíselnými děliteli $(uMax - uMin)$, resp. $(vMax - vMin)$.

Kód pro generování kvadrik je v podstatě stejný. Jediný rozdíl spočívá v tom, že pro některé kvadriky je potřeba provést určení hodnot parametru u , před začátkem každého for cyklu tohoto parametru, tedy pro každou hodnotu parametru v .

2. U křivek, kuželoseček a přímek dosazujeme tak, abychom získali posloupnost bodů, jež po vložení do bloku `LineStrip`, vytvoří sekvenci na sebe navazujících úseček, tvořících aproximaci daného objektu.

Aby objekt působil prostorově, je potřeba scénu nasvítit. S tím je spojena nutnost nastavit každému vertexu grafického objektu, který chceme mít osvětlený, normálový vektor potřebný pro výpočty osvětlení knihovnou OpenGL. Normálové vektory nastavujeme pouze kvadrikám a rovinám. Povolení osvětlení u křivek,

kuželoseček a přímek je nežádoucí. Jelikož ke každé kvadrice i rovině existuje explicitní rovnice R , pak nejsnáze nalezneme souřadnice normálového vektoru v libovolném bodě kvadriky nebo roviny zderivováním této rovnice. Tuto rovnici parciálně zderivujeme podle všech proměnných. Do získaných výrazů $u_1 = \frac{\partial R}{\partial x}$, $u_2 = \frac{\partial R}{\partial y}$ a $u_3 = \frac{\partial R}{\partial z}$ dosadíme souřadnice bodu, jehož normálový vektor počítáme. Z hodnot výrazů v pořadí u_1, u_2, u_3 , pak sestavíme hledaný normálový vektor.

2.5. Ořezávání grafických objektů

Jelikož většina grafických objektů, se kterými program umožňuje pracovat, obsahuje nekonečně mnoho bodů, je nutno řešit ořezávání takových grafických objektů. Proto je stanovena oblast, do které se grafické objekty kreslí. Pro jednoduchost má tato oblast tvar krychle, která je reprezentována pomocí souřadnic středu a délky hrany.

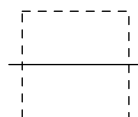
Protože všechny objekty generujeme dosazováním parametrů do jejich parametrických rovnic, musíme pro správné ořezání určit hodnoty těchto parametrů. Ve většině případů budeme tyto hodnoty hledat v jistém základním tvaru daného objektu, který získáme aplikací jisté transformační matice na daný objekt. Pro kuželosečky a kvadriky za tento tvar považujeme tvar kanonický. U rovin to bude rovina o rovnici $z = 0$ a pro přímky přímka o rovnicích $x = t, y = 0, z = 0$. Je-li na objekt v obecné poloze aplikována transformace, je třeba transformovat i střed a tím pádem i hranice ořezávací oblasti. Dále v textu budeme používat konstanty $xMin, xMax, yMin, yMax, zMin$ a $zMax$ což jsou hranice ořezávací oblasti vzhledem k aktuální pozici středu ořezávací oblasti.

1. Je-li parametrizace dané složky daného objektu závislá na jednom parametru

$$x = X(u), \quad x \in (xMin, xMax),$$

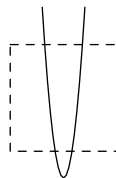
pak tuto rovnici vyřešíme pro hodnoty $xMin$ a $xMax$, čímž obdržíme hodnoty parametru u , pro které daná složka protíná hranice ořezávané oblasti. Vzhledem k charakteru zvolených parametrizací mohou nastat 4 případy.

- (a) Obdržíme-li pro $x = xMin$ hodnotu a a pro $x = xMax$ hodnotu b , pak je výsledný interval hodnot parametru u roven intervalu (a, b) . Tato situace nastává například jedná-li se o přímku, jejíž složka má parametrizaci ve tvaru $x = u$.



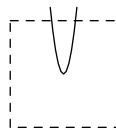
- (b) Obdržíme-li pro $x = xMin$ hodnoty a_1, a_2 a pro $x = xMax$ hodnoty b_1, b_2 , pak z těchto hodnot vytvoříme intervaly $A = (a_1, a_2)$ a $B = (b_1, b_2)$. Pro intervaly A a B určitě platí $A \subseteq B$ nebo $B \subseteq A$, což plyne ze zvolených parametrizací. Za intervaly hodnot parametru u zvolíme neprázdný rozdíl intervalů A a B .

Tato situace nastává například jedná-li se o parabolu, jejíž složka má parametrizaci ve tvaru $x = u^2$, v situaci, kdy se vrchol paraboly nachází pod spodní hranicí ořezávané oblasti.



- (c) Obdržíme-li pro $x = xMin$ dvě hodnoty a, b a pro $x = xMax$ méně než dvě hodnoty nebo naopak, pak je výsledný interval hodnot parametru u roven intervalu (a, b) .

Tato situace nastává například jedná-li se o parabolu, jejíž složka má parametrizaci ve tvaru $x = u^2$, v situaci, kdy se vrchol paraboly nachází uvnitř ořezávané oblasti nebo se dotýká hrany ořezávací oblasti.



- (d) Neobdržíme-li žádnou hodnotu pro $x = xMin$ ani pro $x = xMax$, potom je buď celá složka uvnitř nebo mimo ořezávanou oblast. Rozhodnout mezi těmito situacemi je triviální.

2. Mějme parametrizaci složky závislou na dvou parametrech u a v , přičemž tato složka je v jednom z následujících tvarů

$$x = X(u) \cdot \sin(v), \quad x \in (xMin, xMax)$$

nebo ve tvaru

$$x = X(u) \cdot \cos(v), \quad x \in (xMin, xMax).$$

Protože jsou v parametrizacích objektů funkce sinus a cosinus závislé pouze na parametru v a protože jsou tyto funkce periodické, stačí pro kompletní vykreslení objektu vzít za interval hodnot parametru v interval $(0, 2\pi)$ a postupovat stejně jako by byla parametrizace ve tvaru

$$x = X(u), \quad x \in (xMin, xMax).$$

Takto získáme hodnoty parametru u . Pro zvýšení efektivity výpočtu můžeme následně pro každou hodnotu parametru u určit intervaly parametru v přesně.

3. Poslední případ, kdy je parametrizace složky objektu závislá na dvou parametrech, nastává pouze v situaci, kdy je objektem hyperbolický nebo eliptický paraboloid. Tyto objekty mají parametrizace tvaru

$$\begin{aligned}x &= u, & x &\in (xMin, xMax) \\y &= v, & y &\in (yMin, yMax) \\z &= a \cdot u^2 \pm b \cdot v^2, & z &\in (zMin, zMax)\end{aligned}$$

Proto abychom mohli vypočítat hranice parametrů u a v pro složku z , musíme využít jedné z ostatních složek parametrizace. Bez újmy na obecnosti za tuto parametrizaci zvolíme parametrizaci složky y . Tato složka nám omezuje hodnoty, kterých může parametr v nabývat na interval $(yMin, yMax)$. Poté pro každou hodnotu parametru v určíme intervaly parametru u dosazením konkrétní hodnoty v do složky z .

2.6. Ořezávání křivky průniku dvou kvadrik

Křivka průniku dvou kvadrik je dána svými parametrickými vyjádřeními složek x, y, z a definičním oborem. Můžeme předpokládat, že parametrizace každé složky obsahuje maximálně jeden parametr a tento je pro všechny složky stejný. Bez újmy na obecnosti můžeme předpokládat, že je to parametr u . Pro úplnost dodejme, že může nastat situace, kdy parametrizace nějaké složky obsahuje dva parametry, ale v tom případě je průnikem kvadrik simple ruled kvadrika.

Hledané hodnoty parametru u určíme jako průnik hodnot parametru u , pro které leží jednotlivé složky křivky v odpovídajících intervalech. Hledání intervalů parametru u budeme provádět v kanonické bázi simple ruled kvadriky, protože v této bázi je parametrizace křivky průniku nejjednodušší. Jen je potřeba transformovat do této báze i jednotlivé ořezávací intervaly.

1. Mějme parametrizaci složky x ve tvaru

$$x = f(u), \tag{2.15}$$

kde $f(u)$ je maximálně kvartická funkce neznámé u . Dále mějme interval $I = (min, max)$ tvořící zvolené hranice složky x . Pak rovnici (2.15) upravíme na tvar $f(u) - x = 0$ a vzniklou rovnici vyřešíme pro hodnoty x rovny min a max . Kořeny této rovnice setřídíme vzestupně a z každých dvou po sobě jdoucích kořenů vytvoříme interval K_i , u kterého zkoumáme zda na něm složka x leží uvnitř intervalu I . To provedeme tak, že z intervalu K_i vybereme prostřední prvek a dosadíme do parametrizace složky x .

Leží-li funkční hodnota $f(u)$ v tomto bodě v intervalu I , pak to platí i pro celý interval K_i .

Nyní popíšeme postup při řešení rovnice $f(u) - x = 0$, pro případy, kdy tato není ve tvaru rovnice prvního až čtvrtého stupně.

2. Je-li parametrizace složky x ve tvaru

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

pak tuto rovnici postupně upravíme

$$\begin{aligned} x &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \\ x^2 \cdot a + x \cdot b + c &= 0. \end{aligned}$$

Tato rovnice je vždy maximálně kvartická, což vychází z parametrizací simple ruled kvadrik. Nyní postupujeme stejně jako v bodě 1.

3. Je-li parametrizace složky x ve tvaru

$$x = du + d \cdot \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

nebo ve tvaru

$$x = du - d \cdot \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

pak tuto rovnici postupně upravíme na tvar

$$\frac{(x - du)^2 \cdot a}{d^2} + \frac{(x - du) \cdot b}{d} + c = 0, \quad (2.16)$$

resp. na tvar

$$\frac{(x - du)^2 \cdot a}{d^2} - \frac{(x - du) \cdot b}{d} + c = 0. \quad (2.17)$$

Tyto situace nastávají pouze v případě, kdy je simple ruled kvadrikou hyperbolický paraboloid. Díky zvoleným parametrizacím hyperbolického paraboloidu jsou výsledné rovnice (2.16) a (2.17) maximálně kvartické a dále se postupuje stejně jako v bodě 1.

4. Je-li parametrizace složky x ve tvaru

$$x = -p \cdot u \cdot \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

pak tuto rovnici postupně upravíme na tvar

$$x^2 \cdot a - x \cdot bpu + cp^2u^2 = 0. \quad (2.18)$$

Podobně jako v předchozím případě nastává tato situace pouze pokud je simple ruled kvadrika hyperbolický paraboloid. A opět je vzniklá rovnice maximálně kvartická a dále se postupuje stejně jako v bodě 1.

3. Programátorská příručka

Pro vytvoření programu byl zvolen programovací jazyk *C#* a jeho vývojové prostředí *Microsoft Visual Studio 2010*. Pro vykreslování scény byla použita grafická knihovna *OpenGL*, kterou pro *C#* zpřístupňuje například volně dostupná knihovna *Tao framework*.

Výsledný program je rozdělen do několika logických celků. V následujících podkapitolách uvedeme stručný popis jednotlivých celků.

3.1. Matematika

Většina matematického aparátu je umístěna ve jmenném prostoru *Gmath*. Mezi nejpoužívanější zástupce z této skupiny tříd patří

3.1.1. Vector

Třída reprezentující vektor trojrozměrného afinního prostoru. Obsahuje metody implementující základní algebraické operace s vektory jako jsou součet, rozdíl a součin vektorů, násobení vektoru skalárem, délka vektoru a úhel vektorů.

3.1.2. Matrix

Třída reprezentující matici libovolného řádu. Tato třída obsahuje metody implementující základní algebraické operace s maticemi jako jsou součet a součin matic nebo výpočet determinantu a inverzní matice. Od této třídy dědí různé speciální typy matic jako jsou transformační matice, matice kvadriky a matice kuželosečky. Tyto třídy obsahují implementaci dalších potřebných operací. Mezi nejzajímavější patří převod eulerovských úhlů na transformační matici a naopak nebo aplikace transformační matice na matici kvadriky či kuželosečky.

3.1.3. Interval

Třída reprezentující jednorozměrný interval hodnot. Tato třída je hojně využívaná při výpočtech ořezávání objektů. Obsahuje metody implementující operace s intervaly jako jsou sjednocení, průnik a doplněk intervalů, délka intervalu nebo třídění intervalů.

3.1.4. Algebraic Equations

Algebraic Equations je skupina tříd reprezentujících algebraické rovnice až do čtvrtého řádu. Lineární a kvadratické rovnice jsou řešeny pomocí známých vzorců. Kubické a kvartické rovnice jsou řešeny pomocí Cardanových vzorců.

Praktickým testováním bylo zjištěno, že je velmi obtížné jednoznačně rozhodnout, jakého stupně je daná rovnice. Proto je vstupní rovnice až n -tého stupně vyřešena jako rovnice stupně n i $n - 1$. Poté je jako správné řešení původní rovnice vzato to z řešení, které má po substituci vybraných hodnot, nejmenší součet rozdílů funkčních hodnot oproti původní rovnici. Dalším problémem při řešení těchto rovnic jsou zaokrouhlovací chyby FPU aritmetiky. Tyto chyby sice způsobují jen mírné nepřesnosti ve vypočítaných kořenech rovnice, avšak v jistých situacích může i nepatrná odchylka způsobit velký skok funkční hodnoty. Z těchto důvodů je proto občas možno pozorovat mírné chyby ve vykreslovaných průnicích.

Mimo metody pro vyřešení dané rovnice, obsahují tyto třídy i metody pro substituci hodnoty do dané rovnice, či metody implementující součet a součin těchto rovnic.

3.1.5. EquationSystem

Třída *EquationSystem* reprezentuje soustavu lineárních rovnic. Jednotlivé lineární rovnice jsou reprezentovány třídou *Equation*, která obsahuje implementaci metod pro sčítání a odčítání jednotlivých rovnic, či násobení rovnice skalárem. Soustavy rovnic jsou řešeny Gausovou eliminací.

3.1.6. CommonEquation

Jako předek všech obecných rovnic grafických objektů vznikla třída *CommonEquation*, která obsahuje slot nesoucí seznam všech koeficientů konkrétní rovnice. Mezi nejzajímavější metody této třídy patří metoda *Equal* porovnávající dvě rovnice a metoda *Normalize*, která provede vynásobení koeficientů takovou mocninou deseti, aby byl součet exponentů v rovnici co nejbližší nule. Od této třídy následně dědí třídy reprezentující obecné rovnice jednotlivých grafických objektů. Jsou to třídy

QuadraticEquation – reprezentuje obecnou rovnici kvadriky,

ConicEquation – reprezentuje obecnou rovnici kuželosečky,

PlanEquation – reprezentuje obecnou rovnici roviny.

Tyto třídy implementují metody pro získání konkrétního koeficientu dané rovnice.

3.1.7. ParametricEquation

Pro reprezentaci parametrických rovnic grafických objektů vznikla třída *ParametricEquation*. Tato třída opět obsahuje slot nesoucí seznam koeficientů konkrétních parametrických rovnic. Od této třídy dědí třídy reprezentující parametrické rovnice jednotlivých grafických objektů. Jsou to třídy

QuadricParametricEquation – reprezentuje parametrické rovnice jednotlivých kvadrik,

ConicParametricEquation – reprezentuje parametrické rovnice jednotlivých kuželoseček,

PlanParametricEquation – reprezentuje parametrické rovnice roviny,

LineParametricEquation – reprezentuje parametrické rovnice přímky,

PointParametricEquation – reprezentuje parametrické rovnice bodu.

Tyto třídy implementují metody pro získání konkrétního koeficientu dané rovnice.

3.2. Grafické objekty

3.2.1. GraphicObject

Tato třída reprezentuje předka všech grafických objektů. Obsahuje sloty

_ID – identifikátor grafického objektu,

_displayListIdentificator – číselný identifikátor displayListu objektu,

_visible – booleovská hodnota udávající, zda je objekt vykreslován či nikoli,

_color – instance třídy *Color* nesoucí informaci o barvě objektu,

_name – textový řetězec pojmenovávající objekt,

_wordDescriptionFirstCase – slovní popis objektu v prvním pádu,

_wordDescriptionSecondCase – slovní popis objektu v druhém pádu,

_wordDescriptionFourthCase – slovní popis objektu ve čtvrtém pádu,

_isInScene – booleovská hodnota udávající, zda je objekt ve scéně,

_isDeleted – booleovská hodnota udávající, zda je objekt smazán.

Kromě getterů a setterů některých slotů tato třída obsahuje i virtuální metody

BuildDisplayList – tato metoda vytvoří displayList objektu,

Draw – vykreslí grafický objekt,

AreEqual – vrací booleovskou hodnotu udávající, zda jsou dva objekty stejné.

3.2.2. WithParametricEquationObject

Potomek třídy *GraphicObject*, představující předka objektů, reprezentovatelných parametrickými rovnicemi. Obsahuje sloty

`_parametricEquation` – instance třídy *ParametricEquation*, představující parametrické rovnice objektu,

`_parametricEquationsFormatString` – instance třídy *ParametricEquationsFormatString*, představující předpis pro vytvoření grafické reprezentace parametrických rovnic,

`_transformationAlgebraic` – instance třídy *TransformationMatrix*, představující transformační matici vzhledem ke standardní bázi, jež je aplikována na objekt,

`_transformationOpenGL` – instance třídy *TransformationMatrix*, jež je používána při vykreslování grafického objektu,

`_axesDisplayListIdentificator` – číselný identifikátor displayListu os objektu,

`_axesVisible` – booleovská hodnota udávající, zda jsou osy objektu vykreslovány,

`_lastBuildPosition` – instance třídy *Vector* nesoucí informaci o pozici objektu, během posledního utváření jeho displayListu.

Potomkům třídy *WithParametricEquationObject* je možno měnit jejich transformační matice. K tomuto účelu slouží metody

SetTransformationMatrixAlgebraic – nastaví objektu novou transformaci vzhledem ke standardní bázi,

SetCenterInStandardBasis – nastaví objektu nové posunutí vzhledem ke standardní bázi,

ApplyRotationInStandardBasis – současnou transformaci objektu složí s předanou rotací.

Tato třída dále definuje virtuální metodu *MakeCopyOfObject*, která vytvoří kopii volajícího objektu s novými koeficienty parametrických rovnic a novými souřadnicemi středu. Třída *WithParametricEquationObject* obsahuje implementaci virtuální metody *Draw* jejího předka *GraphicObject*. Od této třídy dědí

WithCommonEquationObject – třída tvořící základ objektů s obecnou rovnicí,

Point – třída reprezentující bod,

Line – třída reprezentující přímku.

3.2.3. WithCommonEquationObject

Potomek třídy *WithParametricEquationObject*, reprezentující předka objektů, které mají obecnou rovnici. Tato třída obsahuje sloty

`_commonEquation` – instance třídy *CommonEquation*, představující obecnou rovnici objektu.

`_commonEquationFormatString` – instance třídy *CommonEquationFormatString*, představující předpis pro vytvoření grafické reprezentace obecné rovnice.

Tato třída definuje virtuální metodu *MakeCopyOfObject*, která vytvoří kopii volajícího objektu s novými koeficienty obecné rovnice. Dále tato třída definuje virtuální metodu *ActualizeEquation*, která aktualizuje obecnou rovnici objektu. Od této třídy dědí

WithCanonicEquationObject – třída tvořící základ objektů s kanonickým tvarem rovnice,

Plan – třída reprezentující rovinu

3.2.4. WithCanonicEquationObject

Potomek třídy *WithCommonEquationObject*, reprezentující předka objektů, které mají kanonický tvar obecné rovnice. Tato třída obsahuje sloty

`_canonicEquation` – instance třídy *CommonEquation* představující kanonický tvar obecné rovnice objektu,

`_canonicEquationFormatString` – instance třídy *CanonicEquationFormatString*, představující předpis pro vytvoření grafické reprezentace kanonického tvaru obecné rovnice.

Obsahuje metody pro transformaci souřadnic mezi standardní bází a polární bází objektu. Od této třídy dědí

Quadric – třída představující předka všech kvadrik, od této třídy dědí jednotlivé typy kvadrik,

ConicSection – třída představující předka všech kuželoseček, od této třídy dědí jednotlivé typy kuželoseček.

3.2.5. ConicSection

Kuželosečky jsou na rozdíl od ostatních podporovaných geometrických objektů definované pouze v rovině. Proto je nutno pro každou kuželosečku definovat i rovinu v níž je daná kuželosečka definovaná. Jako výchozí rovina je pro každou kuželosečku stanovena rovina R o rovnici $R : z = 0$. Tuto rovinu je možno v detailu kuželosečky editovat. Za z -tovou osu kuželosečky považujeme normálový vektor roviny R . Při editaci kuželosečky nebo její roviny se výsledná transformační matice rozloží na transformační matici kuželosečky a transformační matici roviny. Tyto transformační matice slouží k výpočtu obecné rovnice kuželosečky a rovnic roviny. Souřadnice středu a úhly rotace v detailu kuželosečky se vztahují ke složení těchto dvou transformačních matic.

3.3. Manipulace s objekty

3.3.1. ManipulationManager

Tato třída zprostředkovává všechny manipulace s objekty. Stisknutí tlačítka editovat nebo vybrání položky v kontextovém menu se přepoše ManipulationManageru, který vytvoří instanci konkrétního editačního dialogu. Tento dialog přidá do seznamu otevřených dialogů a zobrazí jej. V případě uzavření dialogu je o tomto ManipulationManager informován a zařídí všechny potřebné úkony spojené s danou akcí. Mezi tyto úkony patří mimo jiné i aktualizace obsahu otevřeného detailu ovlivněného objektu. Dialog je následně smazán ze seznamu dialogů. Tato třída obsahuje sloty

_graphicObjects – slot nesoucí seznam všech grafických objektů,

_graphicObjectDetails – slot nesoucí seznam všech otevřených detailů objektů,

_graphicObjectDialogs – slot nesoucí seznam všech otevřených editačních dialogů,

_history – slot nesoucí historii,

_addManager – slot nesoucí instanci třídy *AddManager* starající se o přidávání nových grafických objektů.

3.3.2. Editační dialogy

Editací dialogy jsou potomky třídy *IDialog*. Tento předek obsahuje slot nesoucí editovaný objekt a slot nesoucí odkaz na ManipulationManager. Všichni potomci této třídy pracují podobným způsobem. Při vytvoření nastaví dialog

hodnoty editované vlastnosti do funkčních prvků, zároveň si tyto hodnoty zapamatuje jako inicializační hodnoty. Pokud dojde k potvrzení stiskem tlačítka OK, vyparsují se nové hodnoty z funkčních prvků a uloží se do slotu jako aktuální hodnoty. Pokud jsou nové hodnoty různé od inicializačních, nastaví se DialogResult dialogu na OK, jinak zůstane nastaven na hodnotu NONE. Během parsování může dojít k selhání, o kterém je uživatel informován a proces reakce na stisk tlačítka OK je přerušen. Uzavření dialogu je následně přeposláno Manipulation-Manageru, který se dále rozhoduje podle hodnoty slotu DialogResult.

3.4. Vytvoření grafické reprezentace rovnice

Grafickou reprezentaci jednoho řádku rovnice vytváří třída *EquationLineControl*. Tato třída po obdržení formátovacího řetězce vytvoří jednotlivé členy rovnice zakódované ve formátovacím řetězci a vloží je do svého slotu Controls.

3.4.1. Formátovací řetězec

Formátovací řetězec je zřetěžením dílčích formátovacích řetězců popisujících jednotlivé členy rovnice. Dílčí formátovací řetězec obsahuje na prvním místě znak určující o jaký typ členu rovnice se jedná. Dále za tímto znakem následují uvedené argumenty daného členu uzavřené do hranatých závorek, oddělené středníkem. Přičemž je možné, aby byl argument členu rovnice opět členem rovnice.

3.4.2. MemberControl

Tato třída je potomkem třídy *Control*, tvořícím společného předka controlům, které představují grafickou reprezentaci nějakého prvku rovnice. MemberControl definuje dvě virtuální metody *DrawToToolTip* a *GetGraphicsSize* umožňující vykreslování potomků třídy MemberControl do toolTipu. Všichni potomci třídy MemberControl tyto virtuální metody přepisují.

3.4.3. LabelControl

Tento potomek třídy *MemberControl* reprezentuje mnohočlen. Jediným slotem této třídy je instance třídy *Label*, v níž je zobrazován text mnohočlenu.

3.4.4. FracControl

Tento potomek třídy *MemberControl* reprezentuje zlomek. Obsahuje sloty *_numerator* a *_denominator*, což jsou instance typu *MemberControl*, v nichž jsou uloženi čitatel a jmenovatel zlomku. Zlomková čára je reprezentována pomocí instance třídy *PictureBox*.

3.4.5. SqrtControl

Tento potomek třídy *MemberControl* reprezentuje odmocninu. Jediným slotem této třídy je instance třídy *MemberControl*, nesoucí odkaz na výraz pod odmocninou. Znak odmocniny je zobrazován v instanci třídy *Label* a odmocninová čára je reprezentována pomocí instance třídy *PictureBox*.

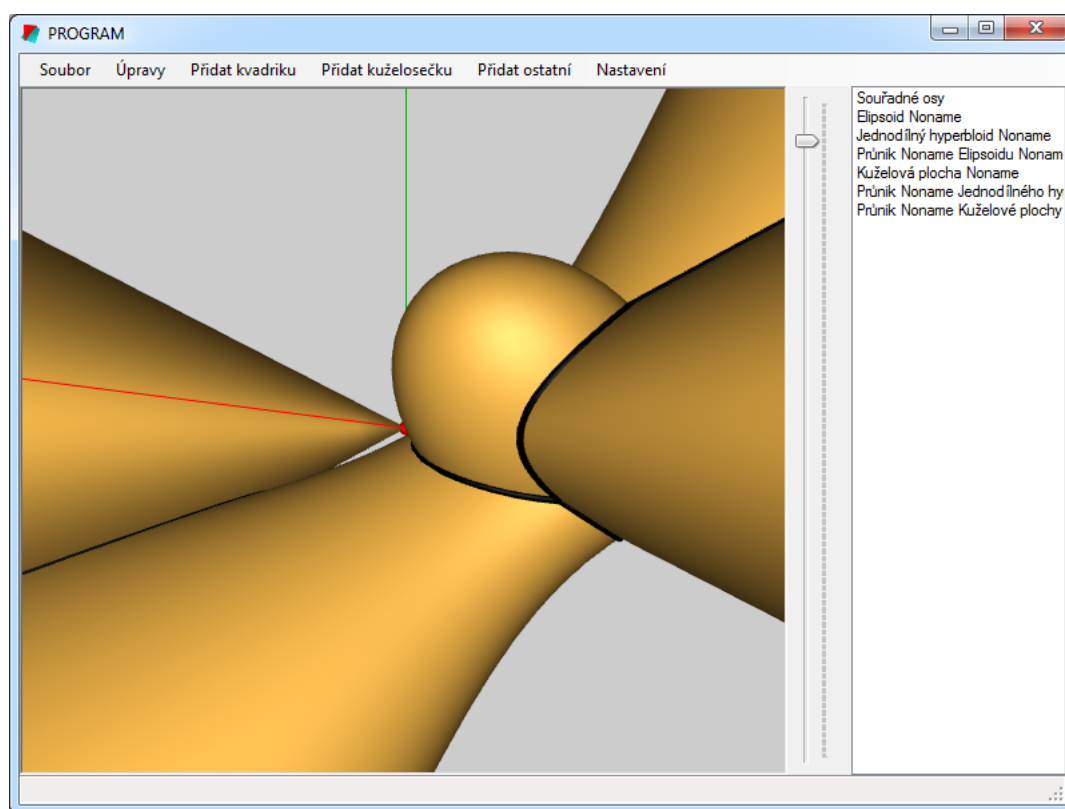
3.4.6. CombinedControl

Tento potomek třídy *MemberControl* reprezentuje výraz složený z více *MemberControl*ů zapsaných za sebou. Jeho jediným slotem je seznam *MemberControl*ů. Tyto *MemberControl*y jsou *CombinedControl*em zarovnány na osu pomyslného řádku.

4. Uživatelská příručka

4.1. Hlavní okno programu

Hlavní okno programu je rozděleno na dvě části. V levé části okna se nachází oblast do níž je vykreslován obsah scény. V pravé části okna se nachází posuvník přiblížení a výčet geometrických objektů nacházejících se aktuálně ve scéně. Hranice mezi těmito částmi okna je nastavitelná uchopením a tažením myši. V horní části okna se nachází standardní menu.



Obrázek 1. Hlavní okno programu

4.2. Manipulace se scénou

Scénu je možno tažením myši při stisknutí levém tlačítku myši posunovat horizontálně i vertikálně. Posunutou scénu lze vrátit do výchozí pozice klepnutím na položku menu *Úpravy/Vycentrovat* scénu nebo stiskem kombinace kláves *Ctrl+Space*. Tažením myši při stisknutí pravém tlačítku myši dochází k rotaci scény okolo středu obrazovky. Tažením myši ve vertikálním směru při stisknutí

středním tlačítku myši dochází k přibližování a oddalování scény. Přiblížení či oddálení scény je možno realizovat i rotací kolečka myši nebo pomocí posuvníku, který se nachází na pravém okraji vykreslovací oblasti.

4.3. Kontextové menu

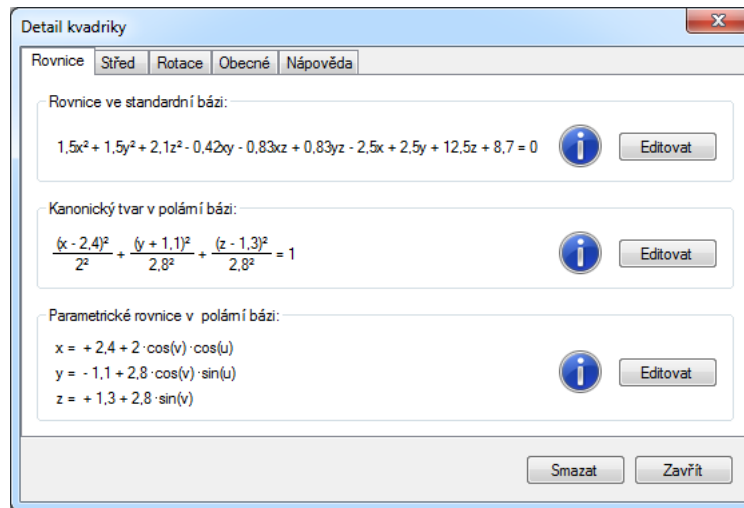
Prvek zobrazující seznam všech objektů nacházejících se ve scéně, umožňuje zobrazit kontextové menu s jehož pomocí lze rychle provádět vybrané operace nad objekty. Mezi tyto operace patří:

1. Skrýt výběr – skryje všechny označené objekty,
2. Zobrazit výběr – zobrazí všechny označené objekty,
3. Smazat výběr – smaže všechny označené objekty,
4. Přejmenovat výběr – zobrazí dialog editace jména, změnu aplikuje označenému objektu,
5. Přebarvit výběr – zobrazí dialog editace barvy, změnu aplikuje na všechny označené objekty,
6. Přidat průnik výběru – zobrazí dialog pro přidání průniku dvou označených objektů,
7. Skrýt osy výběru – skryje osy všech označených objektů,
8. Zobrazit osy výběru – zobrazí osy všech označených objektů.

Pro označení položky v seznamu objektů klepneme na položku. Pro označení další položky stiskneme *Ctrl* a klepneme na položku. Zobrazení kontextového menu provedeme klepnutím pravého tlačítka myši nad jednou z označených položek. Není-li daná akce kontextového menu aplikovatelná na počet nebo typ označených objektů, je tato akce nedostupná.

4.4. Přidání geometrického objektu

Přidání geometrického objektu do scény provedeme klepnutím na příslušnou položku standardního menu. Možností pro přidání průniku dvou geometrických objektů je i výše zmíněné kontextové menu.



Obrázek 2. Detail grafického objektu

4.5. Editace geometrických objektů

Pro editaci a prohlížení vlastností objektů slouží dialog, který se zobrazí po poklepnutí na editovaný objekt v seznamu objektů. Tento dialog umožňuje, aby mohl uživatel současně manipulovat s tímto dialogem a zároveň i s hlavním oknem programu. Tento dialog zobrazuje na několika stránkách všechny editovatelné vlastnosti objektu. Obsah tohoto dialogu závisí na typu objektu, jehož vlastnosti chceme prohlížet nebo editovat. Tento dialog může obsahovat:

1. stránku zobrazující rovnice objektu,
2. stránku zobrazující vlastnosti roviny v níž je objekt definován,
3. stránku zobrazující souřadnice středu objektu,
4. stránku zobrazující informace o rotaci objektu,
5. stránku zobrazující jméno, barvu, viditelnost objektu a viditelnost os objektu.

Většinu zobrazených vlastností je možno buď přímo nebo otevřením editačního dialogu dané vlastnosti editovat. Editační dialog dané vlastnosti se zobrazí po klepnutí na tlačítko přidružené k dané vlastnosti. Editační dialogy vlastností umožňují, aby mohl uživatel současně manipulovat s těmito dialogy i s hlavním oknem programu. V případě, že dojde k editaci takové vlastnosti objektu, která změní jeho tvar nebo pozici, jsou případné průniky s tímto objektem přepočítány.

Dialog upravující rotaci objektu se jako jediný odlišuje od ostatních dialogů tím, že se každá změna rotace ihned projeví na editovaném objektu, ale přitom informace o této akci není až do klepnutí na tlačítko OK do historie uložena. Navíc změna provedená v tomto editačním dialogu může ovlivnit obsah jiného dialogu. Kvůli složité synchronizaci těchto změn je v jednu chvíli povoleno editovat pouze jednu vlastnost ovlivňující tvar nebo pozici objektu. Toto je zajištěno zakázáním daných tlačítek v okně detailu otevírajících editační dialog. Ze stejného důvodu jsou zakázány akce *vpřed* a *zpět* v čase, kdy je otevřen libovolný dialog editace rotace.

4.6. Historie

Program si pamatuje historii provedených akcí a umožňuje uživateli se v historii pohybovat. Do historie se zapisují pouze akce manipulující s grafickými objekty. Posun v historii o krok zpět provedeme klepnutím na položku *Úpravy/Zpět* ve standardním menu nebo stiskem kombinace kláves *Ctrl+Z*. Pro posun o krok vpřed klepneme na položku *Úpravy/Vpřed* ve standardním menu nebo stiskneme kombinaci kláves *Ctrl+Y*.

4.7. Uložení

Uložení obsahu scény provedeme klepnutím na položku *Soubor/Uložit* ve standardním menu nebo stiskem kombinace kláves *Ctrl+S*. Následně se zobrazí standardní dialog pro uložení souboru, ve kterém vybereme umístění ukládaného souboru a specifikujeme jeho název.

4.8. Načtení

Načtení uložené scény provedeme klepnutím na položku *Soubor/Načíst* ve standardním menu nebo stiskem kombinace kláves *Ctrl+O*. Poté se program zeptá na potvrzení této volby a nabídne uložení stávající scény. Dojde-li k chybě během ukládání, je akce načítání stornována. Po úspěšném uložení nebo po přeskočení uložení se zobrazí standardní dialog pro otevření souboru. Dojde-li k chybě během načítání nové scény, zůstane aktivní stávající scéna.

4.9. Nová scéna

Resetování scény provedeme klepnutím na položku *Soubor/Nový* ve standardním menu nebo stiskem kombinace kláves *Ctrl+N*. Poté se program zeptá na potvrzení této volby a nabídne uložení stávající scény. Dojde-li k chybě během ukládání, je akce resetování stornována.

4.10. Export do obrázku

Export do obrázku provedeme klepnutím na položku *Soubor/Export do obrázku* ve standardním menu nebo stiskem kombinace kláves *Ctrl+E*. Následně se zobrazí standardní dialog pro uložení souboru, ve kterém vybereme umístění ukládaného souboru a specifikujeme jeho název.

4.11. Nastavení

Uživateli je umožněno změnit několik vybraných grafických vlastností, kterými jsou

1. Kvalita objektů – v zobrazeném dialogu lze nastavit přesnost aproximace křivek a ploch,
2. Barva pozadí – v zobrazeném dialogu lze nastavit barvu pozadí scény,
3. Vyplnit plochy – po klepnutí na tuto položku bude trojúhelníková síť ploch vyplněna barvou,
4. Trojúhelníková síť – po klepnutí na tuto položku budou plochy kresleny jako trojúhelníkové sítě,
5. Antialiasing – klepnutím na tuto položku se zobrazí následující kategorie menu, ve které lze klepnutím nastavit úroveň antialiasingu. Nepodporované úrovně antialiasingu jsou nedostupné.

4.12. Uživatelská nápověda

Zobrazení nápovědy provedeme klepnutím na položku *Nápověda/Zobrazit nápovědu* ve standardním menu nebo stisknutím kombinace kláves *Ctrl+H*.

4.13. Ukončení programu

Ukončení programu provedeme uzavřením hlavního okna programu libovolným ze standardních způsobů. Jimi jsou klepnutí na uzavírací tlačítko v pravém horním rohu okna, stisk kombinace kláves *Alt+F4* nebo klepnutí na položku *Soubor/Ukončit* ve standardním menu. Poté se program zeptá na potvrzení této volby a nabídne uložení stávající scény. Dojde-li k chybě během ukládání, je akce ukončení stornována.

Závěr

Výsledkem práce je softwarový nástroj GEOMETRIE. Tento nástroj umožňuje zobrazovat vybrané geometrické objekty, kterými jsou body, přímky, roviny, kuželosečky a kvadriky. Každý objekt může být zadán specifikováním koeficientů v různých jeho rovnicích. Kvadriky a kuželosečky lze zadat pomocí jejich obecné rovnice a nástroj provede jejich převedení do kanonického tvaru a rozpoznání typu. Jednotlivé rovnice přidaného objektu je možno editovat stejně tak jako jeho posunutí, rotaci a různé grafické vlastnosti. Každým dvěma objektům je nástroj schopen vypočítat vyjádření průniku a průnik graficky zobrazit. Je umožněno nastavení čtyř úrovní kvality, které ovlivňují přesnost aproximace zobrazených objektů. Pro co nejlepší kvalitu zobrazení je také umožněno měnit úroveň anti-aliasingu, tuto funkcionalitu však musí podporovat grafická karta a její ovladač. Samozřejmostí je historie provedených akcí a možnost se v ní pohybovat.

Reference

- [1] Michal Krupka. *Geometrie pro informatiky*. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, 2008.
- [2] Josef Janyška, Anna Sekaninová. *Analytická teorie kuželoseček a kvadrik*. Masarykova univerzita, Přírodovědecká fakulta, Brno, 2001.
- [3] Roman Hašek, Pavel Pech. *Kvadratické Plochy a jejich reprezentace v programu Maple*. Jihočeská univerzita v Českých Budějovicích, 2010.
- [4] Pavel Pech. *Kuželosečky*. Jihočeská univerzita v Českých Budějovicích, 2004.
- [5] Daniel Lazard, Laurent Dupont, Sylvain Lazard and Sylvain Petitjean. *Near-optimal parameterization of the intersection of quadrics: I. The generic algorithm*. Journal of Symbolic Computation, Volume 43, Issue 3, March 2008, Pages 168–191.
- [6] Zhi-qiang Xu, Xiaoshen Wang, Xiao-diao Chen, Jia-guang Sun. *A Robust Algorithm for Finding the Real Intersections of Three Quadric Surfaces*. Computer Aided Geometric Design, Volume 22, Issue 6, September 2005, Pages 515–530.
- [7] Wenping Wang, Ronald Goldman, Changhe Tu. *Enhancing Levin’s method for computing quadric-surface intersections*. Computer Aided Geometric Design, Volume 20, Issue 7, October 2003, Pages 401–422.
- [8] Joshua Zev Levin. *Mathematical models for determining the intersections of quadric surfaces*. Computer Graphics and Image Processing, Volume 11, Issue 1, September 1979, Pages 73–87.

A. Obsah přiloženého CD

V samotném závěru práce je uveden stručný popis obsahu přiloženého CD.

bin/

Ve složce bin se nachází instalátor, kterým lze vytvořený program nainstalovat. Pro správné spuštění programu je nutné mít nainstalován i Microsoft .NET Framework 4 Client Profile. Má-li cílový počítač přístup k internetu, je možno chybějící .NET Framework nainstalovat automaticky během instalace programu.

doc/

Složka doc obsahuje text práce ve formátu PDF, vytvořený dle závazného stylu KI PřF pro diplomové práce. Součástí této složky je i ZIP archiv obsahující všechny soubory nutné pro bezproblémové vygenerování PDF souboru práce.

src/

Kompletní zdrojové texty programu GEOMETRIE se všemi potřebnými knihovnami a dalšími soubory pro bezproblémové vytvoření spustitelné verze programu se nachází ve složce src.