

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Metody zobrazování iso-ploch a neskalárních veličin v E3.

Plzeň, 2002

Petr Kahler

Děkuji rodině a přátelům za duševní a finanční pomoc během studia.

Prohlašuji, že jsem svou diplomovou práci vypracoval výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Plzni dne 17.5.2002

Petr Kahler

Abstract

The name of the thesis is:

The visualization techniques for the iso-surfaces and non scalar data in E3.

This document presents the collection of visualization techniques for different data from the industry or another source.

The first part describes about the algorithm that is responsible for the computation and the conversion of the 2D triangle mesh to the 3D surface whereby rotation around the selected axis. This algorithm searches for the isolines in the mesh. The second part of the algorithm links the isolines to connected sections. In the next part, the found connected sections of isolines are rotated around the selected axis.

The final chapters of the thesis talk about using of the visualization techniques for the testing data and the real data from industry. At the end the results of the computations are illustrated.

Obsah

1	ÚVOD	1
2	ZOBRAZOVÁNÍ VÍCEROZMĚRNÝCH FYZIKÁLNÍCH VELIČIN	2
2.1	PŘÍKLADY NĚKTERÝCH VIZUALIZAČNÍCH METOD	6
2.1.1	<i>Barva</i>	6
2.1.2	<i>Glyfy</i>	7
2.1.3	<i>Kontury / Iso-plochy</i>	8
2.1.4	<i>Pružná vrstva (Rubbersheet)</i>	8
2.1.5	<i>Animace</i>	9
2.1.6	<i>Zobrazování objemů</i>	9
2.1.7	<i>Paralelní souřadnice</i>	10
2.1.8	<i>Rozptýlené body</i>	11
2.1.9	<i>Násobné grafy</i>	12
2.2	APLIKACE ŘEŠÍCÍ PROBLÉM ZOBRAZOVÁNÍ N-D DAT.....	14
2.2.1	<i>Micro AVS 4.0</i>	14
3	VIZUALIZACE POMOCÍ ISOČAR A ROTAČNÍ SYMETRIE	15
3.1	NÁHLED NA PROBLÉM	15
3.2	HLEDÁNÍ ISOČAR VE 2D TROJÚHELNÍKOVÉ SÍTI.....	16
3.3	PROPOJOVÁNÍ ÚSEKŮ ISOČAR	17
3.4	VYTVORENÍ ROTAČNÍHO TĚLESA Z MNOŽINY ISOČAR	20
4	IMPLEMENTACE.....	22
4.1	ZVOLENÉ PROSTŘEDÍ PRO REALIZACI.....	22
4.2	VIZUALIZACE N-D DAT	22
4.3	REALIZACE VIZUALIZACE POMOCÍ ROTAČNÍ SYMETRIE.....	25
4.3.1	<i>Formát dat</i>	25
5	DOSAŽENÉ VÝSLEDKY.....	27
5.1	ČASOVÁ A PAMĚŤOVÁ SLOŽITOST	27
5.1.1	<i>Časová složitost</i>	27
5.1.2	<i>Paměťová složitost</i>	32
5.2	ZOBRAZENÍ VSTUPNÍCH A VÝSTUPNÍCH DAT	33

5.2.1	<i>Testovací množina dat</i>	33
5.2.2	<i>Reálná data - průmysl</i>	37
6	ZÁVĚR A ZHODNOCENÍ	42
6.1	BUDOUCÍ PRÁCE	42
	LITERATURA	43
	PŘÍLOHA A	i
	PŘÍLOHA B	xxii
	PŘÍLOHA C	xxiii
	PŘÍLOHA D	xxv

1 Úvod

V dnešní době se v průmyslu často vyskytuje problém, jak co nejlépe zobrazit data, která obsahují mimo informací o pozici bodu v prostoru také další informace. Tyto informace mohou být například: teplota, tlak, rychlost, akcelerace a další. Získáváme tak problém zobrazování *N-rozměrných* dat (dále jen *N-D dat*) v prostoru. Typickým příkladem tohoto zobrazení je mapa meteorologické předpovědi. Tato data lze reprezentovat různými způsoby zobrazení.

2 Zobrazování vícerozměrných fyzikálních veličin

Zobrazování N-D dat lze rozdělit podle následující tabulky 2.1. Toto rozdělení navrhl Ken Brodli [Bro92]. V tabulce jsou dány Entity dat, které jsou definovány nad d -dimenzionální oblastí a jedná se o (n -) Skalární data. Takto definované zobrazení označujeme $E_d^{(n)S}$.

Tímto způsobem zobrazíme N-D skalární entitu dat na d -dimenzionální oblast. Když se stav entity dat mění v čase, přidá se dolní index t . S každou barvou, s barevnou mapou nebo rozměrem *glyphu* (bude vysvětleno dále) dostaneme další stupeň zobrazení skalárních hodnot. Pomocí tvaru *glyphu* můžeme dále rozšířit stupeň dimenze zobrazených dat.

Systematika	Aplikace	Technika zobrazení
E_1^S	Velké množství (Funkce $y = f(x)$)	2D graf
E_2^S	Meteorologie Aerodynamika/Letecký průmysl Fyzika, astronomie	Kontury Zobrazení rastrem Povrchový pohled Pomocí oblastí (kobercově)
E_2^{nS}	Geografie (Kartografie, data závislá na nadmořské výšce) Fyzika, astronomie Lékařství	Pomocí oblastí s definovanou výškou (Kombinace barev a glyphů)
E_3^S	Dálkové snímání Fyzika, astronomie, chemie Lékařství	Iso-plochy a povrchové zobrazení „Basket weave“ Objemové zobrazení Řezy Ořezávání “Capping“
E_3^{nS}	Fyzika Fyzikální a biologická chemie Návrh léků	Viz. E_3^S (Kombinace barev a glyphů)

	Dokončení částicové analýzy Lékařská věda Archeologické práce – rekonstrukce Technologie naftových nádrží	
$E_{2,t}^{nS}$	Astrofyzika Meteorologie Výpočet kapalinové dynamiky	Viz. E_2^{nS}
$E_{3,t}^{nS}$	Astrofyzika Chemický průmysl Výpočet kapalinové dynamiky Oceánografie	Viz. E_3^{nS}
E_m^{nS}	Fyzika (Dynamické systémy) Počítačová věda (Zobrazení algoritmů)	Projekce do nižší dimenze: 3D-křivky a povrchy “Ribbons“ 2D-kontury

Tabulka 2.1: Rozdělení dle systematiky pro skalární data

Následující tabulka 2.2 je organizována stejným způsobem jako předchozí až na jeden rozdíl, že entity E mají (n) -vektor dat nad d -dimenzionální oblastí. Toto zobrazení značíme $E_d^V_n$ nebo pro $n \times n$ tenzor: $E_d^T_{n;n}$ (kde většinou $n = 3$).

Jako v předchozím případě (skalární data) indikuje index t časovou závislost.

Vizualizace není většinou limitována použitím samostatných skalárních nebo vektorových dat, ale kombinacemi těchto dat. Barva, velikost a tvar glyfů dále opět posunuje dimenzi zobrazovaných dat.

Systematika	Aplikace	Technika zobrazení
$E_2^V_2$	Fyzika Oceánografie	Oblasti s šipkami Sledování částic Časové čáry Pásové čáry
$E_{2,t}^V_2$	Výpočet kapalinové dynamiky	Proudové čáry Proudové polygony Glyfy

E_2^V $E_{2,t}^V$	Fyzika Meteorologie	Viz. E_2^V “Ježci“
E_3^V $E_{3,t}^V$	Fyzika Meteorologie Aerodynamika Výpočet kapalinové dynamiky	Viz. E_2^V “streamribbons“ Proudové plochy Trubice Trsy Texturově založené metody Metody kritického bodu
E_3^T $E_{3,t}^T$	Dokončení částicové analýzy Analýza namáhaných částí	Glyfy Hyper proudové čáry Proudové polygony

Tabulka 2.2: Rozdělení dle systematiky pro vektorová data

Další skupina entit je definována pro d-dimensionální oblast a týká se diskretních (bodových) dat, jinak též nazývaných „rozptýlené body“. Označujeme ji E_d^P . Příkladem je vzdálenost planet od země (E_1^P). Jestliže budeme brát v úvahu hmotnost a poloměr planety je označení E_2^P . Mezi typické techniky patří označování nápisy, barvení, zobrazení pomocí glyfů a čar.

Dále uvedená tabulka 2.3 obsahuje algoritmy pro techniky obsažené v předchozích dvou tabulkách.

Technika	Algoritmy / Popis
2D graf	Interpolace: Čar “Splines“

Kontury	Pochodující čtverce (Marching squares)
Zobrazení rastrem	Zpracování obrazu
Povrchový pohled Zobrazení pomocí oblastí (kobercově)	Plovoucí horizont Odstranění skrytých čar Stínování Osvětlení
Oblasti s definovanou výškou	Kombinace povrchového pohledu Zobrazení rastrem nebo barvou
Iso-plochy a povrchové zobrazení	Pochodující kostky (Marching cubes) „Cuberilles“ (neprůhledné kostky) Rozdělené kostky Propojování kontur
„Basket weave“	Konturové křivky (protínající se sekce) jako silné neprůhledné pruhy Odstranění skrytých ploch
Objemové zobrazení	„Ray casting“ (vs) „Splatting“ Klasifikace a stínování Texturově orientované techniky
Řezy Ořezávání “Capping“	Viz povrchové zobrazení
Glyfy	Geometrická primitiva
Sledování částic	Numerická integrace
Časové čáry	Postupující pozice čar a částic v čase
Pásové čáry	Exploze částic
Proudové čáry	Polyčáry a numerická integrace
„Streamribbons“ Proudové plochy Trubice Trsy	Postupující fronta Polygoniální síť, proudové čáry a základní křivky Mapování textur

Proudové polygony	Deformované polygony (eventuálně s proudovými čarami)
Ježci Oblasti s šipkami	Orientované čáry a primitiva
Texturově založené metody	Místní šum Přímkový integrál Konvoluce
Metody kritického bodu	Body nulové rychlosti
Hyperproudové čáry	Proudové čáry , vlastní pole a geometrická primitiva
Označování nápisy	Pole obsahující text

Tabulka 2.3: Techniky zobrazení

2.1 Příklady některých vizualizačních metod

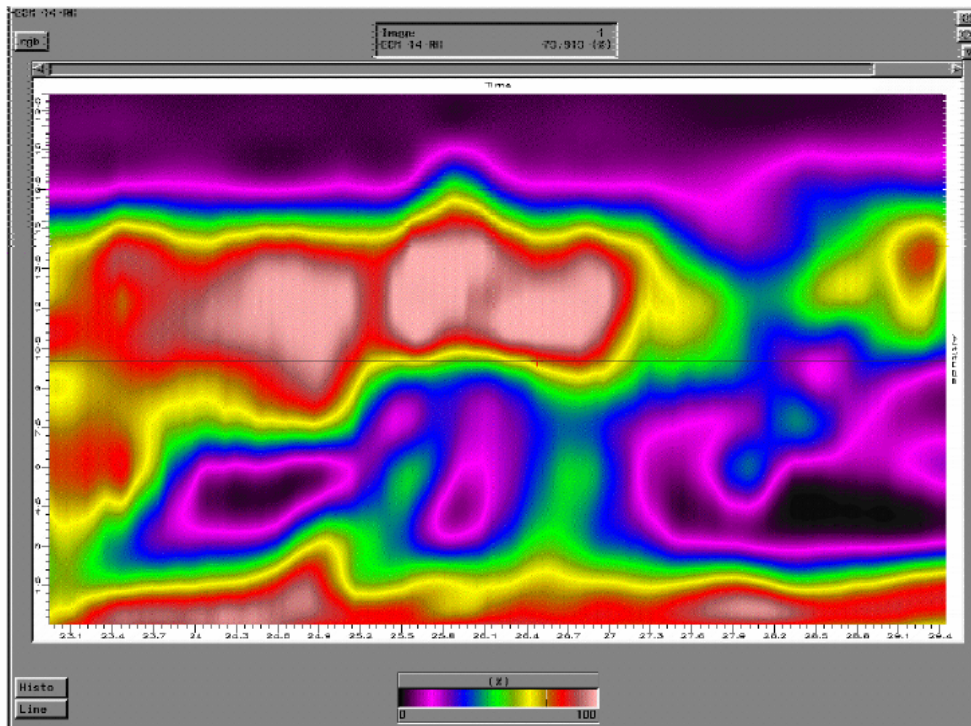
Zde se nacházejí hlavní vizualizační metody pro N-D data. Použití těchto metod závisí na tvaru a dimenzi vstupních dat. Dalším aspektem zobrazení těchto dat je, z jaké oblasti jsou vstupní data.

2.1.1 Barva

Jedná se o vizualizaci skalárních dat; př. vysoká teplota je reprezentována červenou barvou.

Omezení a doporučení podle [DataMin99] jsou následující:

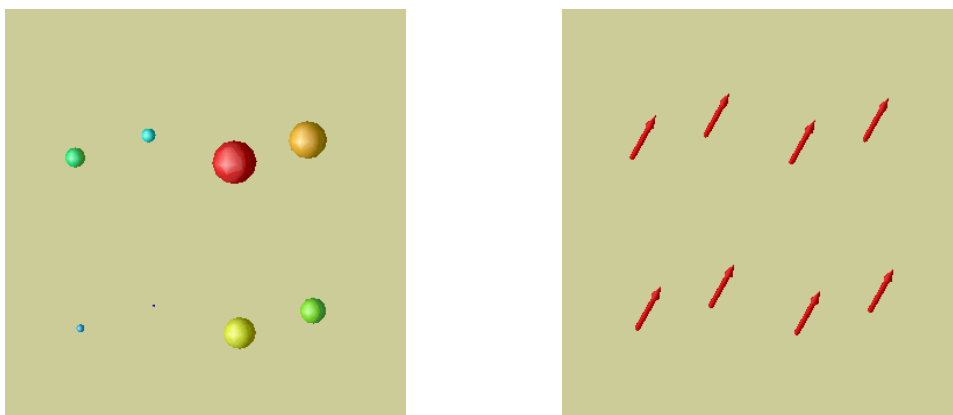
- Barva je jedna z rozhodujících složek pro náhled na data. Wickens (1992) [DataMin99] doporučil limit pěti nebo šesti barev v jednom obraze.
- Barevný odstín nepoužívat jako přírodně řazené kontinuum
- Barevné stereotypy (červená = nebezpečí, modrá = zima) můžou způsobit problémy. Př.: Snížení teploty v mrazícím systému může být nebezpečné.
- Irelevantní barevné kódování může odvádět pozornost



Obrázek 2.1: NASA vizualizace aerosolu rozptýleného v Tichém oceánu
(viz [DataMin99])

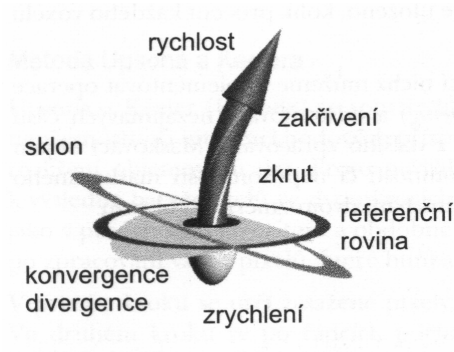
2.1.2 Glyfy

Glyfem rozumíme geometrický objekt, který může reprezentovat lokálně více veličin v diskretních místech. Zde můžeme libovolně dimenzi přiřadit barvu a velikost. Tvar můžeme přiřadit odlišným polím dat, př. koule pro skalární pole a šipky pro vektorové pole, nebo koule pro jedno skalární pole a kostky pro ostatní.



Obrázek 2.2: Glyfy pro skalární hodnoty (vlevo) a glyfy pro vektorové hodnoty
(vpravo) (viz [SV])

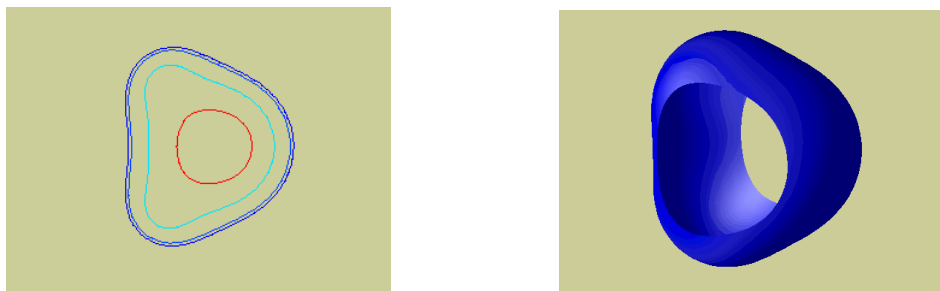
Další náhled na složitější zobrazení pomocí glyfů uvedených v [Zar98] si můžeme udělat z následujícího obrázku 2.3. Takto zobrazený glyf můžeme použít i jako sondu do zajímavých míst v datech, která chceme pozorovat.



Obrázek 2.3: Glyf, jako prostředek vizualizace vektorového pole
(Viz [Zar98])

2.1.3 Kontury / Iso-plochy

Tato vizualizace je vhodná pro zobrazení skalárních dat úrovnových hodnot ve dvou nebo více dimensionálním prostoru. Reprezentace dat jako povrchů v tří dimensionálním prostoru se nazývá renderování povrchů (surface rendering).

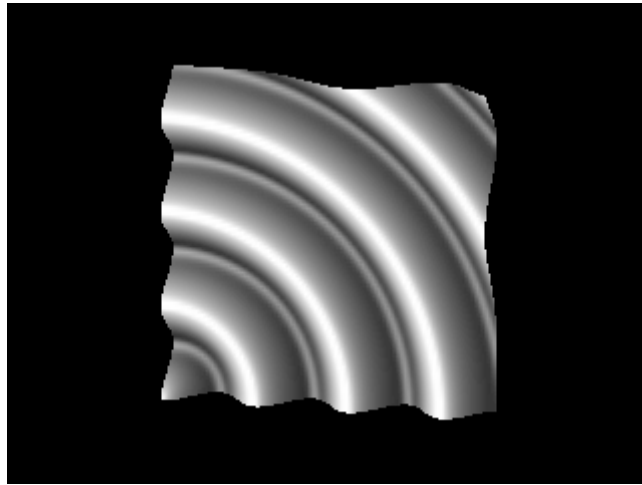


Obrázek 2.4: Isočáry (vlevo) a iso-plochy (vpravo) (viz [SV])

2.1.4 Pružná vrstva (Rubbersheet)

Je vhodná opět pro zobrazování skalárních dat, ale jinak než pomocí barev (ve speciálních případech glyfy) a to jmenovitě pomocí výškově deformovaného povrchu. Při použití tohoto způsobu zobrazení můžeme přidat do této vrstvy další skalární data za použití barvy.

Důležitým aspektem použití pružné vrstvy je přesnost datové reprezentace a také závislost na dopadajícím světle. Stíny indikují výšku (stínování).



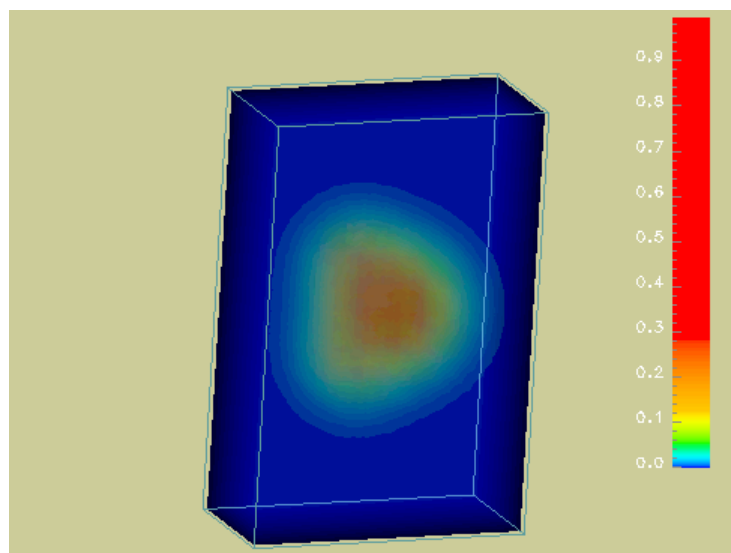
Obrázek 2.5: Pružná vrstva (Rubbersheet) (viz [SV])

2.1.5 Animace

Animace je změna dat (pozice či jiné hodnoty) jako funkce času nebo jiného nezávislého parametru.

2.1.6 Zobrazování objemů

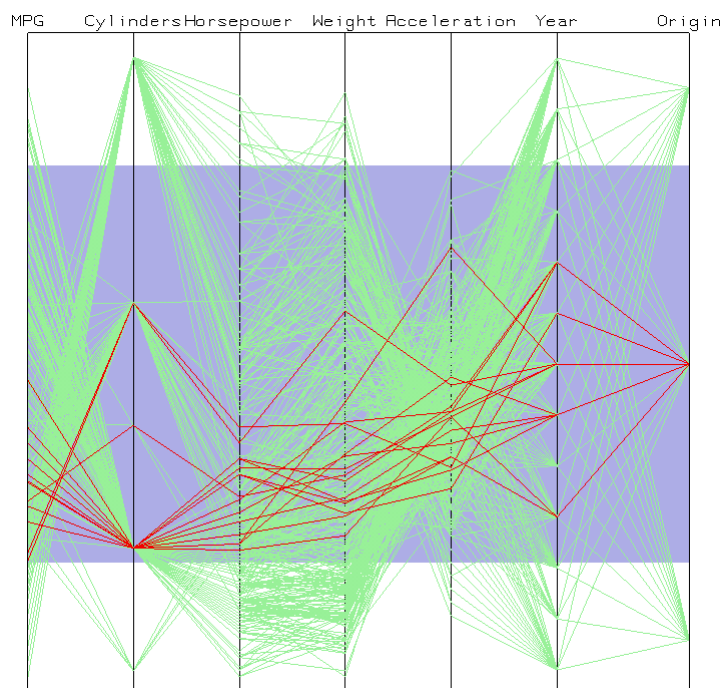
Tato technika umožňuje okamžitý náhled na skalární data v třídímníonálním prostoru. Data nemapujeme podle první dimenze na geometrický útvar (jako v iso-plochách), ale každému bodu dat je přiřazena barva a průhlednost.



Obrázek 2.6: Zobrazování objemů (viz [SV])

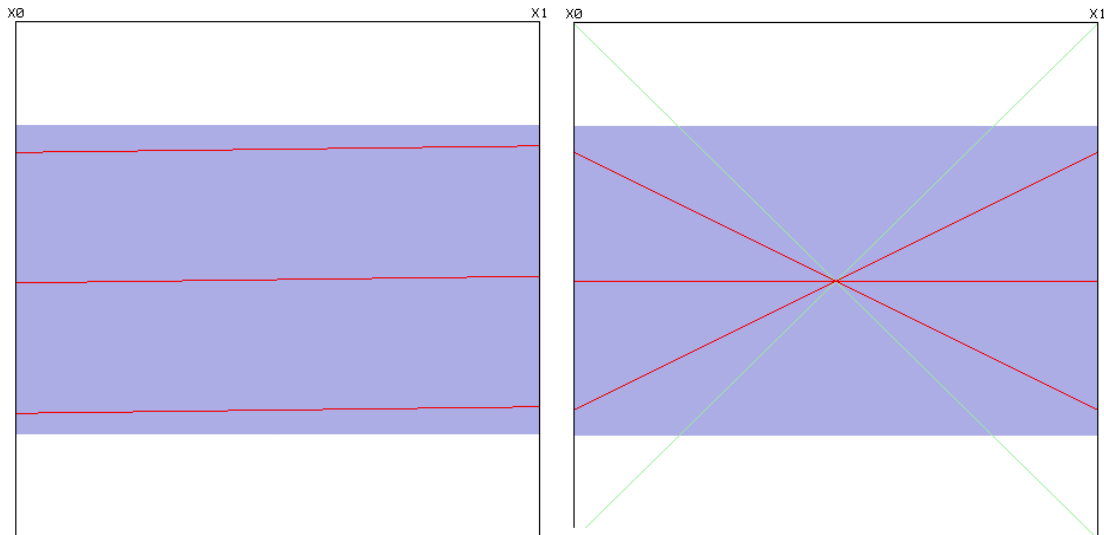
2.1.7 Paralelní souřadnice

Tato technika umožňuje n -dimensionální prostor zobrazit na 2D povrch, kde je zobrazena každá dimenze na vertikální osu a také na příslušný prostorový interval na horizontální ose.



Obrázek 2.7: Paralelní souřadnice (viz [MVT])

Výhody tohoto zobrazení spočívají v jednoduchém hledání vzájemných vztahů mezi daty. Jednoduchostí hledání vztahů míníme, že například na obrázku 2.8 vlevo mají tři různé vektory dat stejnou hodnotu v souřadnicích x_0 a x_1 , kdežto na obrázku 2.8 vpravo má pět různých vektorů dat rozdílné hodnoty v souřadnicích x_0 a x_1 .

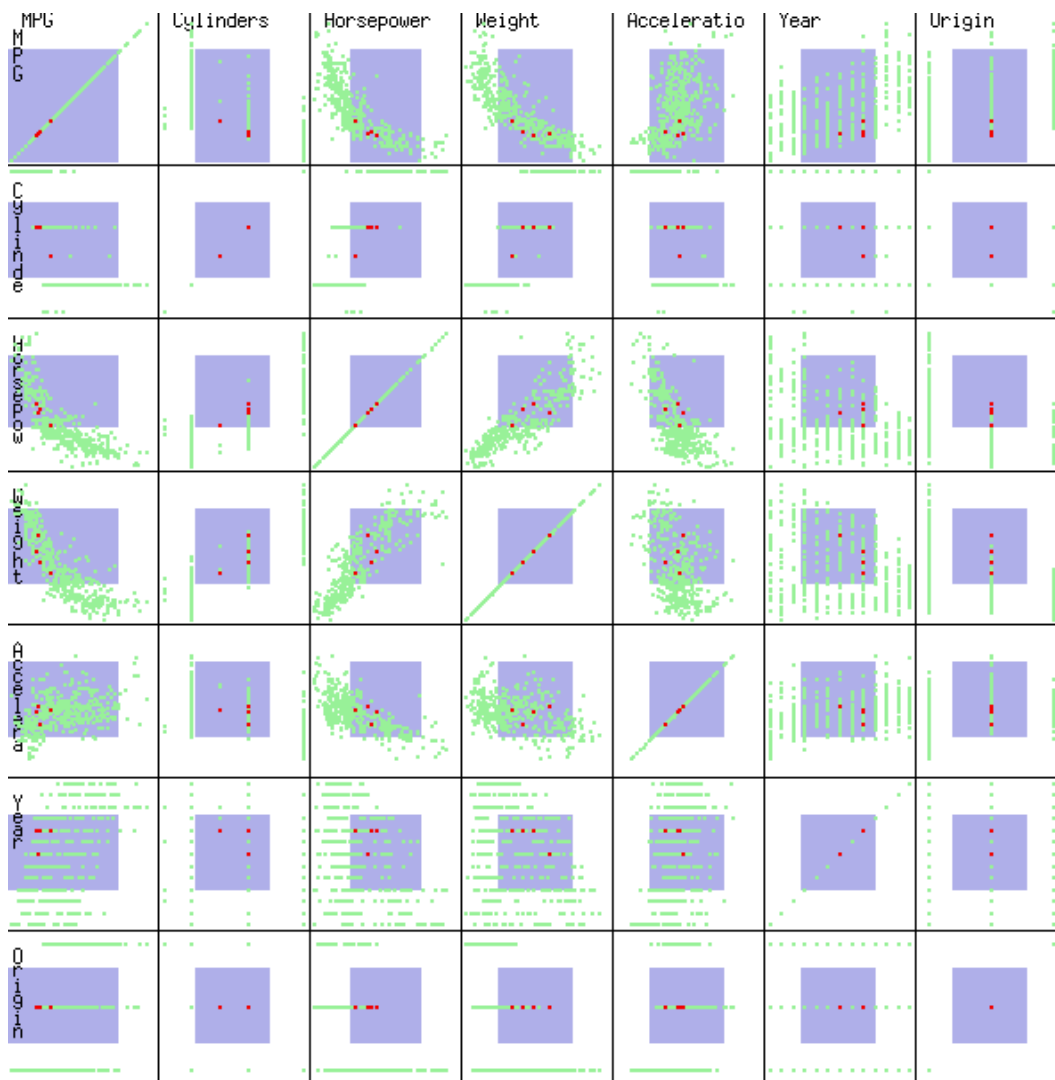


Obrázek 2.8: Paralelní souřadnice – výhody (viz [MVT])

Mezi nevýhody patří zhoršující se kvalita vizualizace při vzrůstajícím počtu dat a také tato metoda není vhodná pro pozorování dat jako celku, ale pouze pro hledání vztahů mezi daty.

2.1.8 Rozptýlené body

Tato technika vizualizace zobrazuje každou dimenzi proti ostatním dimenzím na 2D plochu. Takto definovaným zobrazením docílíme vždy porovnání každé dimenze se všemi ostatními. Mezi výhody patří použitelnost při velkých objemech dat a dále je tato vizualizace vhodná pro pozorování dat. Jako nevýhoda se jeví zobrazování velkého množství dimenzí (doporučené maximum je 8, viz. obrázek 2.9) a dále nutnost velké zobrazovací plochy.



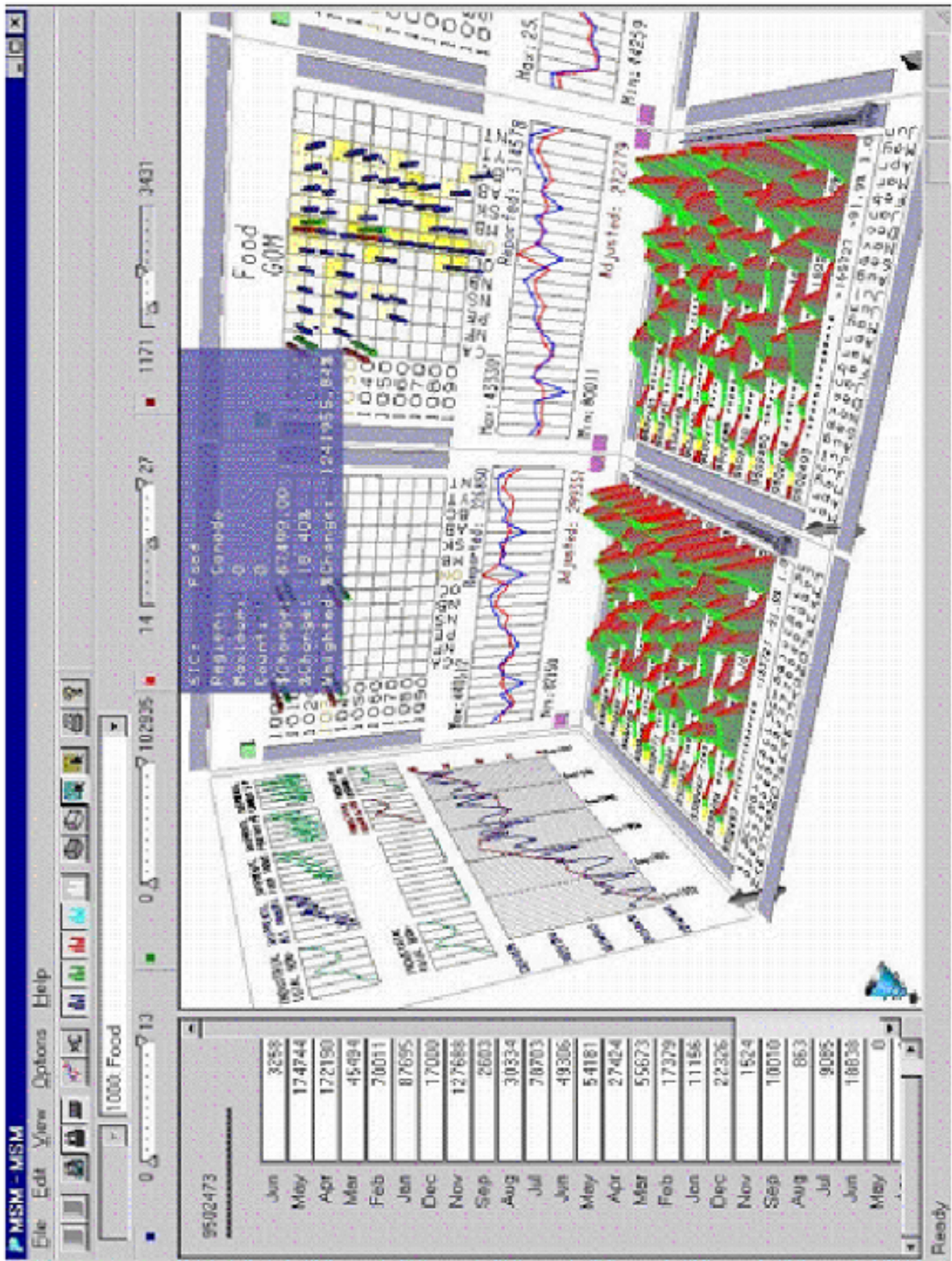
Obrázek 2.9: Rozptýlené body (viz [MVT])

2.1.9 Násobné grafy

Jsou navrhnuté pro komplexní sadu dat, kde vizualizace dat je prováděna jako sada současně viditelných grafů.

Vlastnosti a doporučení pro násobné grafy jsou následující:

- Konzistentnost – použití stejných měřítek pro všechny grafy
- Důraz na rozdíly – zobrazování stejného bodu do stejných grafů by mělo být provedeno různým způsobem, aby nedošlo k záměně
- Různé názvy – názvy grafů by měly mít výrazně odlišné názvy, aby nedošlo k záměně



Obrázek 2.10: Vizualizace násobnými grafy (viz [DataMin99])

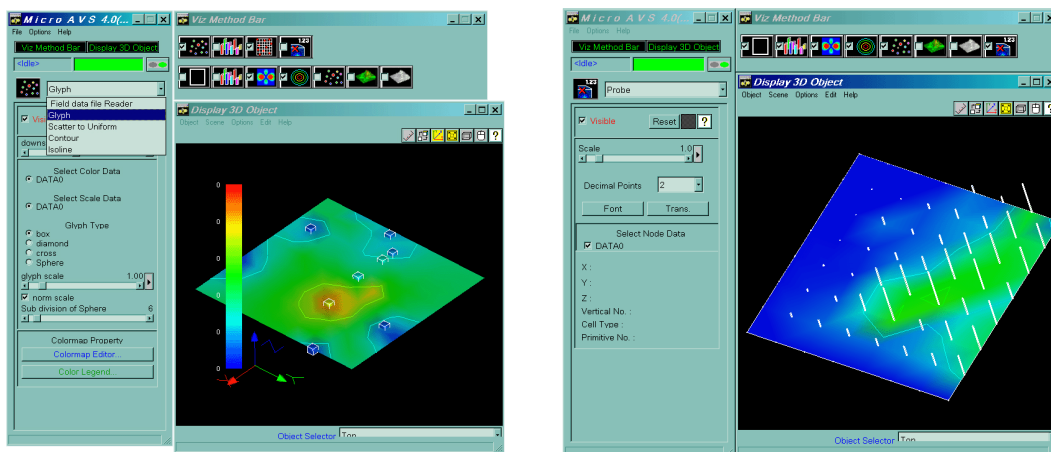
2.2 Aplikace řešící problém zobrazování N-D dat

Existuje celá řada programů řešících zobrazování N-D dat, ale jen malá část těchto programů je volně šiřitelná. Jako příklad volně šiřitelného programu uvádím *Micro AVS 4.0* (viz Kapitola 2.2.1) [AVS].

2.2.1 Micro AVS 4.0

Tento program vytvořila japonská firma KGT Inc. pro vizualizaci dat z různých oblastí týkajících se nejenom průmyslu.

Jedná se o program používající bodové a vektorové zobrazení. Tento vizualizační prostředek využívá zobrazení pomocí glyfů, isočar, isopovrchů a pomocí mnoha dalších metod.



Obrázek 2.11: Micro AVS 4.0 (viz [AVS])

3 Vizualizace pomocí isočar a rotační symetrie

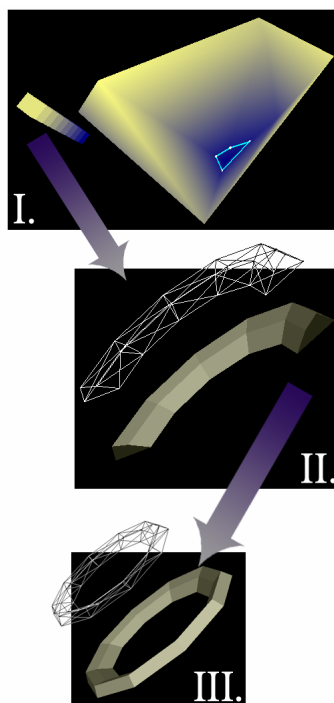
Jako vhodný vizualizační prostředek pro reprezentaci výškově závislých dat se jeví vizualizace pomocí isočar.

Isočáru můžeme definovat jako konturu, která je výškově závislá na datech. Tato data protíná v bodech, které splňují požadavek, který je definován jako konstantní hodnota (zvolená výška).

Množina isočar je výchozí (vstupní) datová množina pro vytvoření rotačního tělesa.

3.1 Náhled na problém

Pro vizualizaci iso-ploch ve 3D prostoru potřebujeme získat nejprve isočáry z trojúhelníkové sítě. Tyto isočáry dále rotujeme podle zvolené osy rotace, čímž získáme rotační iso-plochu ve 3D (viz obrázek 3.1).

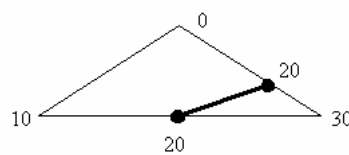


Obrázek 3.1: Tvorba rotační iso-plochy ve 3D

3.2 Hledání isočar ve 2D trojúhelníkové síti

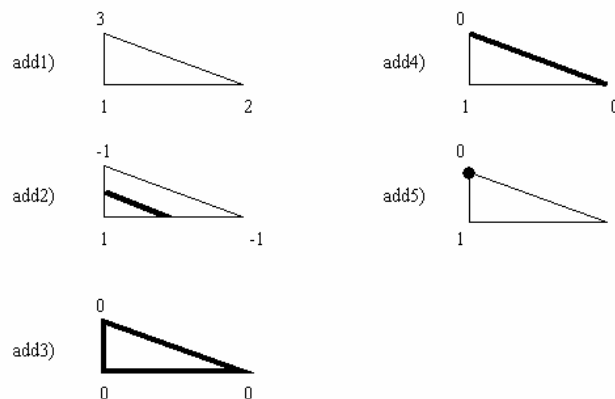
Vycházíme z toho, že v 2D trojúhelníkové síti má každý bod souřadnice $[x,y]$ s ohodnocením v každém bodě. Tímto ohodnocením je míněna jedna z dalších dimenzí (teplota, tlak, atd).

V každém trojúhelníku se hledá isočára podle kritéria, které splňuje počáteční podmínku (vstupní konstanta), tj. kupříkladu teplota rovna $20\text{ }^{\circ}\text{C}$. Viz následující obrázek:



Obrázek 3.2: Hledání isočáry, kde konstanta pro hledání isočáry je rovna 20

Jak je vidět z obrázku 3.2, trojúhelník daná isočára protíná ve dvou bodech. Všechny možnosti, jak daná hodnota může protnout trojúhelník a vytvořit tak množinu isočar, jsou zobrazeny na dalším obrázku 3.3:



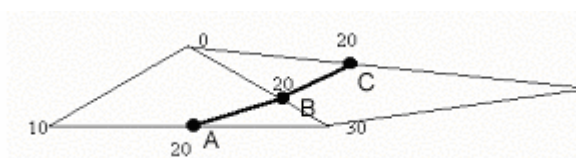
Obrázek 3.3: Všechny možnosti protnutí trojúhelníku isočárou

Tyto možnosti lze popsat podle následující tabulky, která obsahuje jednak popis výše uvedených situací a jednak akci, která vyjadřuje činnost algoritmu - hledání isočar v trojúhelníkové síti.

Možnost č.	Popis	Následná akce
1	zadaná mez trojúhelník vůbec neprotíná	nic se neděje, pokračuje se v hledání
2	protíná trojúhelník ve dvou bodech	klasická isočára, uloží se do seznamu isočar
3	protíná trojúhelník ve všech bodech	celý trojúhelník patří do výběru, do seznamu se uloží všechny tři nalezené isočáry
4	protíná trojúhelník jen ve dvou bodech, které určují vrcholy trojúhelníka	uloží se jako klasická isočára
5	protíná trojúhelník jen v jednom bodě	nic se neděje, pokračuje se v hledání

Tabulka 3.1: Popis všech možností, jak isočára může protnout trojúhelník

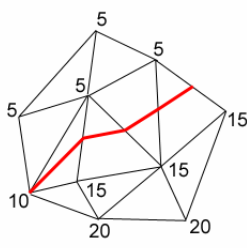
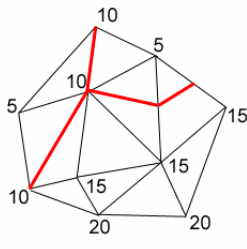
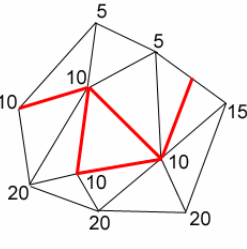
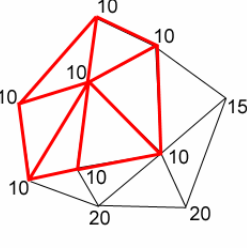
Postupné hledání isočáry v každém trojúhelníku vede k získání propojených úseků isočar (viz obrázek 3.4).



Obrázek 3.4: Průběh napojování isočar – úsek tvořený body ABC

3.3 Propojování úseků isočar

Problém propojení úseků se zdá být na první pohled bez větších úskalí. Ale jestliže se na problém podíváme důkladněji zjistíme, že můžeme nalézt čtyři základní varianty, ze kterých se může finální útvar libovolně skládat.

Varianta:	Náhled (příklad):	Reálný geografický příklad:
1)		Řeka
2)		Soutok dvou řek
3)		Řeka a rybník
4)		Jezero

Tabulka 3.2: Varianty úseků isočar

Problém lze vyřešit algoritmem, který respektuje všechny uvedené případy. Tento algoritmus pracuje se zásobníkem podle následujícího popisu:

- I) Hledání startovních bodů – pro úseky typu 1,2 a 3
 - Pro každou isočáru proved':
 - Detekce, zda je jeden nebo druhý bod isočáry v množině isočar obsažen pouze 1x. Jestliže ano, ulož ho do zásobníku.

(složitost – $O(N^2)$, přičemž N je počet isočar)

II) Vlastní algoritmus – pro všechny čtyři typy

- Nastav všechny úseky jako nezpracované. (složitost $O(N)$)
- Jestliže počet zbývajících nezpracovaných úseků je větší než nula a zásobník je prázdný proved': (pro typ 4)
 - Vyber první nezpracovaný úsek a vlož ho do zásobníku.
($O(N)$ v nejhorším případě)

- Když není zásobník prázdný proved':

(složitost $O(S)$, kde S je počet segmentů)

- Vyber úsek ze zásobníku.
- Zpracuj daný úsek – tj. ulož ho do výsledného pole.
- Jestliže je úsek nezpracovaný proved':
 - Pro každý úsek proved':
 - Vyhledej návaznost na daný úsek, a jestliže daný úsek navazuje na tento úsek a je různý, ulož ho do zásobníku.
 - Vyjmi úsek ze zásobníku.

(složitost $O(N)$)

- Jestliže je úsek zpracovaný, vyjmi ho ze zásobníku.

(složitost $O(1)$)

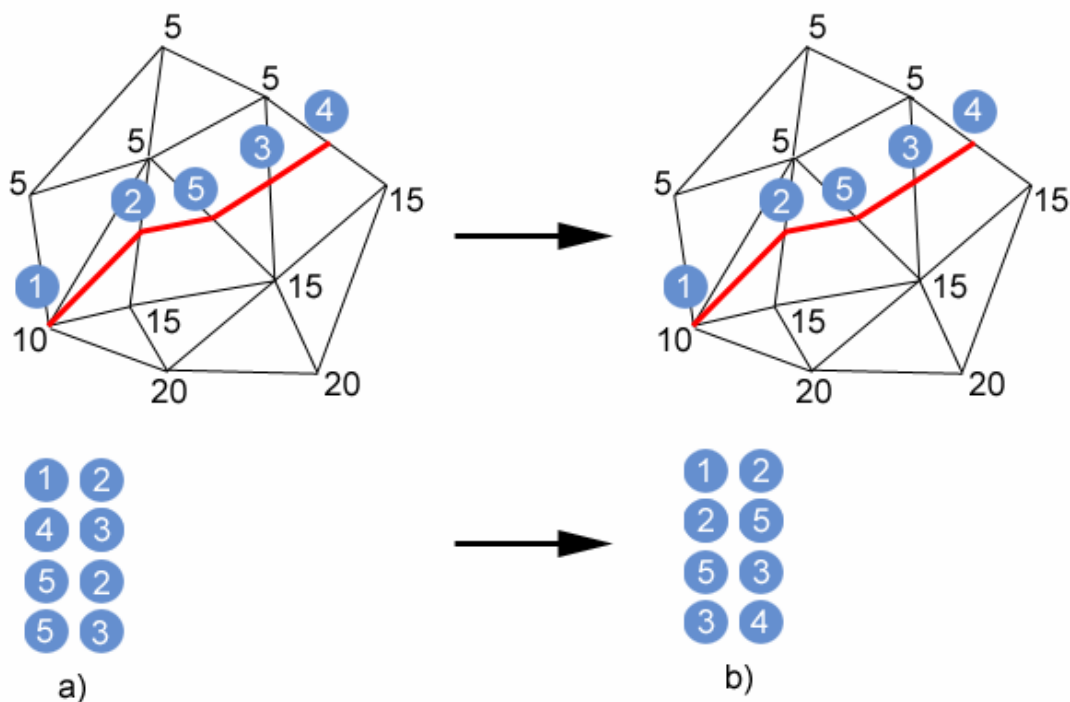
- Jestliže je zásobník prázdný a počet nezpracovaných úseků je větší než nula, proved':

Vyber první nezpracovaný úsek a vlož ho do zásobníku.

($O(N)$ v nejhorším případě)

Po provedení algoritmu máme v nově vytvořeném poli postupně uloženy všechny úseky tak, aby na sebe navazovali. Tento krok je důležitý pro správnou orientaci normál, protože kdybychom předchozí algoritmus nepoužili, orientace normál stejného úseku isočar by se mohla lišit, což by působilo rušivě.

Výsledkem algoritmu je seznam seřazených úseků, tak jak jdou za sebou (viz. obrázek 3.5). Na obrázku je vidět seznam čtyř isočar. Jestliže bychom tyto isočáry neseřadili, dostali bychom po rotaci těleso, které by mělo 2x prohozené normály. Konkrétně by se jednalo o isočáry 43 a 52. Tento algoritmus je nutný pouze pro správnou vizualizaci.



Obrázek 3.5: Napojování úseků.

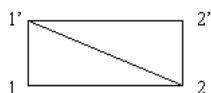
a) neseřazený seznam isočar, b) seřazený (postupný) seznam isočar

3.4 Vytvoření rotačního tělesa z množiny isočar

Takto připravený seřazený seznam isočar (viz kap. 3.1) je vstupem pro rotační část úlohy. Rotační těleso lze jednoduše vytvořit podle vzorce pro rotaci kolem osy x , kde transformujeme každý bod $[x, y, z]$ do nových souřadnic $[x', y', z']$ podle [Zar98]:

$$\begin{aligned}
 x' &= x \\
 y' &= \cos\beta * y - \sin\beta * z \\
 z' &= \sin\beta * y + \cos\beta * z
 \end{aligned}
 \tag{3.1}$$

Pro naši úlohu nejprve v již vytvořeném seznamu vynulujeme z-ovou složku (počáteční nastavení rotace) a dále seznam isočar postupně rotujeme kolem osy x. Tato rotace se provádí ve dvou krocích. Nejprve se zapisuje do sítě trojúhelník [1,2,1'] a poté [1',2',2] (1 a 2 jsou původní body isočáry, 1' a 2' jsou otáčivé body o úhel určující rotaci). Tímto postupem získáváme množinu trojúhelníků (výstupní trojúhelníková síť). Viz obrázek 3.6.



Obrázek 3.6: Napojování trojúhelníků při rotaci

Takto uvedeným postupem bychom získali rotační těleso, které je celé rotováno dle osy x, čímž bychom se omezili pouze na vybranou množinu dat. Proto definujeme rotaci podle libovolné osy definované vektorem:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Rotační matice pro rotaci kolem libovolné osy vypadá podle [GTMath] následujícím způsobem:

$$R_{u,\theta} = \begin{bmatrix} u_1^2 + (1 - u_1^2)c & u_1 u_2 (1 - c) - u_3 s & u_1 u_3 (1 - c) - u_2 s \\ u_1 u_2 (1 - c) - u_3 s & u_2^2 + (1 - u_2^2)c & u_2 u_3 (1 - c) - u_1 s \\ u_1 u_3 (1 - c) - u_2 s & u_2 u_3 (1 - c) - u_1 s & u_3^2 + (1 - u_3^2)c \end{bmatrix} \quad (3.2)$$

Přičemž $c = \cos(\theta)$ a $s = \sin(\theta)$ pro přehlednost.

Takto definovaný vztah platí pro jednotkový směrový vektor u . Následná zřetěžená transformace R pro rotaci kolem libovolné osy vypadá následujícím způsobem:

$$R = T^{-1} R_{u,\theta} T$$

Přičemž T^{-1} je matice posunu do počátku a T je inverzní maticí k T^{-1} . Matice T vypadá následujícím způsobem (viz [Skala92]):

$$T = \begin{bmatrix} 1 & 0 & -x_1 \\ 0 & 1 & -y_1 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

4 Implementace

V této části bude popsána realizace uvedených postupů a principů vizualizace.

4.1 Zvolené prostředí pro realizaci

Jak již vyplývá ze zadání úlohy, program musí být realizován jako knihovna DLL pro modulární systém MVE [Rou00], proto byl zvolen systém Microsoft Windows ve verzích 9x/Me/2000/NT/XP.

Jako programový nástroj byl vybrán produkt firmy Inprise Corporation Borland Delphi 5. Autor tento produkt zvolil díky předchozím znalostem s tímto vývojovým nástrojem.

Pro vizualizační část byla použita knihovna OpenGL firmy Sgi.

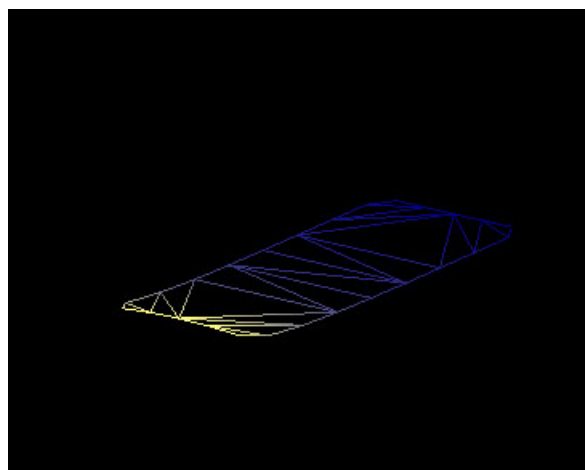
4.2 Vizualizace N-D dat

Vizualizace vstupních dat využívá principy uvedené v části 2.

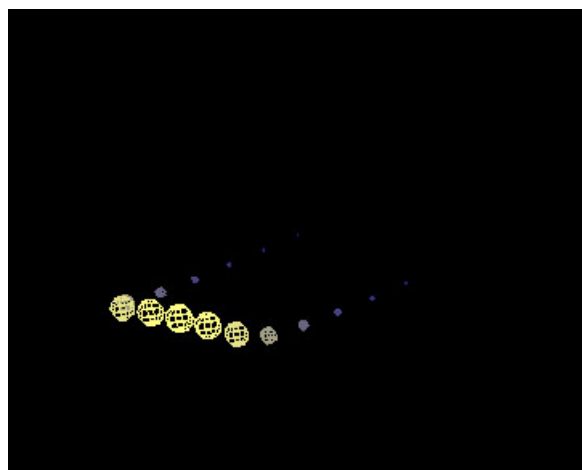
Konkrétně se jedná o následující části:

1. 2D vizualizace sítě s informací o třetím rozměru obsahuje následující informace:
 - X-ová souřadnice
 - Y-ová souřadnice
 - Z-ová souřadnice – třetí rozměr
 - Barva
2. Glyfy – jednoduché objekty popisující data v zadaném místě, které obsahují následující informace:
 - Barva
 - Velikost
3. Jednoduché vektory - jsou definované směrovým vektorem v každém bodě a obsahují následující informace:
 - X-ová souřadnice
 - Y-ová souřadnice
 - Z-ová souřadnice
 - Barva
4. Plné vektory – stejná funkčnost jako jednoduché vektory, ale s tím rozdílem, že vizualizace je provedena odlišným způsobem

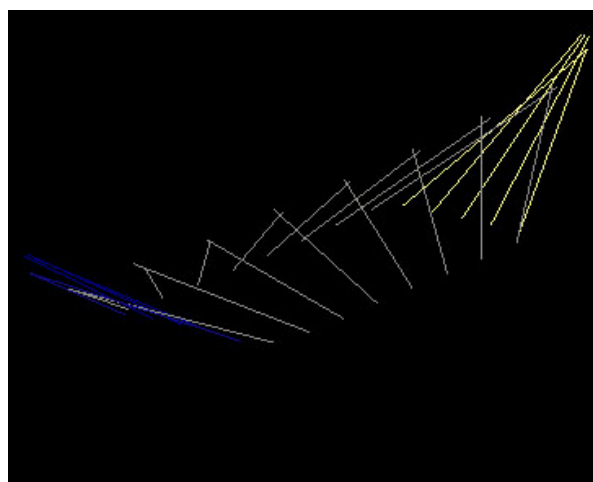
Náhledy na jednotlivé části jsou uvedeny v následujících obrázcích 4.1 a 4.2:



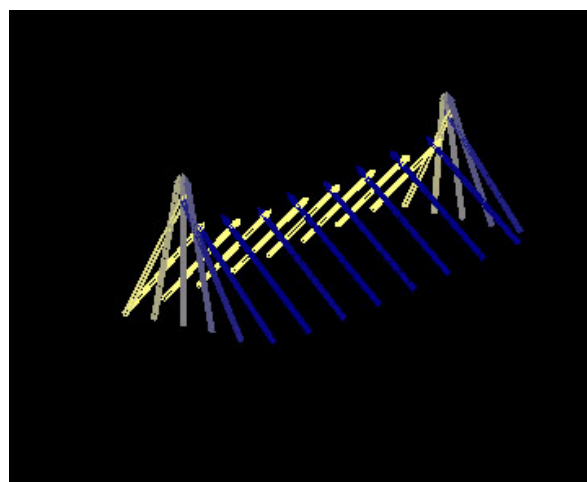
A



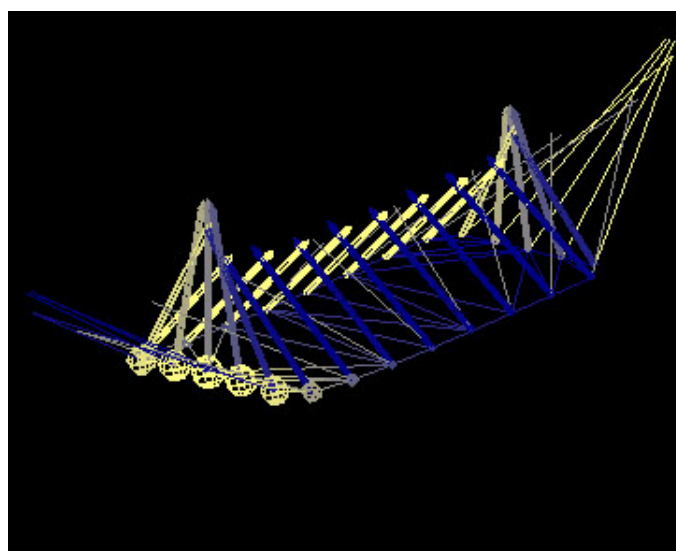
B



C

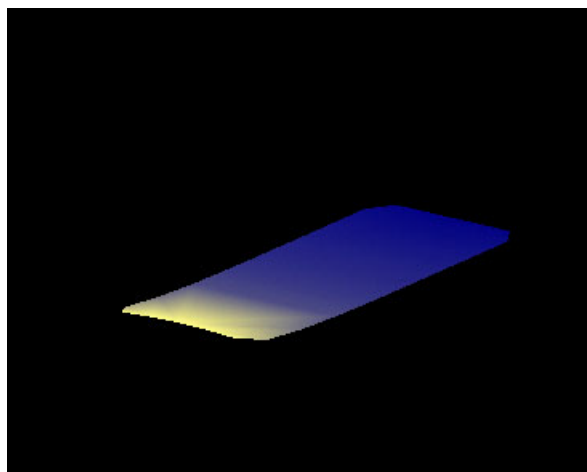


D

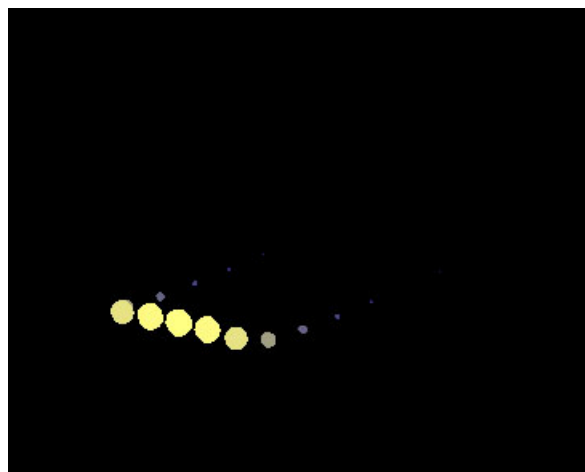


E

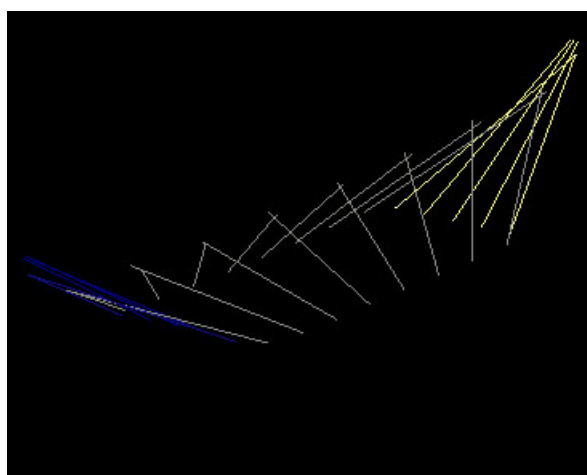
Obrázek 4.1: Principy vizualizace – pro drátěný model,
A – síť, B – glyfy, C – jednoduché vektory, D – plné vektory a E - všechny 4 části ABCD



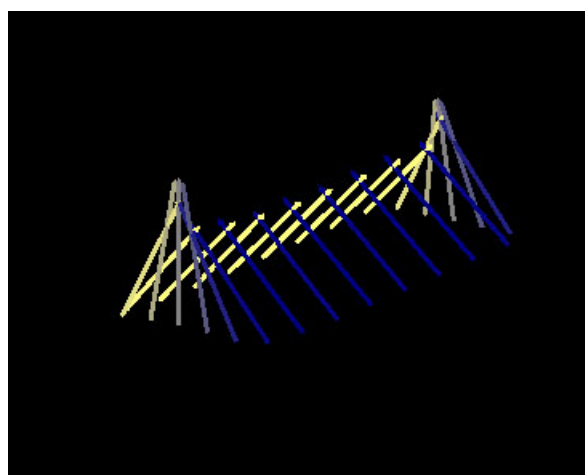
A



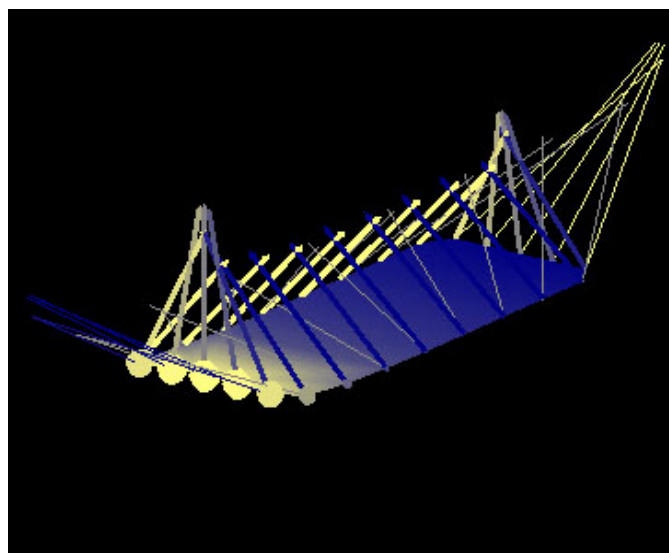
B



C



D



E

Obrázek 4.2: Principy vizualizace – pro plošnou reprezentaci,
A – síť, B – glyfy, C – jednoduché vektory, D – plné vektory a E - všechny 4 části ABCD

Těmito vizualizačními metodami docílíme celkem 14 dimenzí.

4.3 Realizace vizualizace pomocí rotační symetrie

Celý algoritmus pro vytvoření rotačního tělesa lze shrnout do těchto částí:

1. načtení dat
2. zvolení osy rotace
3. zvolení z-ové souřadnice a její hodnoty, podle které se budou hledat isočáry
– vizualizace vstupních dat
4. hledání isočar (viz 3.2)
5. odstranění duplicit (pro typ „jezero“)
6. propojování úseků isočar (viz 3.3)
7. transformace do počátku
8. provedení rotace kolem zvolené osy
9. transformace zpět
10. vizualizace rotačního tělesa - výstup

4.3.1 Formát dat

Jak již plyne z předchozích faktů, modul pro zobrazení N-rozměrných dat vyžaduje jako vstup minimálně 2D data s informací v každém bodě. Hlavním vstupem modulu je triangularizovaná 2D síť s dalšími informacemi o dalších hodnotách. Triangularizaci provádí modul Doc. Dr. Ing. Ivany Kolingerové `DTLib2.dll` (viz. **[Kolinger02]**) s podporou N-D dat.

Vlastní vstup (soubor) do modulu DTlib2.dll je realizován následovně:

```
10
0.1860 0.6415 0.0000 1.0000 2.0000 0.0000 2
0.1003 0.2919 0.2000 1.0000 2.0000 0.0000 3
0.9346 0.8100 0.3000 1.0000 2.0000 0.0000 4
0.9332 0.7182 0.4000 1.0000 2.0000 0.0000 3
0.9334 0.9280 0.5000 1.0000 2.0000 0.0000 7
0.0853 0.9923 0.6000 0.0000 2.0000 0.0000 3
0.3812 0.1081 0.7000 0.0000 2.0000 0.0000 7
0.4596 0.0762 0.8000 0.0000 2.0000 1.0000 0
0.0785 0.0525 0.9000 0.0000 2.0000 1.0000 0
0.8170 0.7745 1.0000 0.0000 2.0000 1.0000 0
```

Obrázek 4.1: příklad vstupního souboru

Na prvním místě ve vstupním souboru je počet bodů. Dále jsou uvedeny souřadnice/hodnoty každého bodu. Na prvních dvou místech jsou uvedeny souřadnice X,Y a jsou směrodatné, tzn. jsou nutné pro chod programu a nelze je zaměnit s jinými dimenzemi vstupních dat při nastavování vstupu – vstupní vizualizační část (viz dále).

5 Dosažené výsledky

Dále budou uvedeny nároky na systém a výsledky s testovacími a reálnými daty.

5.1 Časová a paměťová složitost

V této části budou uvedeny časové a paměťové nároky na systém.

5.1.1 Časová složitost

Zde budou rozebrány časové nároky jednotlivých dílčích částí algoritmu. Mezi hlavní a nevypustitelné části patří hledání isočar a algoritmus rotace. Algoritmy pro propojení úseků isočar a pro odstranění duplicit jsou zde jenom pro přehlednost a čistotu výsledného výstupu.

Všechny testy byly prováděny v laboratoři ZČU-FAV-IVT-PG UL407 na počítači DELL PIII 500 MHz s 1GB paměti.

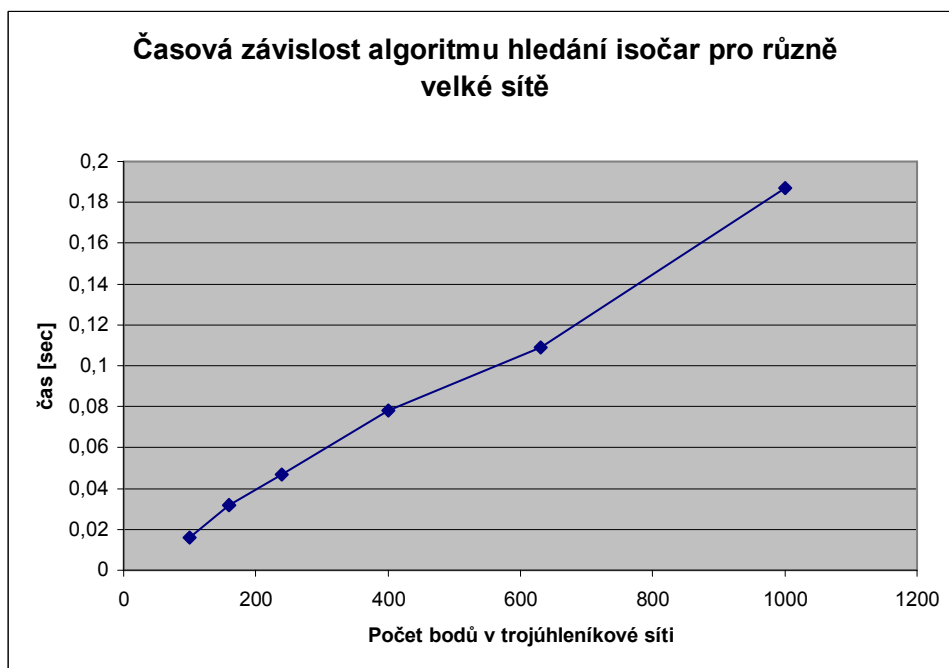
Pro testování byla použita testovací množina dat - 2D síť s ohodnocením v každém bodě. Ohodnocení bylo vypočítáno pomocí vztahu $z = [\tanh(9y-9x)+1]/9$, kde

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Hledání isočar

Algoritmus hledání isočar má složitost $O(N)$, jelikož celou síť procházíme pouze 1x a neprovádíme další dodatečné kontroly.

Chování algoritmu pro hledání isočar je zobrazeno na následujícím Grafu 5.1.



Graf 5.1

Odstranění duplicit

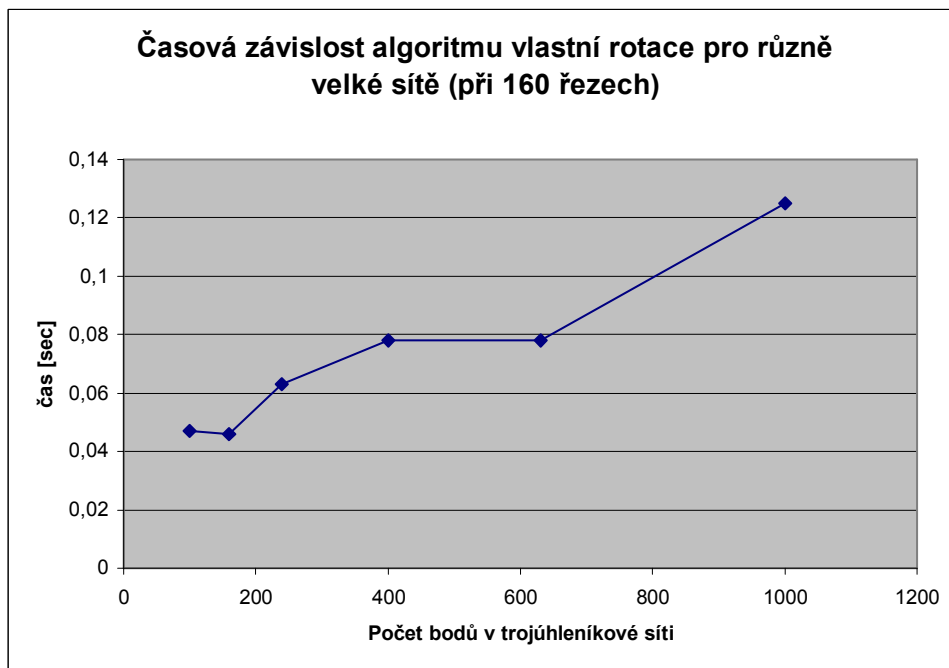
Tento algoritmus má složitost $O(N^2)$, kde N je počet isočar. Tento algoritmus je nepovinný a je zde jen pro přehlednost výstupu. Tento algoritmus by šel vylepšit použitím *hash* funkce (viz [SK01]), ale pro tento problém postačuje stávající řešení (viz graf 5.5).

Propojování úseků isočar

Tato část by se mohla zdát jednou z nejnáročnějších na časovou složitost, ale jak uvidíme dále tento algoritmus používající zásobník má malé nároky na čas, jelikož v síti většinou nenacházíme velké množství isočar. Tento algoritmus, jak již bylo uvedeno výše, není potřebný pro chod vlastního algoritmu a má složitost $O(N^2)$, kde N je počet isočar.

Algoritmus rotace

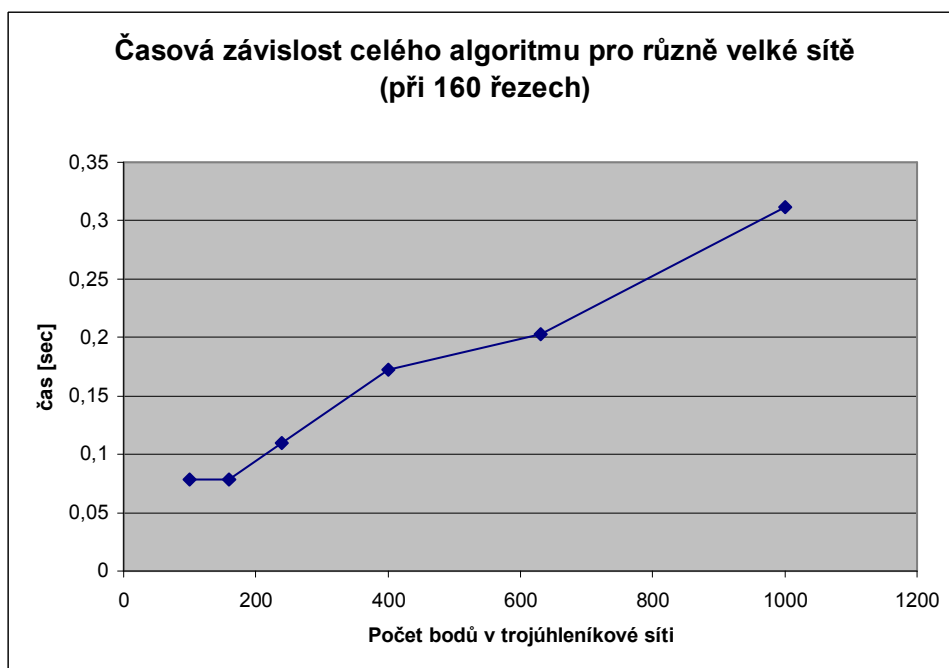
Tato hlavní část je poměrně rozsáhlým algoritmem, který má složitost $O(N \cdot C \cdot S)$, kde N je počet isočar, C je počet řezů a S je velikost jednoho segmentu isočar. Na grafu 5.2 je znázorněna časová závislost pro algoritmus rotace:



Graf 5.2

Jak je vidět z grafu 5.2, složitost algoritmu pro rotaci má složitost $O(N)$, jelikož C a S jsou konstanty.

Na dalším grafu 5.3 je znázorněna celková časová závislost algoritmu při konstantním počtu řezů. Je zřejmé, že teoretická složitost algoritmu se neliší od naměřených časů.



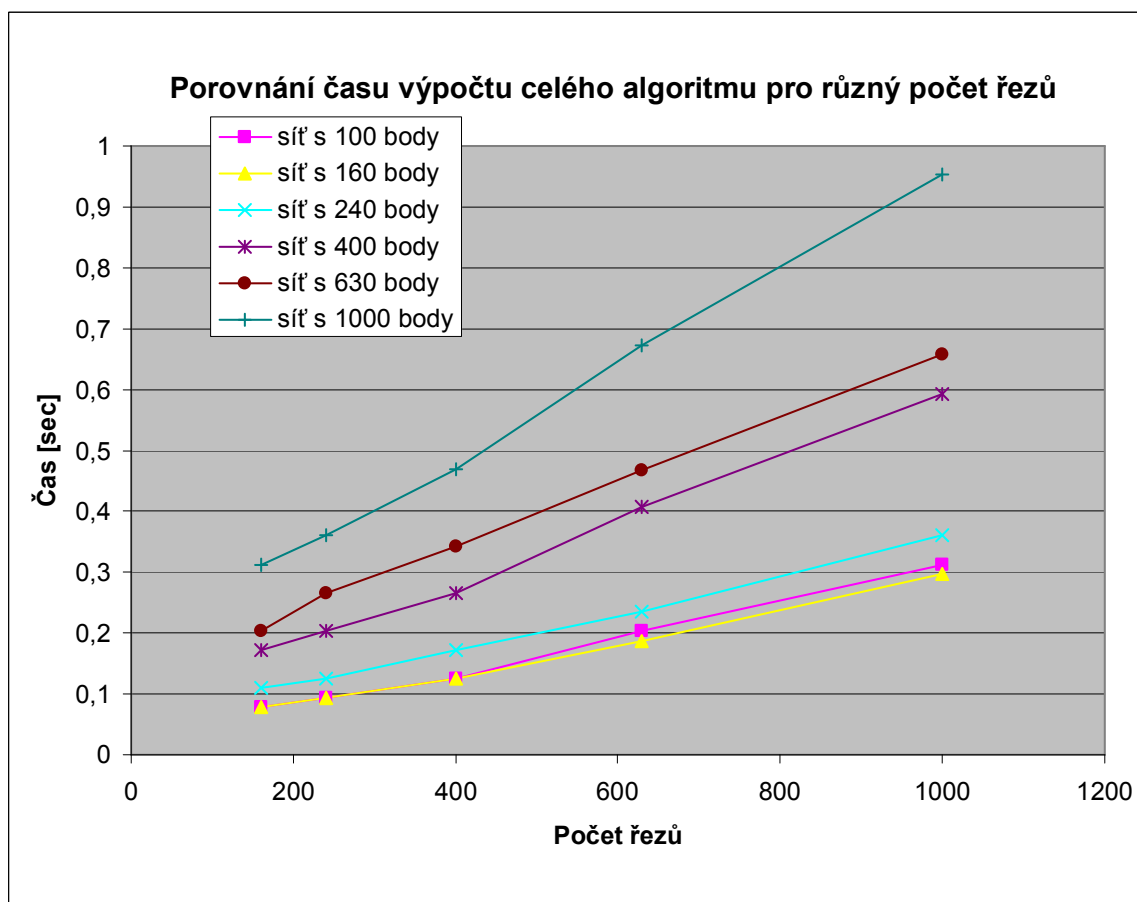
Graf 5.3

Následuje tabulka příslušných hodnot, ze kterých byly vytvořeny grafy 5.1, 5.2 a 5.3:

Počet bodů v síti	Celkový čas [sec]	Dílčí časy algoritmu [sec]				Nalezených isočar	Výstupní počet trojúhelníků
		Hledání isočar	Odstranění duplicit	Propojování úseků isočar	Vlastní rotace		
100	0,078	0,016	0,016	0	0,047	36	11520
160	0,078	0,032	0	0	0,046	31	9920
240	0,11	0,047	0	0	0,063	41	13120
400	0,172	0,078	0,016	0	0,078	66	21120
630	0,203	0,109	0,016	0	0,078	72	23040
1000	0,312	0,187	0	0	0,125	97	31040

Tabulka 5.1: Porovnání časové složitosti jednotlivých částí algoritmu pro konstantní počet řezů (160)

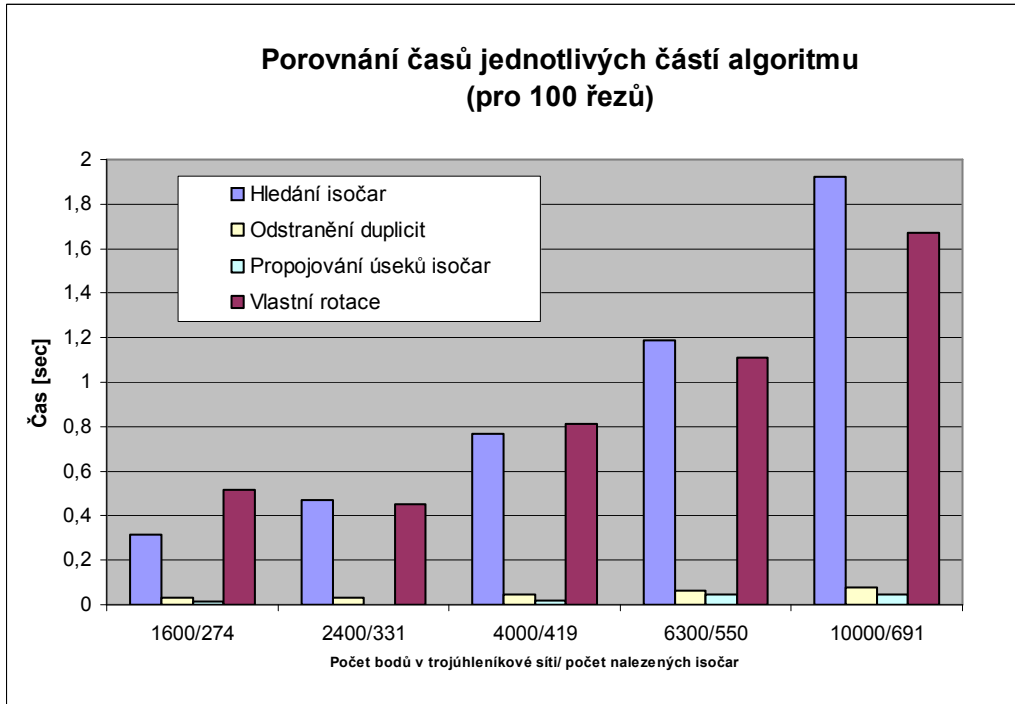
Na dalším grafu 5.4 vidíme závislosti časů pro různý počet řezů a pro různě velké sítě:



Graf 5.4

Pro jednotlivé porovnání dílčích částí algoritmu byla použita testovací množina $[x,y,z]$,
 přičemž $z = \sin(4 * \pi * x) \exp(-2 * y - 1.5)$.

Výsledky testů jsou na následujícím grafu 5.5:



Graf 5.5

Následuje příslušná tabulka 5.2 pro graf 5.5:

Vstupní počet bodů		1600	2400	4000	6300	10000
Vstupní počet trojúhelníků		3182	4778	7972	12573	19973
Nalezený počet isočar		274	331	419	550	691
Počet trojúhelníku na výstupu		54800	66200	83800	110000	138200
Počet bodů na výstupu		27500	33200	42206	55100	69200
Celkový čas [sec]		0,875	0,953	1,641	2,406	3,719
Dílčí časy [sec]	Hledání isočar	0,313	0,469	0,766	1,188	1,922
	Odstranění duplicit	0,031	0,031	0,047	0,062	0,078
	Propojování úseků isočar	0,015	0	0,016	0,047	0,047
	Vlastní rotace	0,516	0,453	0,812	1,109	1,672

Tabulka 5.2 : Časové nároky pro konstantní počet řezů 100

5.1.2 Paměťová složitost

Paměťové nároky celého algoritmu jsou jednak závislé na vstupních datech a dále také na velikosti časového souboru, ve kterém jsou uvedeny průběhy veličin (časový průběh jednotlivých veličin může, ale nemusí být použit). Další závislosti lze odvodit z celkového poměru nalezených isočar ve vstupních datech.

V těchto alokacích není započítána ostatní režie programu, ale vzhledem k datům je zanedbatelná.

Vstupní data

Závislost vstupních dat je ovlivněna jednak dimenzí a jednak velikostí dat.

Pro alokaci pole bodů je potřeba: $8B * \text{počet_bodů} * \text{počet_dimenzí}$

Pro alokaci vlastní vstupní trojúhelníkové sítě je potřeba: $4B * \text{počet_trojúhelníků} * 3$

Dalším vstupem můžou být data, která jsou časově závislá. Pro tato data se alokuje další paměť s tím rozdílem, že se nealokují dimenze pro souřadnice X a Y.

Dočasná alokace - Hledání isočar

Maximální počet všech isočar je $3 * \text{počet_trojúhelníků}$. Alokace paměti je následující:

Pro alokaci bodů je potřeba: $8B * \text{počet_bodů} * \text{počet_dimenzí} * 2$

(Pozn. Celý výraz je vynásoben 2, protože vstupem mohou být degenerovaná data)

Pro alokaci isočar: $4B * \text{počet_trojúhelníků} * 3$

Dočasná alokace - Odstranění duplicit

Pro tento jednoduchý algoritmus se dočasně alokuje paměť pro mezivýsledek.

Alokace pro dočasnou množinu isočar je následující: $4B * \text{počet_nalezených_isočar} * 2$

Výstupní data - Algoritmus rotace

Maximální počet výsledných trojúhelníků je ovlivněn počtem řezů. Tímto získáváme následující paměťové nároky:

Pro alokaci bodů: $8B * \text{počet_bodů_isočar} * \text{počet_dimenzí} * \text{počet_řezů}$

Pro alokaci trojúhelníkové sítě: $4B * 3 * \text{počet_isočar} * \text{počet_řezů} * 2$

(Pozn. Celý výraz je vynásoben 2, protože ke každé isočáře jsou generovány dva trojúhelníky v každém kroku rotace)

Stejná alokace jako pro trojúhelníky je provedena pro normály s tím rozdílem, že výraz je vynásoben ještě dvěma, jelikož datový typ pro skalární hodnoty je 2x než pro index do pole bodů.

5.2 Zobrazení vstupních a výstupních dat

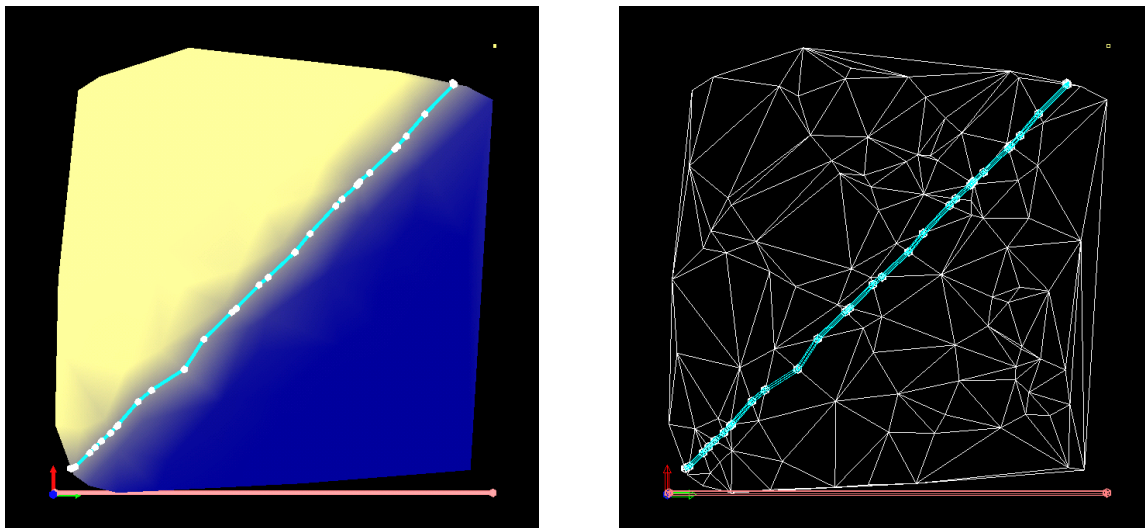
Tato podkapitola se bude věnovat použitým metodám pro zobrazení N-D dat na testovacích a reálných datech.

5.2.1 Testovací množina dat

V podkapitole 5.1 byly pro testování použity dvě testovací množiny.

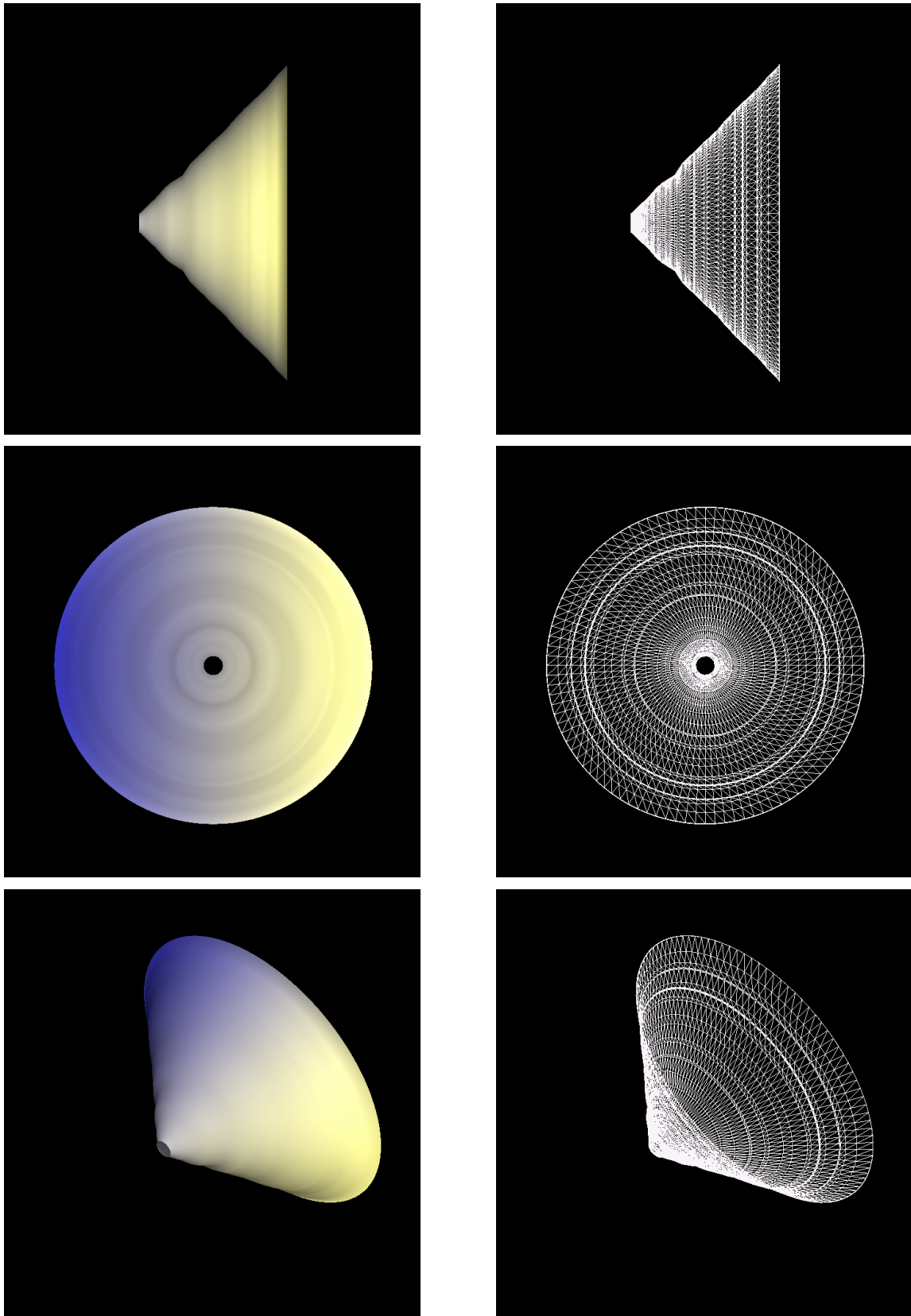
Pro první uvedenou množinu dat $[x,y,z]$, přičemž $z = [\tanh(9y-9x)+1]/9$, dostáváme následující výsledky:

Na obrázku 5.1 vidíme vstupní množinu dat s vypočtenými isočarami (světle modrá barva) a osou rotace (červená barva). Z-ová souřadnice je vstupní hodnotou pro barevné spektrum celé sítě. Počet bodů trojúhelníkové sítě na obrázku 5.1 je 100.

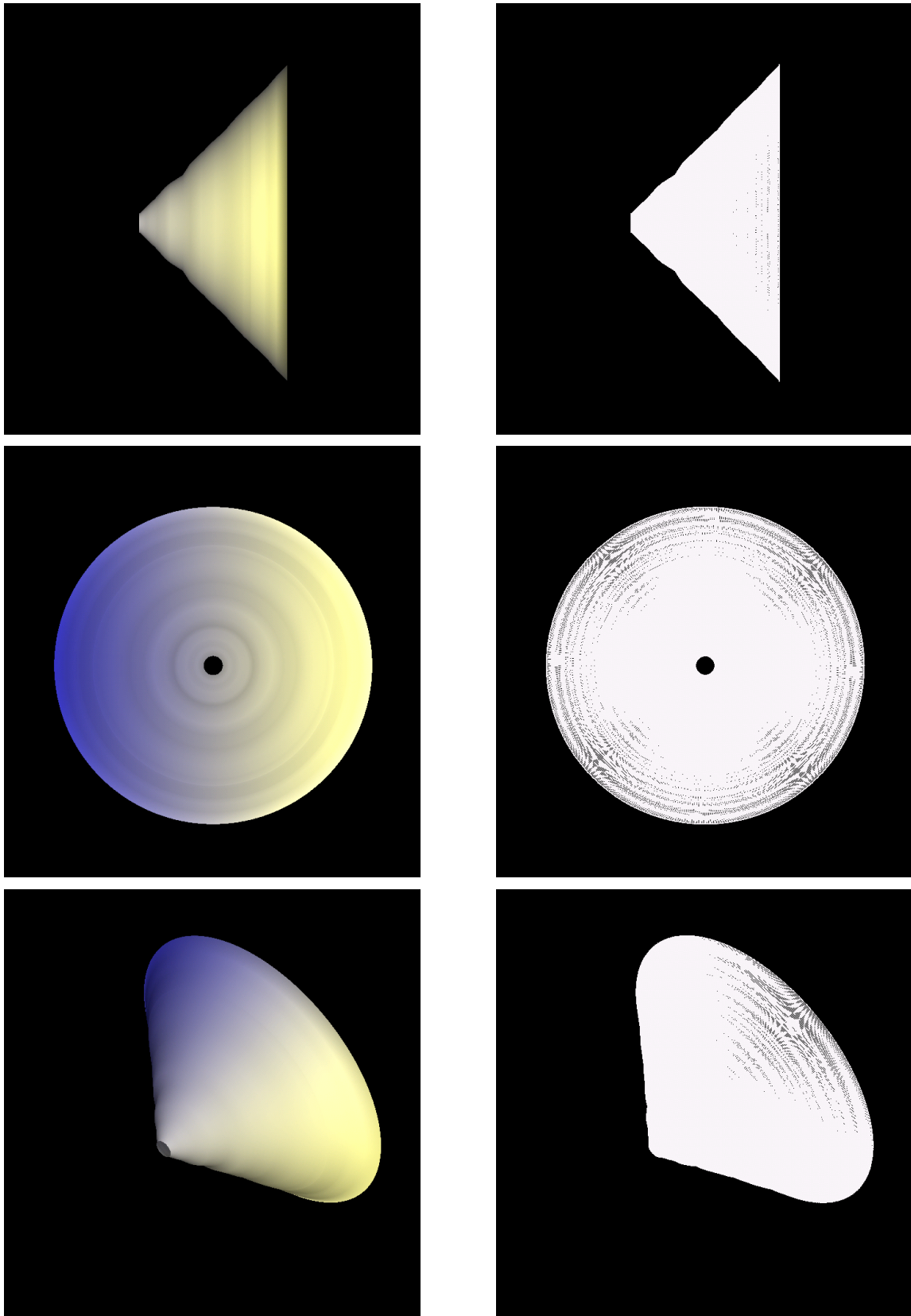


Obrázek 5.1: Vstupní testovací množina dat se 100 body
(vlevo stínovaný model, vpravo drátěný model)

Na další sérii obrázků (obrázek 5.2 a obrázek 5.3) již vytvořeného rotačního tělesa můžeme pozorovat rozdíl výstupu při různém počtu řezů.



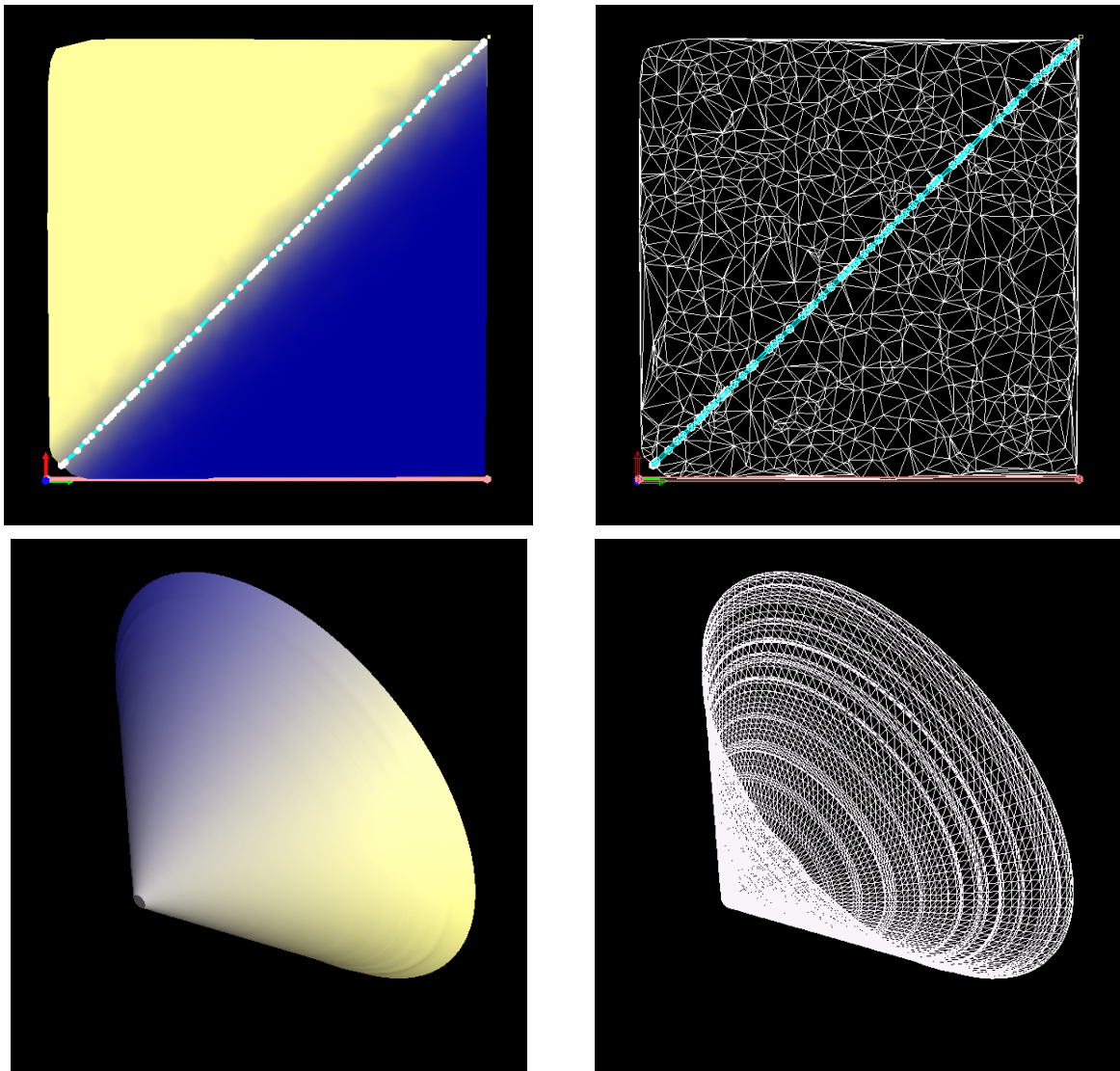
Obrázek 5.2: Výstup pro testovací množinu dat
(vlevo stínovaný model, vpravo drátěný model) při počtu řezů 100



Obrázek 5.2: Výstup pro testovací množinu dat
(vlevo stínovaný model, vpravo drátěný model) při počtu řezů 1000

Jak je patrné z obrázků 5.1 a 5.2, zvýšením počtu řezů zdaleka nedocílíme přehlednosti a dále se rapidně zvyšují nároky na rychlost systému. Pro většinu vstupních dat je dostačující použití 100 až 200 řezů nebo dokonce méně. Dalším aspektem rychlosti vizualizace může být použití dalších vizualizačních prvků jako jsou glyfy nebo vektory, jak uvidíme dále.

Jestliže zvýšíme počet bodů trojúhelníkové sítě na 1000 získáme následující výsledky, které jsou zobrazeny na obrázku 5.3:

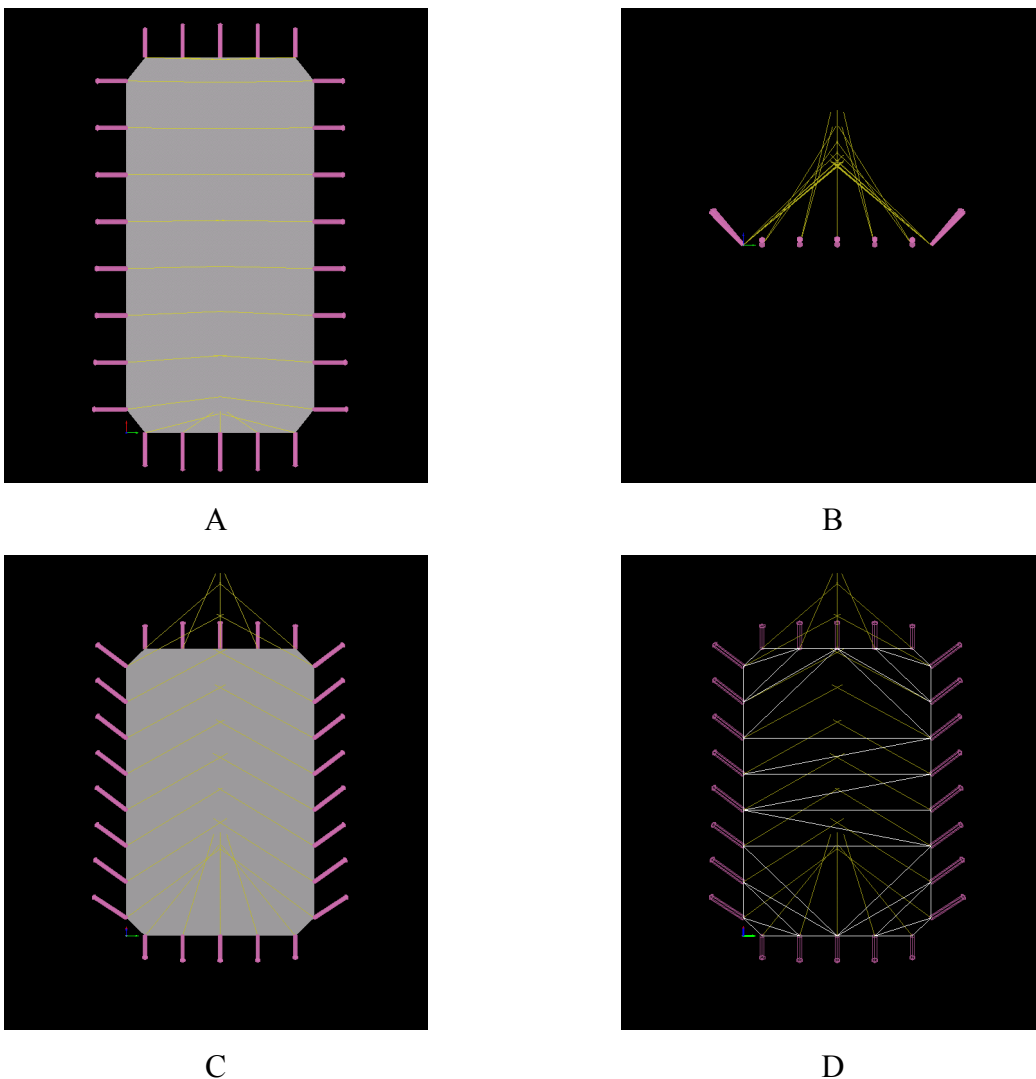


Obrázek 5.3: Vstupní testovací množina dat s 1000 body a rotovaný výstup o 100 řezech (vlevo stínovaný model, vpravo drátěný model)

5.2.2 Reálná data - průmysl

Aplikace byla vyvíjena částečně na žádost Katedry teoretické elektrotechniky pro vizualizaci dat, obsahují informace o vektorech v každém bodě (Síla, Magnetická indukce, atd.).

Jako první úloha pro katedru teoretické elektrotechniky byla testována úloha vizualizace magnetické podušky [KTE02a]. Tato úloha je simulací systému EDL (systém s elektrodynamickou levitací) pro železniční dopravu. Úkolem bylo zobrazení vektorů síly a magnetické indukce. Vstupními daty byly hodnoty následujícího formátu: $[x, y, z, F_x, F_y, F_z, B_x, B_y, B_z]$. V tomto vektoru dat jsou uvedeny jednak souřadnice každého bodu v prostoru a jednak hodnoty vektorů síly a magnetické indukce pro uvedené body. Na následujícím obrázku můžeme pozorovat výsledek vizualizace:

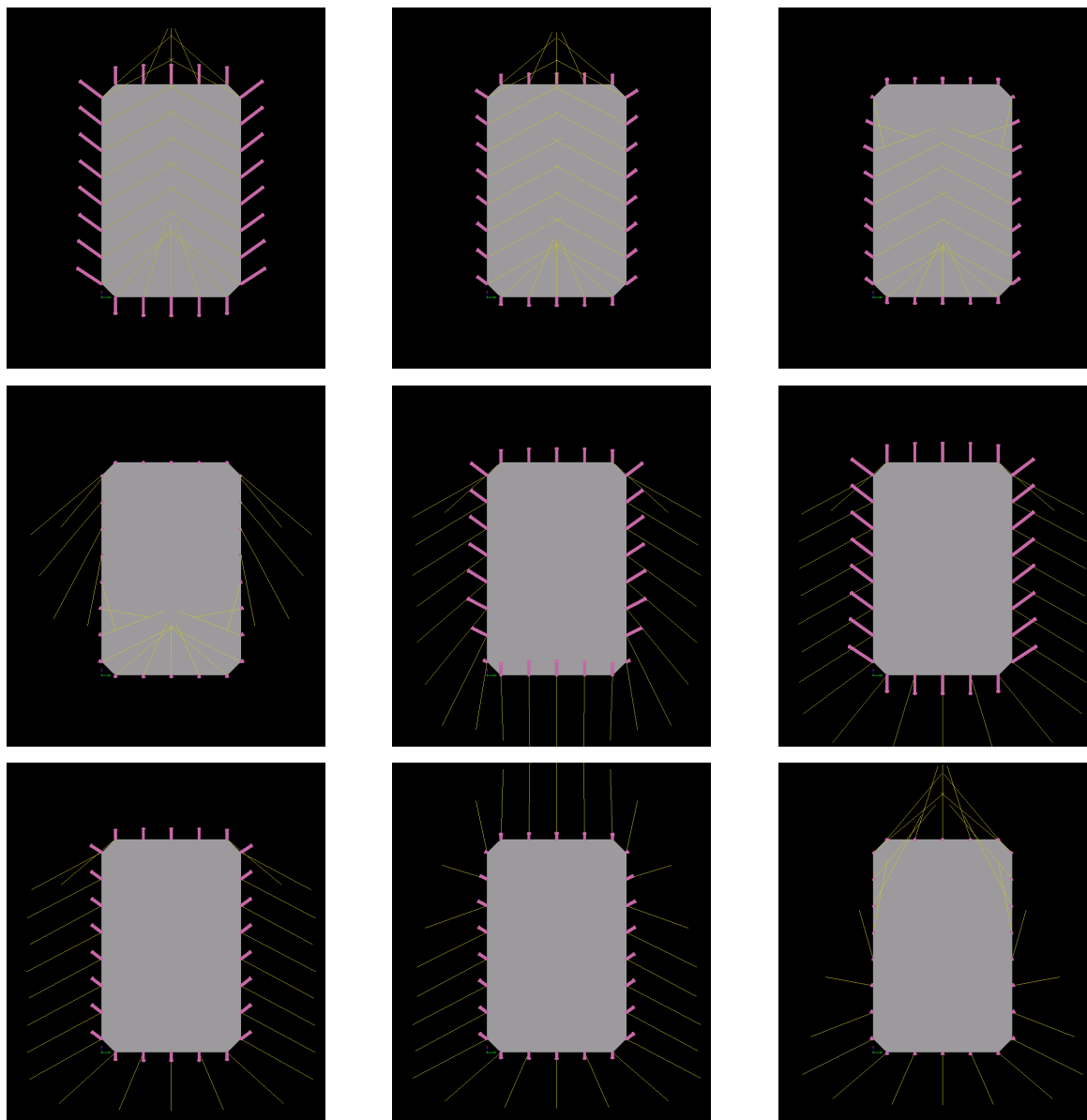


Obrázek 5.4: Vizualizace systému EDL pro železniční dopravu

A – pohled XY, B – pohled XY, C – 3D náhled, D – drátěný 3D náhled

Na obrázku 5.4 jsou zobrazeny čtyři náhledy na vstupní data. Červenou barvou jsou vyvedeny vektory síly a žlutou barvou jsou zobrazeny vektory magnetické indukce. Šedě je zobrazena triangularizovaná plocha pro souřadnice bodů v prostoru.

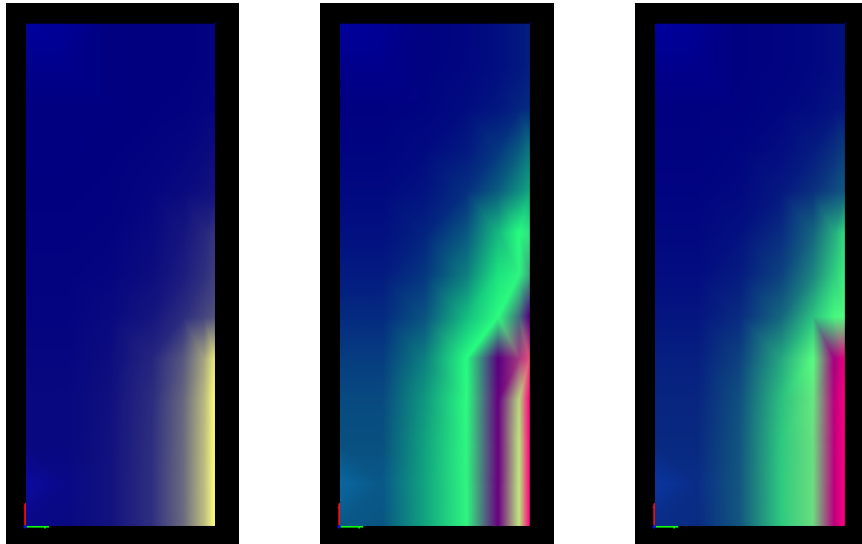
Další částí úlohy bylo zobrazení těchto vektorů v časové závislosti. Na dalším obrázku můžeme vidět postupnou animaci uvedených vektorů v 3D stínovaném náhledu.



Obrázek 5.5: Vizualizace systému EDL pro železniční dopravu - animace

Tato úloha nevyžaduje pro vizualizaci výpočetní část aplikace (vizualizace pomocí rotační symetrie).

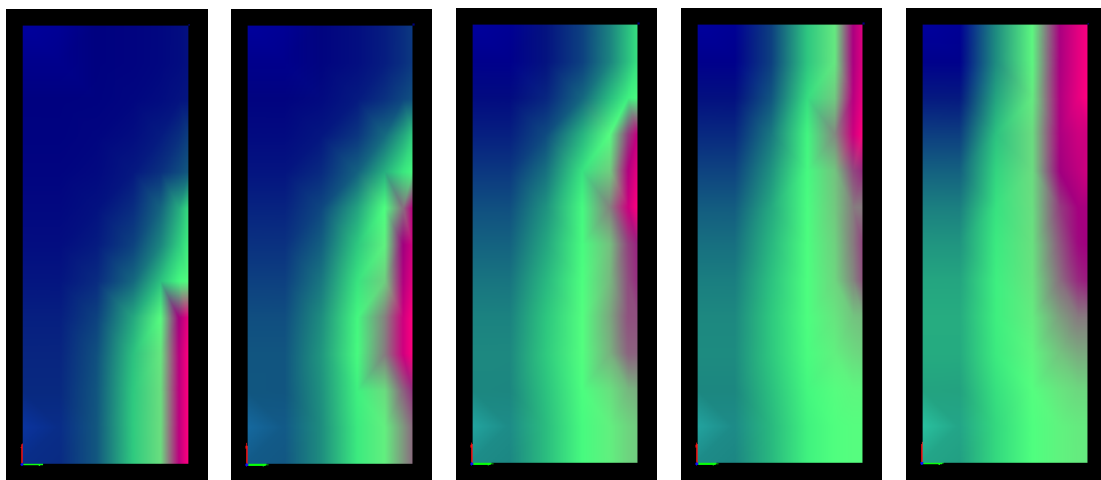
Další úlohou pro katedru teoretické elektrotechniky byla vizualizace posuvného indukčního ohřevu – posuvné kalení [KTE02b]. Tato úloha využívá vizualizaci pomocí rotační symetrie. Na dalším obrázku můžeme vidět zobrazení vstupních dat:



Obrázek 5.6: Vizualizace postupného kalení – použití různých barevných zobrazení pro stejná data

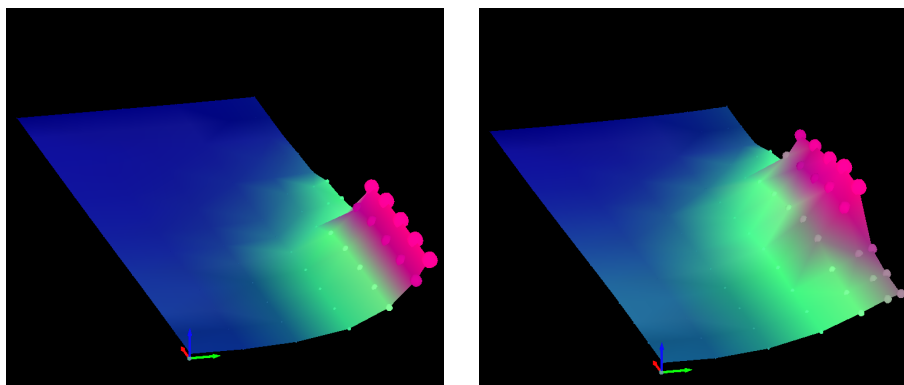
Jak můžeme pozorovat z obrázku 5.6 vizualizace pomocí více barev může být správně podána, jestliže je síť dostatečně hustá. Správné vizualizace můžeme vždy docílit použitím aproximace dvou barev (Obr. 5.6 – vlevo). Uprostřed na obr. 5.6 můžeme pozorovat zhoršující se aproximaci mezi několika barvami, vpravo je vizualizace správná, ale nutnou podmínkou je dostatečně hustá síť, jak již bylo uvedeno.

Na obrázku 5.7 můžeme pozorovat průběh teploty čase:



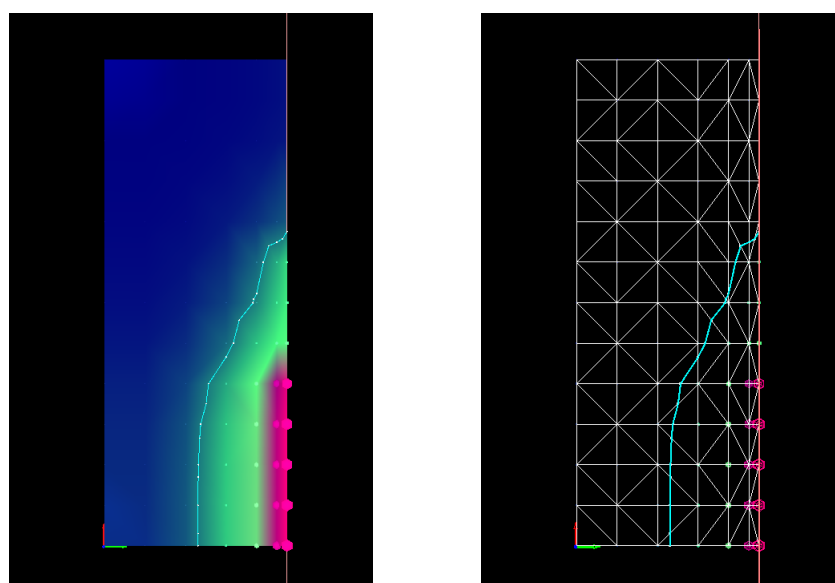
Obrázek 5.7: Vizualizace postupného kalení – časový průběh

Takto uvedené zobrazení můžeme získat pouhým mapováním barevné složky na síť. Použijeme-li další techniky jako je velikost z-ové složky sítě, velikost a barva glyfu, docílíme následující vizualizace (viz obr. 5.8):



Obrázek 5.8: Vizualizace postupného kalení – použití glyfů a z-ové složky pro reprezentaci teploty (dva časové okamžiky ve 3D zobrazení)

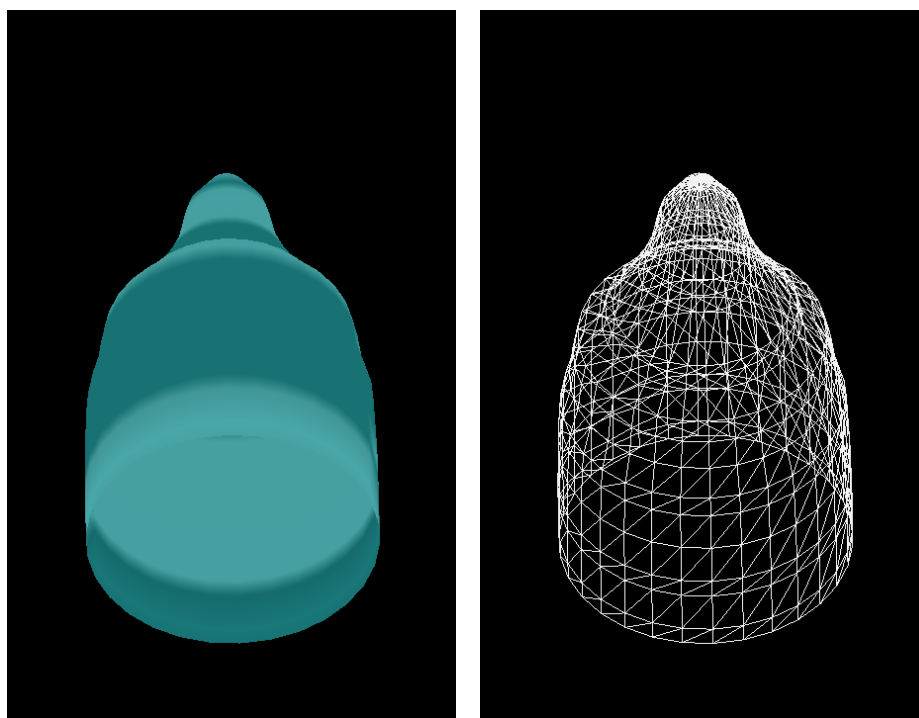
Hodnoty teplot jsou v rozmezí 30 – 700 °C. Když zvolíme z-ovou složku jako teplotu (viz obr. 5.8), spustíme výpočet s hodnotou 100 °C a nastavíme příslušnou osu rotace, dostaneme výsledky na obr. 5.9. a 5.10.



Obrázek 5.9: Vizualizace postupného kalení – zvolená teplota 100 °C
(vlevo – stínovaný model, vpravo – drátěný model)

Na obrázku 5.9 můžeme vidět nalezenou isočáru splňující kritérium 100 °C. Dále můžeme pozorovat osu rotace (červenou barvou), podle které budeme danou isočáru rotovat.

Rotační těleso můžeme vidět na obr. 5.10, přičemž barva tělesa je použita ze vstupní vizualizace na obr. 5.9.



Obrázek 5.10: Vizualizace postupného kalení – zvolená teplota 100 °C
– rotační těleso (vlevo – stínovaný model, vpravo – drátěný model)

6 Závěr a zhodnocení

Diplomová práce dává jednak ucelený pohled na vizualizaci N-D dat, dále pak některé z uvedených technik využívá při vlastní vizualizaci. Aplikace představuje jednoduchý systém pro zobrazování průmyslových dat. Systém je navržený co nejobecněji tak, aby zobrazení libovolných N-D dat nepředstavovalo žádný problém. Program může být využíván i bez použití uvedených technik pro zobrazení isočar a isopovrchů jednoduchým způsobem pro vizualizaci dat z různých oblastí. Další výhodou zobrazení je možnost vizualizace až 12 dimenzí v jediném 3D zobrazení bez větších ztrát přehlednosti a srozumitelnosti výstupu.

6.1 *Budoucí práce*

Dalším krokem výzkumu by se mohla stát vizualizace pomocí více prvků, popř. ucelení vizualizačních nástrojů do jednodušší struktury. Touto strukturou je míněna vícenásobná použitelnost jednotlivých komponent vizualizace, což by mohlo tvořit komplexnější nástroj pro vizualizaci. Mezi další kroky zkoumání by mohlo patřit rozpracování vizualizace pomocí více barev. Tím je míněno buď texturování, popřípadě jiná technika pro aproximaci barev.

Literatura

[Bro92] Brodlie K.W., Carpenter L.A., Earnshaw R.A., Gallop J.R., Hubbard R.J., Mumford A.M., Osland C.D., Quarendon P.: Scientific Visualization - Techniques and Applications, Springer-Verlag, 1992

[AVS] Micro AVS 4.0

http://www.kgt.co.jp/avs/index_e.html

[DataMin99] Morgan W., Chapple T.- A report on Data Mining and Data visualisation, 1999

http://www.cs.aue.auc.dk/datamining/tutorials/vp_report.pdf

[Zar98] Žára, J., Beneš, B., Felkel, P.: Moderní počítačová grafika, Computer Press, 1998.

[SV] <http://www.nat.vu.nl/~pwgroen/scivis/>

- Scientific Visualization

[MVT] <http://www.man.ac.uk/MVC/students/summer95/ieuan/MultidimVis/>

- Multidimensional Visualization Technique

[GTMath] <http://www.math.gatech.edu/~bourbaki/math2601/pdf/rotation.pdf>

- Georgia Tech - School of Mathematics

[Skala92] Skala, V.: Algoritmy počítačové grafiky II, Ediční středisko ZČU, 1992.

[Rou00] Roušal, M.: Prostředí pro modulární vizualizaci dat (MVE), Diplomová práce, Západočeská univerzita, Plzeň 2000.

[Kolinger02] <http://herakles.zcu.cz/research/triangulation/index2.php>

- Kolingerová, I.: Triangulations and tetrahedronizations, 2002

[KTE02a] Ulrych, B., Mayer, D.: Elektrodynamické síly v trakčním magneticky levitovaném systému, Internatinal XIII Symposium on Micromachines & Servodrives, Krasieczyn, 2002

[KTE02b] Ulrych, B., Mayer, D., Barglick, J.: Posuvný indukční ohřev - posuvné kalení -
článek na konferenci Kompei'02, 2002

[SK01] Skala, V., Kuchar, M.: The Hash Function and the Principle of Duality, Computer
Graphics International 2001, Honk Kong, China, IEEE Proceedings, ISSN 1530-1052,
pp.167-174, 2001.