

Tvorba vědeckých publikací a dokumentů

Pavel Strachota

FJFI ČVUT v Praze

17. února 2017

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

Vědecké dokumenty

Typy dokumentů:

- odborné články, zprávy (reports)
- rozsáhlejší texty (bakalářské, diplomové, dizertační práce)
- prezentace (výuka, přednášky na konferencích, obhajoba studentské práce)
- postery („plakáty“ prezentované na konferencích nebo za účelem propagace)

Typy obsahu:

- matematické vzorce
- tabulky
- diagramy, schémata
- grafy, vědecká vizualizace
- animace, video (v prezentacích)

Co zde probírat...

... **nebudeme:**

- editory typu WYSIWYG (What You See Is What You Get)
 - MS Office - Word, Excel, PowerPoint...
 - OpenOffice
- komerční software pro tvorbu diagramů a vektorové grafiky
 - MS Visio
 - Corel Draw atp.

... **budeme:**

- volně dostupné technologie pro tvorbu a ukládání vědeckých dokumentů

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů**
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

PostScript

- formát určený pro přímé zpracování tiskárnou (PostScriptové tiskárny, obvykle laserové)
⇒ RIP (Raster Image Processor)
- jedná se o **programovací jazyk**
⇒ soubor typu PostScript (.ps) je vlastně zdrojový kód v **jazyce** PostScript
- vznik v r. 1982, vyvíjí **Adobe Systems**. 3 vylepšené verze:
 - 1984: PostScript Level 1
 - 1991: PostScript Level 2
 - 1997: PostScript 3
- softwarové implementace PostScriptového RIP na PC: Adobe PostScript, GhostScript (open source), ...

Jazyk PostScript

- jazyk interpretující složení funkcí pomocí zřetězení (konkatenace) jejich jmen
- používá **zásobník** operandů a zpětnou (postfixovou) notaci
- datové typy dynamické (přidělené proměnným dle potřeby) - integer, real, double, boolean, array
- operátory: aritmetické (add, sub, mul, div, ...), grafické (moveto, lineto, setwidth, ...), zásobníkové (manipulují se zásobníkem)
- konstrukce pro definici proměnných, podmínky, cykly
- systémový a uživatelský slovník: přiřazení jmen operátorů jejich funkcím
- uživatelský slovník umožňuje definovat vlastní operátory, má přednost před systémovým

Jazyk PostScript

Použití zásobníku

$$4 + 7/3$$

```
7 3 div 4 add
```

vstup:	7	3	div	4	add
zásobník	7	3	2.333	4	6.333
		7		2.333	

- objekty na zásobníku: čísla, řetězce, pole, slovníky, ...

Jazyk PostScript

Použití zásobníku

$$4 + 7/3$$

```
7 3 div 4 add
```

vstup:	4	7	3	div	add
zásobník	4	7	3	2.333	6.333
		4	7	4	
			4		

- objekty na zásobníku: čísla, řetězce, pole, slovníky, ...

Jazyk PostScript

Grafické operátory

- manipulují se 3 základními druhy dat
 - text (pozice, orientace, velikost)
 - geometrické objekty
 - bitmapové obrázky
- souřadnicový systém **tiskových bodů**, nezávislý na výstupním zařízení
- zásobník grafického stavu (aktivní barva, parametry „pera“)
- zásobník **aktuální cesty** (množina na sebe navazujících bodů spojených úsečkami či křivkami)
- cestu lze předem připravit (`lineto`, `moveto`), poté vykreslit (`stroke`) nebo vyplnit (`fill`)
- možnosti definice rozměru stránky, vykreslení **aktuální stránky** (`showpage`) a přechodu na novou stránku
- ...

Jazyk PostScript

Hello world

```
%!PS
/Courier findfont
20 scalefont
setfont
72 500 moveto
(Hello world!) show
showpage
```

Dokument PostScript

Fonty 1/2

- dokument obsahuje definici fontů, které se v něm používají
- **vektorové škálovatelné fonty** - sady *glyfů* (tvarů znaků) stejného stylu, popsanych pomocí křivek
 - *outline font* - popis pomocí obrysů a výplní
 - *stroke-based font* - popis pohybu pomyslného polygonálního (resp. eliptického) pera podél křivky
 - vhodné pro asijská znaková písma.
- *font hinting* - doplňkové informace o rasterizaci fontů v malém měřítku, kde nedostatečné rozlišení rastru vede k subjektivní deformaci tvarů
- *kerning* (proložení znaků) - dodatečný popis mezer (roztečí) mezi dvojicemi znaků ve fontu s proměnlivou šířkou (*proportional font*)

Dokument PostScript

Fonty 2/2

- **Adobe Type 1** fonty - definovány Bézierovými kubikami (viz [POGR2](#))
- **True Type** fonty - původně od Apple, Inc., popis kvadratickými Bézierovými křivkami (v PostScriptu tzv. *Type 42*)
- **OpenType** - Adobe + Microsoft - sjednocení Type 1 a True Type

Vlastnosti fontů

Font hinting

abcfgop AO *abcfgop*

abcfgop AO *abcfgop*

abcfgop

abcfgop

Vlastnosti fontů

Proporcionální fonty a fonty s pevnou šířkou

Proportional

Monospaced

Vlastnosti fontů

Typy fontů (*typeface*)

Serif (Minion Pro)

- Old Style (Adobe Jenson Pro)
- Transitional (ITC New Baskerville)
- Modern (Bodoni)

Slab Serif (Clarendon)

Sans serif (Myriad)

Script (Coronet)

Blackletter (Teutonic No. 1)

DISPLAY (LiquidCrystal)

Monospaced (Courier)



(ITC Zapf Dingbats)

(Dingbat)

Vlastnosti fontů

Patkové a bezpatkové fonty

AaBbCc

bezpatkový font (sans serif)

AaBbCc

patkový font (serif)

Vlastnosti fontů

Patkové a bezpatkové fonty

AaBbCc

bezpatkový font (sans serif)

AaBbCc

patkový font (serif)

Vlastnosti fontů

Metrika fontu



Formát PDF 1/2

- **Portable Document Format** - od Adobe Systems, od r. 2008 otevřený standard ISO
- reprezentace dokumentu nezávisle na aplikaci, OS, výstupním zařízení \implies všude identický vzhled (oproti HTML, dokumentu Word atd.)
- implementuje podmnožinu PostScriptu (nemá příkazy pro kontrolu běhu programu - smyčky, podmínky)
 - PostScriptový dokument lze snadno převést do PDF
- data v PDF uspořádána do objektů (datové proudy, slovníky, pole, řetězce, čísla, ...)
- na rozdíl od PostScriptu implementuje průhlednost

Formát PDF 2/2

- prohlížeč PDF na PC (v největší míře Adobe Reader) umí interpretovat rozšířené interaktivní funkce
- hypertextové odkazy (včetně odkazů do jiného souboru PDF, na internet nebo na externí aplikaci)
- efekty pro přechod mezi stránkami
- automatický přechod na další stránku
- formuláře
- multimédia (video, zvuk) - za podpory dalších aplikací (RealPlayer, Windows Media Player)
 - z externího souboru nebo vestavěné do PDF
- JavaScript

Formát DVI

- *DeVice Independent file format*
- formát produkovaný typografickým systémem T_EX
- nezávislý na zařízení, jednoduchý na interpretaci (zobrazovacím zařízením)
- obsahuje pouze odkazy na obrázky, nikoliv samotná obrazová data
- standardními nástroji lze převést do PostScriptu nebo PDF

Formát Deja Vu 1/2

- přípona `.djvu` nebo `.djv`
- formát pro mimořádně efektivní kompresi naskenovaného textu
- enkodér rozdělí vstupní obraz na *popředí*, *pozadí* a *masku* (tj. dvouúrovňový obraz v podstatě specifikující průhlednost popředí)
- popředí a pozadí jsou barevné, resp. šedotónové obrazy, kódované s nižším rozlišením waveletovou kompresí IW44
- maska kódována pomocí JB2 (identifikace téměř identických tvarů na stránce - např. písmen)
 - tvar se kóduje jen jednou
 - pozice se kóduje pro každý výskyt
- může obsahovat vrstvu rozpoznání textu (OCR - *Optical Character Recognition*)

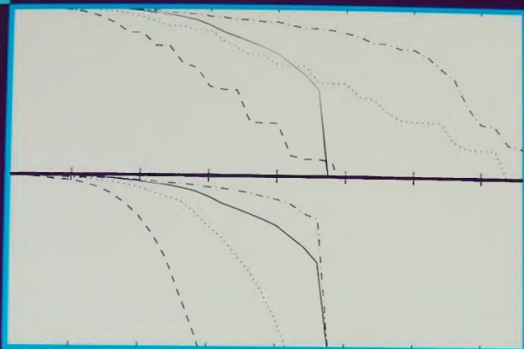
Formát Deja Vu 2/2

- typická ČB stránka má jen cca 3-40kB
- např. anglická verze Rektorysova Přehledu užití matematiky (Survey of Applicable Mathematics)
 - 1369 stránek
 - 9 639 510 bytů
 - průměrně 7kB na stránku
- vývoj od r. 1996, AT&T, později LizardTech
- komerční produkt od LizardTech: enkodér Document Express, prohlížeč DjVu Browser PlugIn
- zdroj. kód algoritmů dostupný pod GPL \implies sada nástrojů DjVU Libre (prohlížeč djview atd.)
 - djvu umí otevřít i Evince, Okular
- konkurence: Adobe *ClearScan* (od Acrobat verze 9)

Deja Vu

Příklad

Iterative Methods for Solving Linear Systems



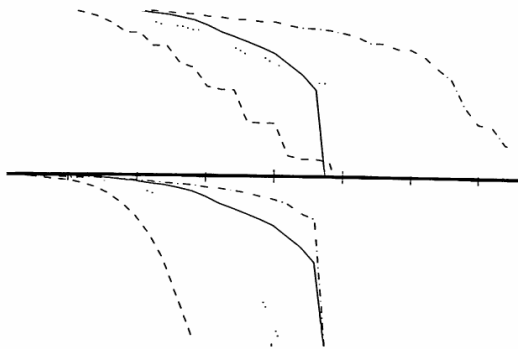
Anne Greenbaum

komplet

Deja Vu

Příklad

Iterative Methods for Solving Linear Systems



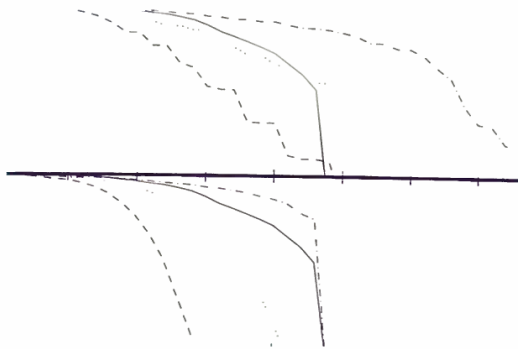
maska

Anne Greenbaum

Deja Vu

Příklad

Iterative Methods for Solving Linear Systems

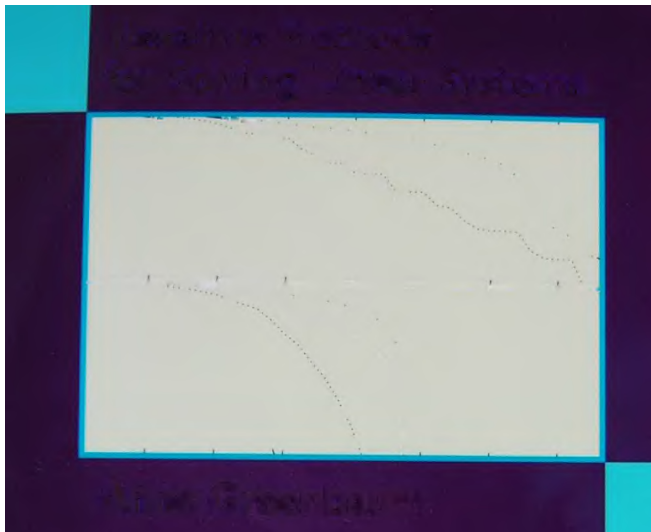


popředí

Anne Greenbaum

Deja Vu

Příklad



pozadí

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX**
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

T_EX

- čti [tech]
- typografický systém pro vysoce kvalitní výstup
- převážně pro tvorbu vědeckých dokumentů, sazbu **matematických vzorců**
- **programovací jazyk** založený na možnosti definice **maker**
- navržený **Donaldem Erwinem Knuthem**, počátky v r. 1978
- několikrát přepsán do jiného programovacího jazyka
- od roku 1989 verze 3, povoleny už jen opravy chyb
- verze se asymptoticky blíží π přidáváním dalších cifer
- současná verze je 3.1415926
- po smrti D. E. Knutha bude verze přejmenována na π , všechny zbývající chyby (*bugs*) se stanou vlastnostmi programu (*features*)

L^AT_EX

- čti [lejtech], [latech]
- sada maker pro T_EX původem od Leslieho Lamporta - vznik v r. 1984
- z T_EXu je uživatelsky příjemnější **značkovací jazyk** (*markup language*) pro tvorbu dokumentů
- **distribuce L^AT_EXu** (např. MikT_EX, T_EXLive) je systém **balíčků** od různých autorů, definujících
 - styly dokumentů (kniha, článek, dopis, ale např. i šachový zápis atd.)
 - pokročilé funkce (práci s obrázky, hypertextovými odkazy, formátování, ...)

myšlenka:

- zvolíte si styl a v L^AT_EXu napíšete, co je podstatné (obsah, členění do kapitol, odstavců atd.)
- L^AT_EX se postará o vše ostatní (formátování, číslování stránek, kapitol, vzorců, generování obsahu, indexu atd.)

L^AT_EX

Ukázkový kód

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system for the \TeX{}
  typesetting program. It offers programmable desktop publishing
  features and extensive facilities for automating most aspects of
  typesetting and desktop publishing, including numbering and
  cross-referencing, tables and figures, page layout, bibliographies,
  and much more. \LaTeX{} was originally written in 1984 by Leslie
  Lamport and has become the dominant method for using \TeX; few
  people write in plain \TeX{} anymore. The current version is
  \LaTeXe.

  % This is a comment; it is not shown in the final output.
  % The following shows a little of the typesetting power of LaTeX
  \begin{align}
    E &= mc^2 && \\
    m &= \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{align}
\end{document}
```

L^AT_EX

Výsledek

L^AT_EX

L^AT_EX is a document preparation system for the T_EX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. L^AT_EX was originally written in 1984 by Leslie Lamport and has become the dominant method for using T_EX; few people write in plain T_EX anymore. The current version is L^AT_EX 2_ε.

$$E = mc^2 \tag{1}$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2}$$

L^AT_EX

Ukázky matematických vzorců

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

L^AT_EX

Ukázky matematických vzorců

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X, Y) =
\frac{\displaystyle
\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}
{\displaystyle\biggl[
\sum_{i=1}^n (x_i - \overline{x})^2
\sum_{i=1}^n (y_i - \overline{y})^2
\biggr]^{1/2}}
\end{displaymath}
```

LyX

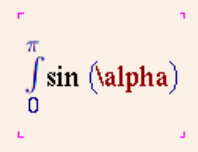
- pamatovat si všechny zkratky a funkce je obtížné \implies spousta editorů s GUI na vkládání L^AT_EXových příkazů (např. oblíbený **WinEdt**)

LyX: tzv. WYSIWYM (What You See Is What You Mean) editor

- front-end pro L^AT_EX (generuje L^AT_EXový kód)
- nezobrazuje přesnou podobu výstupu, ale pro představu to stačí
- umožňuje psát matematické vzorce jak pomocí L^AT_EXových zkratk, tak pomocí dialogových oken nebo klávesových zkratk (asi nejrychlejší)
- všude navigace pomocí myši
- jednoduše umožňuje navrhovat tabulky, vkládat obrázky
- vlastnosti, které v LyXu chybí, lze „doprogramovat“ vložením kódu L^AT_EXu

LyX

Zápis matematického výrazu


$$\int_0^{\pi} \sin(\alpha)$$

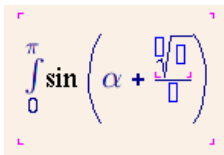
LyX

Zápis matematického výrazu

$$\int_0^{\pi} \sin \left(\alpha + \frac{\pi}{2} \right)$$

LyX

Zápis matematického výrazu

A screenshot of a LyX window showing a mathematical expression. The expression is $\int_0^{\pi} \sin \left(\alpha + \frac{\sqrt{\square}}{\square} \right)$. The expression is displayed in a serif font. The integral sign is large and bold. The sine function is in a smaller font. The argument of the sine function is enclosed in large parentheses. The variable α is in a serif font. The fraction is also in a serif font. The numerator is the square root of a square symbol, and the denominator is another square symbol. The entire expression is centered on a light beige background. There are small red corner markers at the corners of the beige area.

LyX

Zápis matematického výrazu

$$\int_0^{\pi} \sin \left(\alpha + \frac{\sqrt[3]{x^2 + 1}}{x + \sum \beta_i} \right) dx$$

LyX

Zápis matematického výrazu

Mathematical Model of Anisotropic Diffusion

Consider the time interval $\mathcal{J} = (0, T)$, the domain $\Omega \subset \mathbb{R}^3$ in the form of a block and the diffusion tensor $\tilde{D}: \bar{\Omega} \mapsto \mathbb{R}^{3 \times 3}$ representing the input data. The initial boundary value problem for the Allen-Cahn diffusion equation reads

Box (Minipage, double frame)

$$\xi \frac{\partial p}{\partial t} = \xi \nabla \cdot \tilde{D} \nabla p + \frac{1}{\xi} f_0(p) \quad \text{in } \mathcal{J} \times \Omega, \quad (1, \text{eq:Allen-Cahn})$$

$$p = 0 \quad \text{on } \mathcal{J} \times \partial\Omega, \quad (2, \text{eq:boundary-condition})$$

$$p = p_{ini} \quad \text{in } \Omega, \quad (3, \text{eq:initial-condition})$$

Vertical Space (-3mm)

known function $p: \bar{\mathcal{J}} \times \bar{\Omega} \rightarrow \mathbb{R}$ interpreted as the texture intensity, p_{ini} is the initial value and $f_0(p) = p(1-p)(p - \frac{1}{2})$ acts efficiently as a contrast increasing term. The diffusion tensor \tilde{D} and the initial time t_0 and the final time T are chosen appropriately.



Snímky obrazovky

ANALYSIS OF THE PARALLEL FINITE VOLUME SOLVER FOR THE ANISOTROPIC ALLEN-CAHN EQUATION IN 3D

`\noindent` `%%\thispagestyle{...}`

PAVEL STRACHOTA `\footnote`, MICHAL BENES `\ref{FNSPE}`, MARCO GROTTADAUREA `\footnote`,
JAROSLAV TINTĚRA `\footnote`

Comment: General purpose macros

Macro: dd: `\dd` Macro: e: `\e` Macro: Tr: `\Tr` Macro: T: `\T`

Comment: Number set and function space macros

Macro: Cn: `\Cn` Macro: N: `\N` Macro: Q: `\Q` Macro: R: `\R` Macro: Z: `\Z` Macro: J: `\J`

Macro: X: `\X` Macro: Hh: `\mathcal{H}_h`

Comment: Finite volume scheme formalism

Macro: sumk: `\sum_{K \in \mathcal{T}}` Macro: sums: `\sum_{\sigma \in \mathcal{E}}` Macro: sumsk: `\sum_{\sigma \in \mathcal{E}_K}` Macro: hnorm: `\| \cdot \|_{1, \mathcal{T}}`

Macro: hnormsq: `\| \cdot \|_{1, \mathcal{T}}^2`

ABSTRACT. In this contribution, a parallel implementation of the finite volume solver is introduced, designated to numerically solve the initial boundary value problem for the Allen-Cahn equation with anisotropy on large 3D grids. The choice of a suitable numerical scheme is discussed and its convergence properties are investigated by means of evaluation of the experimental order of convergence. Afterwards, the consequent limitations for the theoretical error estimate are pointed out. Furthermore, the results of parallel algorithm efficiency measurements are shown, based on extensive tests performed on high performance computing systems. The final part gives a brief overview of a magnetic resonance tractography (neural tract tracking and visualization) method consisting in the solution of the above problem.

LyX

Snímky obrazovky

Let $\Omega \subset \mathbb{R}^d$ be a polygonal domain and $T > 0$. Denote by \mathcal{T} an *admissible* [FVM-Eymard] mesh defined on Ω , let $k \in (0, T)$ and $N_k = \{n \in \mathbb{N} | nk \leq T\}$.

Furthermore, for all $K \in \mathcal{T}$, denote by p_K^n the value obtained by numerical solution of [Ref: eq:FVM-semidiscrete-scheme] approximating $p(x_K, nk)$ where $x_K \in K$. The pointwise error is then given by

$$e_K^n = p(x_K, t_n) - p_K^n.$$

for all $K \in \mathcal{T}$ and $n \in N_k$. Assuming $p \in C^2(\bar{\Omega} \times \mathcal{J})$, $l \in C^2(\bar{\Omega}, \mathbb{R})$, $\tilde{D} = \tilde{l}$, and using the Dirichlet boundary condition $|p|_{\partial\Omega} = g$, $g \in C(\partial\Omega \times (0, T))$ instead of ([Ref: eq:boundary-condition-D]), there exist positive constants C and k_0 depending only on u , Ω , T , and ξ such that

$$\sqrt{\sum_{K \in \mathcal{T}} (e_K^n)^2} m(K) \leq C(h+k), \quad \forall n \in N_k$$

provided that $k \leq k_0$.

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků**
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

L^AT_EX a obrázky

- **latex** generuje DVI a akceptuje obrázky typu `.eps`
 - EPS (*Encapsulated PostScript*) - speciální tvar souboru v PostScriptu
 - obsahuje parametr `BoundingBox` určující rozsah výstupu (nezobrazuje se jako celá stránka, ale jen jako výřez)
 - konverze do PDF obvykle:

$$\text{DVI} \xrightarrow{\text{dvips}} \text{PS} \xrightarrow{\text{ps2pdf}} \text{PDF}$$

$$\text{DVI} \xrightarrow{\text{dvipdfm}} \text{PDF}$$

- **pdflatex** generuje přímo PDF a akceptuje obrázky typu `.pdf`, `.jpg`, `.png`
- obrázky tvořené přímo L^AT_EXovým kódem - `.pictex`
 - zobrazují se i v souboru DVI
 - složitější konstrukce dělají problémy (vyplňování obrázků apod.)

XFig, JFig

- **XFig** - klasický jednoduchý editor vektorové grafiky pro X11
- ne zcela intuitivní ovládání pro uživatele např. Corel Draw, ale dá se zvyknout
- **jfig** - program formátovému výstupu i ovládáním kompatibilní s XFig, napsaný v Javě (shareware)
 - v některých ohledech šikovnější (a funguje i ve Windows)
- formát `.fig` - ASCII, ale pro člověka nepřehledný (jen čísla)

výhody:

- na každé linuxové distribuci
- konzolový program `fig2dev` umí exportovat `.fig` do mnoha formátů: `.eps`, `.pdf`, `.png`, `.gif`, `.jpeg`, `.pictex`, `.svg`, ...

Integrace L^AT_EXových vzorců do obrázku

- export `schema.fig` do dvojice souborů `schema.pstex` + `schema.pstex_t`
- soubor `.pstex` je vlastně EPS z obrázku vzniklého odebráním všech textových řetězců
- soubor `.pstex_t` je kód L^AT_EXu (obsahující odkaz na soubor `.pstex`), který vložíte do svého textu např. pomocí příkazu

```
\input{schema.pstex_t}
```

⇒ v textových řetězcích lze používat makra definovaná v hlavním dokumentu

⇒ styl písma bude stejný jako v hlavním dokumentu atd.

Inkscape

- pokročilý vektorový editor k práci s formátem SVG
- vkládání a editace křivek, tvarů
- množinové operace s tvary (průnik, sjednocení, rozdíl)
- barvy, barevné přechody, průhlednost, vyplňování
- vrstvy
- vkládání textu
- konverze bitmapy na křivky (trasování)
- export do mnoha vektorových i bitmapových formátů, do .eps i .pdf

Inkscape a L^AT_EX

- 1 rendering textu L^AT_EXem: Rozšíření ▸ Vykreslit ▸ L^AT_EX

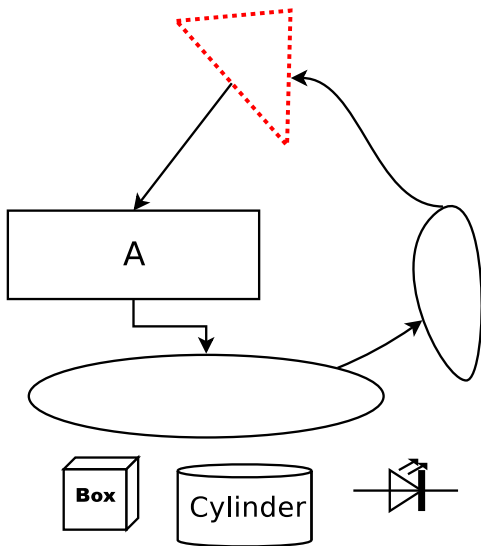
$$\frac{\pi^2}{6} = \lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2}$$

- 2 export do PDF a zaškrtnout **PDF+L^AT_EX**
 - vzniknou soubory .pdf a .pdf_tex , tj. kód L^AT_EXu (obsahující odkaz na soubor .pdf), který vložíte do textu pomocí příkazu

```
\input{schema.pdf_tex}
```

Dia

- obdoba Microsoft Visio pro Linux/GNOME
- tvorba schémat a diagramů
- automatické napojování „bublin“
- export do .eps, .pdf



Ipe

- geometrické tvary, křivky, vyplňování, vrstvy
- text a matematika v L^AT_EXu, výsledek je hned vidět
 - v *stylesheets* i preambule L^AT_EX dokumentu \implies vlastní makra atd.
- přichytávání (*snapping*) k síti, k vrcholům objektů, k obrysu
 - možno měnit nastavení přichytávání během kreslení
- export do .eps, .pdf, .png, vícestránkové dokumenty

Ipe GUI

The screenshot displays the Ipe GUI interface with a drawing on a grid. The drawing includes:

- A red-bordered box containing the text: *Text a matematika $x^2 + \sum_{n=1}^{10} \sqrt{n}$*
- An orange oval shape.
- A dashed purple rectangle.
- A dashed green box containing the text: *text v LaTeXu*
- A black stepped line with an arrow pointing to the orange oval.
- A dashed green box containing a list item: *• Celý odstavec včetně rovnice* followed by the equation $a^2 + b^2 = c^2$ and the text *a. tak dále.*

The interface includes a menu bar (File, Edit, Properties, Snap, Mode, Zoom, Layers, Views, Pages, Ipelets, Help), a toolbar with various drawing tools, a Properties panel on the left, and a Layers panel at the bottom left. The status bar at the bottom right shows coordinates (610, 840, 801, 822) and resolution (88ppi).

Asymptote

- interpretovaný programovací jazyk se syntaxí podobnou C++ pro kreslení:
 - schémat
 - diagramů
 - grafů funkcí
 - 3D objektů
 - animací
- přímo zpracovává L^AT_EXové vzorce
- naopak: obsahuje L^AT_EXový balíček umožňující vkládat kód Asymptote přímo do zdrojového dokumentu L^AT_EXu
- výsledek exportuje do .eps, .pdf, .gif (pro animace) atd.
- stále se rychle vyvíjí a rozšiřuje (v roce 2009 verze 1.90)

Asymptote

Příklad 1

```
import graph;
size(200,IgnoreAspect);

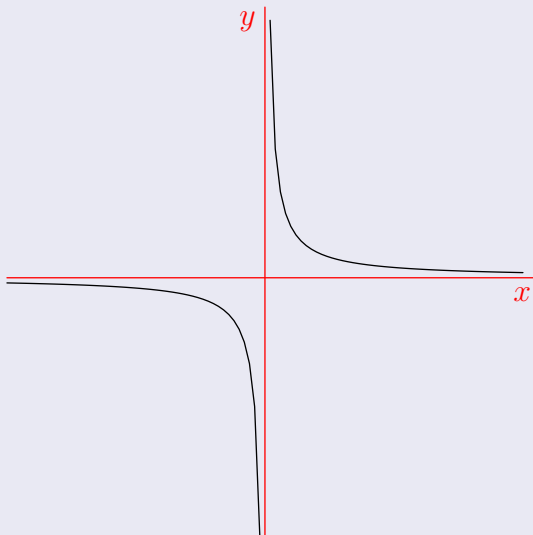
real f(real x) {return 1/x;};

bool3 branch(real x)
{
    static int lastsign=0;
    if(x == 0) return false;
    int sign=sgn(x);
    bool b=lastsign == 0 || sign == lastsign;
    lastsign=sign;
    return b ? true : default;
}

draw(graph(f, -1, 1, branch));
axes("$x$", "$y$", red);
```


Asymptote

Příklad 1



Asymptote

Příklad 2

```
import graph;
size(0,100);

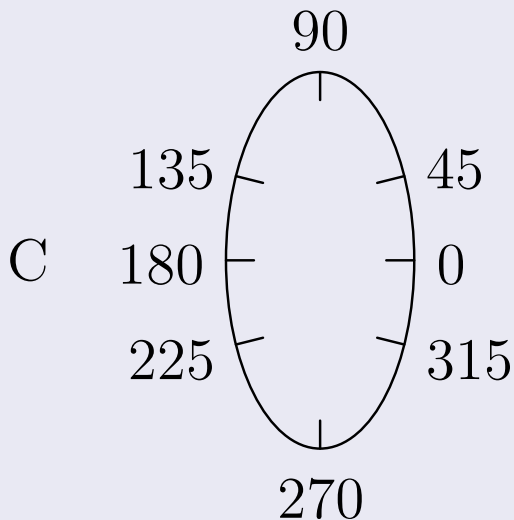
path g=ellipse((0,0),1,2);

scale(true);

axis(Label("C",align=10W),g,LeftTicks(endlabel=false,8,end=false),
      ticklocate(0,360,new real(real v) {
        path h=(0,0)--max(abs(max(g)),abs(min(g)))*dir(v);
        return intersect(g,h)[0];}));
```

Asymptote

Příklad 2



Asymptote

Příklad 3

```
import graph;
import palette;
import contour;

size(200);

int n=100;

real[] x=new real[n];
real[] y=new real[n];
real[] f=new real[n];

real F(real a, real b) {return a^2+b^2;}

real r() {return 1.1*(rand()/randMax*2-1);}

for(int i=0; i < n; ++i) {
    x[i]=r();
    y[i]=r();
    f[i]=F(x[i],y[i]);
}

pen Tickpen=black;
pen tickpen=gray+0.5*linewidth(currentpen);
pen[] Palette=BWRainbow();

bounds range=image(x,y,f,Range(0,2),Palette);
draw(contour(pairs(x,y),f,new real[] {0.25,0.5,1},operator ..));

palette("$f(x,y)$", range,point(NW)+(0,0.5),point(NE)+(0,0.8),Top,Palette,
    PaletteTicks(Tickpen,tickpen));
```

Asymptote

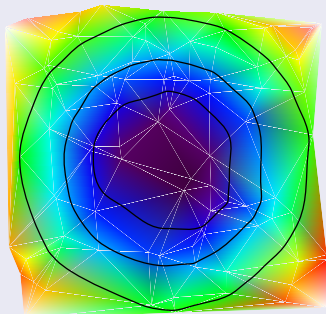
Příklad 3

$$f(x, y)$$

0

1

2



Asymptote

Příklad 4

```
import graph3;
import grid3;
import palette;

currentprojection=orthographic(0.8,1,2);
size(400,300,IgnoreAspect);

real f(pair z) {return cos(2*pi*z.x)*sin(2*pi*z.y);}

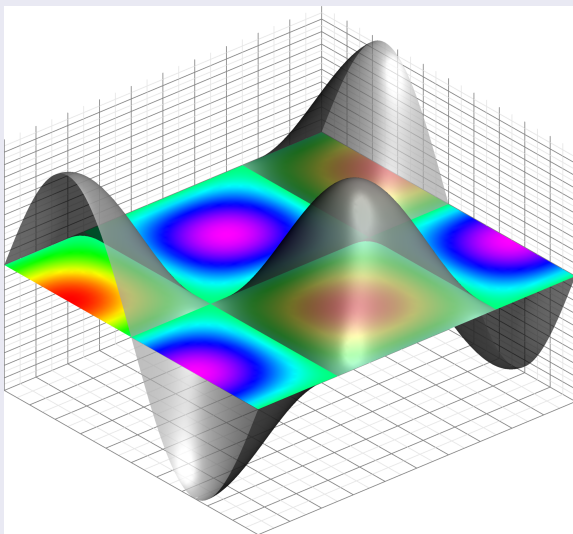
surface s=surface(f, (-1/2, -1/2), (1/2, 1/2), 50, Spline);

surface S=planeproject(unitsquare3)*s;
S.colors(palette(s.map(zpart), Rainbow()));
draw(S, nolight);
draw(s, lightgray+opacity(0.7));


grid3(XYZgrid);
```

Asymptote

Příklad 4



TikZ / PGF

- systém pro programování grafiky přímo v kódu L^AT_EXu
 - "*TikZ ist kein Zeichenprogramm*"
- rozsáhlé možnosti, dokumentace verze 3 má přes 1000 stránek
- podobně jako Asymptote podporuje programovací konstrukce
 - podmínky, cykly, proměnné
- programování trochu netradiční (vše je v rámci T_EXu)
- autor: **Till Tantau**, profesor na univerzitě v Lübecku
- `\tikz \draw (0pt,0pt) --(20pt,6pt);` \implies 

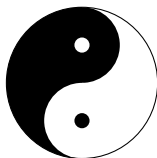
TikZ

Příklad 1 - kód

```
1 \begin{tikzpicture}
2
3   %color one half of a unit circle
4   \begin{scope}
5     \clip (0,0) circle (1cm);
6     \fill[black] (0cm,1cm) rectangle (-1cm, -1cm);
7   \end{scope}
8   %fill heads
9   \fill[black] (0,0.5) circle (0.5cm);
10  \fill[white] (0,-0.5) circle (0.5cm);
11  %fill eyes
12  \fill[white] (0,0.5) circle (0.1cm);
13  \fill[black] (0,-0.5) circle (0.1cm);
14  %outer line
15  \draw (0,0) circle (1cm);
16
17 \end{tikzpicture}
```

TikZ

Příklad 1 - výsledek



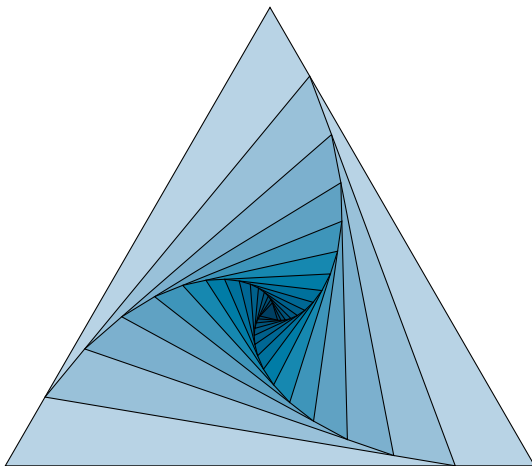
TikZ

Příklad 2 - kód

```
1 \begin{tikzpicture}
2   \coordinate (A) at (0,0);
3   \coordinate (B) at (-60:7cm);
4   \coordinate (C) at (240:7cm);
5   \foreach \density in {20,30,...,160}{%
6     \draw[fill=MidnightBlue!\density] (A)--(B)
7       --(C)--cycle;
8     \path
9       (A) coordinate (X)
10      -- (B) coordinate[pos=.15](A)
11      -- (C) coordinate[pos=.15](B)
12      -- (X) coordinate[pos=.15](C);
13 }
14 \end{tikzpicture}
```

TikZ

Příklad 2 - výsledek



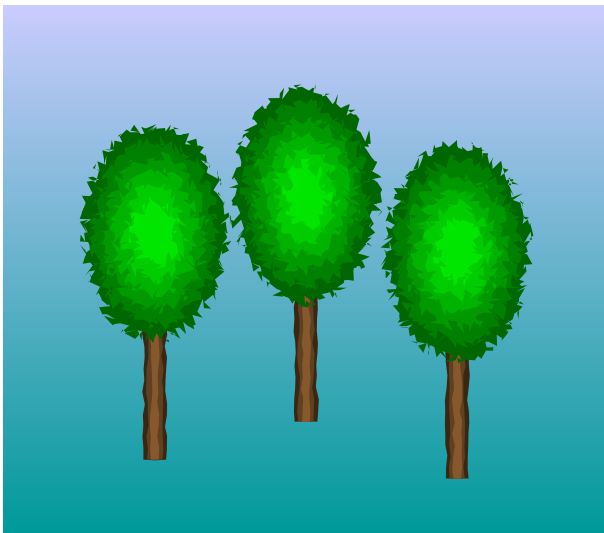
TikZ

Příklad 3 - kód

```
1 \usetikzlibrary{decorations.pathmorphing,calc,shapes,shapes.geometric,patterns}
2 \tikzset{
3   treetop/.style = {decoration={random steps, segment length=0.3mm}, decorate},
4   trunk/.style   = {decoration={random steps, segment length=1.5mm,
5     amplitude=0.2mm}, decorate}}
6 \tikzset{
7   my tree/.pic={
8     \foreach \w/\f in {0.3/30,0.2/50,0.1/70} {
9       \fill [brown!\f!black, trunk] (-\w/2,0) rectangle +(\w,3);
10    }
11    \foreach \n/\f in {1.4/40,1.2/50,1/60,0.8/70,0.6/80,0.4/90} {
12      \fill [green!\f!black, treetop](0,3) ellipse (\n/1.5 and \n);
13    }
14  }
15 }
16 \begin{tikzpicture}
17   \shade[bottom color=cyan!60!black, top color=blue!20!white] (0,0)
18     rectangle (8,7);
19   \pic at (2,1)   {my tree};
20   \pic at (4,1.5) {my tree};
21   \pic at (6,0.75) {my tree};
22 \end{tikzpicture}
```

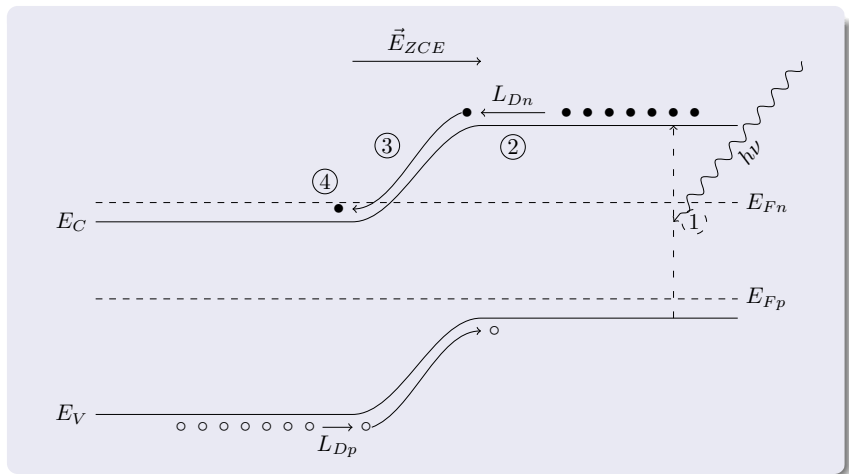
TikZ

Příklad 3 - výsledek



TikZ

Příklad 4 - komplexnější schéma



Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu**
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace

Podpora tvorby prezentací v L^AT_EXu

- balíček **Beamer** (podpora i v L^yXu) ... opět Till Tantau
- balíček **PowerDot** (dříve HA-Prosper)
- výsledek - transparenty (*slides*) v PDF (jako tyto)
- možnost využít schopnosti Adobe Readeru
 - hypertextové odkazy
 - efekty při přechodu na další slide
 - spouštění externích programů
 - animace
- různé předdefinované styly, možno upravit
- důraz na obsah, nikoliv na návrh vzhledu
- odpadá zdlouhavá a nepřesná manipulace s objekty pomocí myši

Beamer

Příklad

```
\documentclass{beamer}

\usepackage{beamerthemesplit}

\title{Example Presentation Created with the Beamer Package}
\author{Till Tantau}
\date{\today}

\begin{document}

\frame{\titlepage}

\section[Outline]{}
\frame{\tableofcontents}

\section[Introduction]
\subsection{Overview of the Beamer Class}
\frame
{
  \frametitle{Features of the Beamer Class}

  \begin{itemize}
    \item<1-> Normal LaTeX class.
    \item<2-> Easy overlays.
    \item<3-> No external programs needed.
  \end{itemize}
}
\end{document}
```

Beamer

Příklad

Example Presentation Created with the Beamer
Package

Till Tantau

December 7, 2009

Beamer

Příklad

Beamer

Příklad

Features of the Beamer Class

- ▶ Normal LaTeX class.

Beamer

Příklad

Features of the Beamer Class

- ▶ Normal LaTeX class.
- ▶ Easy overlays.

Beamer

Příklad

Features of the Beamer Class

- ▶ Normal LaTeX class.
- ▶ Easy overlays.
- ▶ No external programs needed.

Animace v prezentaci

- balíček **animate** umí animovat sekvence snímků uložené jako název-0.png až název-###.png

```
\usepackage{animate}
```

- na místě, kde má být animace, potom napsat něco jako:

```
\animategraphics[buttonfg=0.3:0.3:0.7,  
buttonsize=0.2cm,loop,width=4cm]{8  
{Animations/Adam7-interlacing/frame-}{0}{263}
```

- nefunguje jen v prezentacích, ale v libovolném textu
- potřebuje Adobe Reader (pod Windows i pod Linuxem)

Spuštění externího programu z prezentace

- balíček **hyperref** (Beamer ho používá automaticky)

```
\usepackage{animate}
```

- na místě, kde má být odkaz na aplikaci (tlačítko je funkce Beameru), potom napsat něco jako:

```
\href{run:Demo/application.exe}  
{\beamerbutton{Spustit demo}}
```

- funguje i v Linuxu
- problém pokud chceme, aby stejná prezentace chodila ve Windows i v Linuxu
 - 2 tlačítka, každé spustí jinou verzi aplikace
 - emulátor Wine, vhodně vytvořené dávkové soubory (viz moje prezentace)

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu**
- 7 Vědecká vizualizace

Poster v L^AT_EXu

- možnosti tisku na velký formát: A0,A1

balíček **a0poster**:

- nutnost formátovat text explicitně
- použití příkazů prostého T_EXu pro výpočet rozteče sloupců, velikostí rámečků atd.
- postup komplikovaný, ale nikoliv nezvládnutelný a výsledek stojí za to
- dobré nezačínat od píky, vycházet ze šablony dokumentu

balíček **beamerposter**:

- spojuje **a0poster** a možnosti (a uživatelskou příjemnost) **beameru**
- vlastně "prezentace" na jeden velký A0 slide
- z beameru lze použít barevná schémata, formátování do sloupců, bloky atd.

novější: **baposter**, **fancytikzposter**



FINITE VOLUME METHOD FOR THE ALLEN-CAHN EQUATION WITH ANISOTROPY

Pavel Strachota

Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague

E-mail: pavel.strachota@fjfi.cvut.cz

Introduction

Tractography is a process of displaying neural tracts in human brain. We have developed a visualization method using the data obtained by magnetic resonance diffusion tensor imaging (MR-DTI). The root of the method is the numerical solution of a problem for a diffusion PDE in a 3D domain. The anisotropy incorporated in the equation has to be treated carefully when designing the numerical scheme. For spatial discretization, several finite volume schemes have been implemented and their properties have been compared with respect to the rate of artificial numerical diffusion. The convergence of the scheme of our choice has been proved experimentally.

Formulation

Consider the time interval $\mathcal{J} = [0, T]$, the domain $\Omega \subset \mathbb{R}^3$ in the form of a block and the diffusion tensor field $D: \Omega \rightarrow \mathbb{R}^{3 \times 3}$ representing the input data. For each $x \in \Omega$, $D(x)$ is symmetric and positive definite. The problem for the Allen-Cahn diffusion PDE reads

$$\xi^2 \frac{\partial p}{\partial t} = \xi^2 \nabla \cdot D \nabla p + f_0(p) \quad \text{in } \mathcal{J} \times \Omega, \quad (1)$$

$$\frac{\partial p}{\partial n} \Big|_{\partial \Omega} = 0 \quad \text{on } \mathcal{J} \times \partial \Omega, \quad (2)$$

$$p|_{t=0} = I \quad \text{in } \Omega. \quad (3)$$

where

- p is the unknown function $p: \mathcal{J} \times \Omega \rightarrow \mathbb{R}$,
- f_0 represents an impulse noise initial condition
- $f_0(p) = p(1-p)(p-\frac{1}{2})$,
- ξ is related to the thickness of the diffusion interface layer.

Numerical scheme

The discretization procedure follows the idea of the method of lines. We use an explicit scheme:

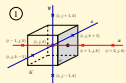
- **time discretization:** 4th order Runge-Kutta, adaptive.
- **spatial discretization:** finite volume method, several versions, higher order of accuracy preferred.
- rectangular structured grid, sliced for MPI parallelization.

Details of the high order scheme

The semidiscrete scheme of (1) assumes the form

$$\xi^2 \partial_t p = \xi^2 \sum_{k \in \mathcal{K}_k} F_{k,\sigma}(t) + f_{0,k}(t),$$

where \mathcal{K}_k is the set of all faces of the cell k . $F_{k,\sigma}(t)$ are the respective numerical fluxes at the time t which contain the difference quotients observing a symmetric 5-point stencil and approximating the derivatives $\partial_x p, \partial_y p, \partial_z p$ at the center of the face σ .



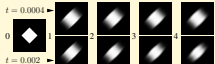
- The difference quotient approximating the derivative in the direction perpendicular to the face σ uses a nonquidistant point distribution in order to avoid redundant interpolation (Fig. 1).
- The remaining derivatives are approximated using a uniform 5-point stencil. The stencil points (depicted as green

crosses) are interpolated from the neighboring grid nodes using 1-dimensional cubic interpolation (Fig. 2).

Artificial diffusion

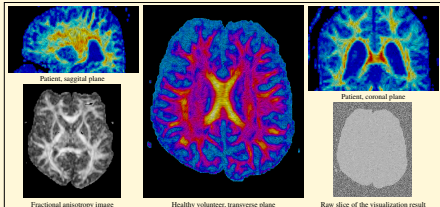
Before the high order derivative approximations were chosen, several tests were performed using simpler

numerical schemes. Below is the comparison of four schemes restricted to \mathbb{R}^2 . The white square underwent a process of anisotropic diffusion directed along the axis $y = x$.



MR-DTI visualization results

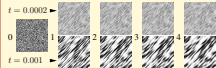
The streamlines indicate the direction of neural fiber bundles. Color mapping added with respect to fractional anisotropy [2].



Spatial discretization schemes:

1. FV, 5-point stencil (4th order), cubic interpolation.
2. FV, 5-point stencil, linear interpolation.
3. Finite volume (FV), 2nd order central difference quotients.
4. Finite difference, 1st order forward-backward quotients.

Comparison using a noisy initial condition:



Conclusion: Low order schemes suffer from artificial (isotropic) diffusion of an unacceptable strength.

Convergence results

Experimental order of convergence (EOC) has been measured for the high order FV scheme with cubic interpolation, considering the solution with mesh size 1.276E-03 precise.

Mesh size	L_2 error	L_2 EOC	L_∞ error	L_∞ EOC
1.020E-02	1.806E-04	—	3.798E-03	—
5.102E-03	6.137E-05	1.557E+00	1.188E-03	1.675E+00
2.551E-03	1.828E-05	1.747E+00	5.256E-04	1.177E+00

Conclusions and prospects

A parallel algorithm for solving the problem for the Allen-Cahn equation in the context of MR-DTI has been developed. The proposed FV scheme exhibits satisfactory rates of artificial diffusion and its convergence has been proved experimentally.

Currently, we focus on the theoretical proof of convergence. Preliminary error estimates have been obtained for the isotropic case ($D = I$).

References

- [1] R. Eymard, T. Gallouët, R. Herbin, Finite Volume Methods, Handbook of Numerical Analysis, 9 G. Ciarlet, J.-L. Lions eds, 715-1022 Elsevier, 2000.
- [2] D. Tuchenschläger, R. Deriche, Tensor Field Visualization with FEM's and Application to DTI-based Fiber Visualization, INRIA Sophia-Antipolis, Orleans Lab, France, 2005.

MEGIDDO: Numerical Issues Behind the MR-DTI Visualization Algorithm

Pavel Strachota

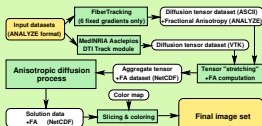
Mathematical Modelling Group, Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague



Introduction

For the purpose of MR-DTI (*Magnetic Resonance Diffusion Tensor Imaging*) data visualization in human brain tractography, we have developed a numerical algorithm based on a mathematical model of texture diffusion. Accompanied by data preprocessing and postprocessing procedures, this algorithm forms the cornerstone of the **MEGIDDO** (*Medical Employment of Generating Images by Degenerate Diffusion Operator*) software tool. To enter clinical use, the model parameters and numerical instruments need to be chosen properly so that the procedure gives satisfactory results with limited computational resources and time available.

Outline of the MEGIDDO visualization workflow



- preprocessing:** *diffusion tensor field* retrieval from **MedNIRIA**, diffusion ellipsoid stretching (elongating its semi-major axis)
 - A symmetric, positive definite tensor $D \in \mathbb{R}^{3 \times 3}$ defines an ellipsoid by $\sqrt{D^{-1}} \cdot x = 1$.
- anisotropic diffusion** of a noisy 3D texture controlled by the tensor field \Rightarrow the emerging streamlines resemble the paths of neural fiber tracts
- postprocessing:** slicing, colorization by fractional anisotropy (\Rightarrow by the density of neural fibers)

Anisotropic diffusion by the Allen-Cahn equation

$$\frac{\partial p}{\partial t} = \xi \sigma \nabla \cdot D \nabla p + \frac{1}{\xi} f_0(p) \quad \text{in } [0, T] \times \Omega,$$

$$\left. \frac{\partial p}{\partial n} \right|_{\partial \Omega} = 0 \quad \text{on } [0, T] \times \partial \Omega,$$

$$p|_{t=0} = l \quad \text{in } \Omega.$$

- $\Omega \dots$ the examined region of interest (ROI)
- $T \dots$ dimensionless final time
- $p = p(t, \mathbf{x}) \dots$ unknown function of picture intensity
- $l \dots$ noisy initial condition
- $D \dots$ "stretched" input diffusion tensor field
- $f_0(p) = p(1-p)(p-\frac{1}{2})$
- $\xi \dots$ contrast adjustment parameter
- $\sigma \dots$ input data scaling factor

Numerical solution by the method of lines

- spatial discretization: **finite volume (FV)** method, structured mesh
- temporal discretization: **Runge-Kutta-Merson** method, 4th order, adaptive time stepping
- semidiscrete scheme for the numerical solution p^h :

$$\xi \frac{d}{dt} p^h(t) = \xi \sum_{K \in \mathcal{T}} F_{K,\sigma}(t) + \frac{1}{\xi} f_0(K(t))$$

- $K \dots$ control volume
- $\sigma \dots$ edge
- $\mathcal{E}_K \dots$ edges of K
- $F_{K,\sigma} \dots$ **numerical flux** across σ

Flux approximation schemes in the FV method

- 2nd order central difference approximation with linear interpolation of the missing points in the difference stencil
- 4th order *multipoint flux approximation* (MPFA) central diff. + linear interpolation
- 4th order MPFA central diff. + cubic interpolation
- classical forward-backward 1st order finite difference (FD) scheme for comparison

Visual assessment of undesired isotropic numerical diffusion

- deteriorates image quality by blurring \Rightarrow needs to be suppressed
- Comparison of crops of one sample DTI visualization by different numerical schemes:



FD FV 2nd order MPFA cubic 4th order

- FD scheme** - asymmetric w.r.t. the (major) direction of diffusion prescribed by $D(\mathbf{x})$: strong num. diff. along ∇ , but not along $\nabla \cdot$
- FV with **2nd order** flux approx. - strong isotropic diffusion everywhere
- FV with **MPFA** - much less num. diff. \Rightarrow good results, **sharper** streamlines visualization

Quantitative assessment of numerical diffusion by total variation

$$TV(p^h) = \sum_{K \in \mathcal{T}} |\nabla_{\sigma} p_K^h| m(K)$$

- TV proportional to the number of edges in the image p^h and its contrast
- generally: **higher TV** \Rightarrow **more (and sharper) edges** in the image

Phantom diffusion tensor fields for TV-based scheme testing

- artificially created uniform tensor field $D(\varphi, \theta)$ with eigenvalues 100, 1, 1 and the principal eigenvector

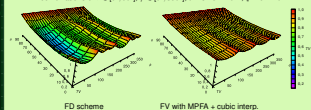
$$\mathbf{v}_1(\varphi, \theta) = (\cos \varphi \cos \theta, \sin \varphi \cos \theta, \sin \theta)$$

- measurement of TV depending on the diffusion direction given by φ, θ

TV-based scheme comparison results

- identical computations with different num. schemes on $\Omega = (0, 1)^3$, comparing TV normalized to the interval $[0, 1]$

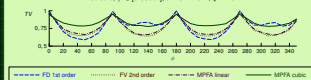
All combinations of $\theta \in [0^\circ, 90^\circ]$, $\varphi \in [0^\circ, 350^\circ]$; $t = 3.75 \times 10^{-4}$, $\xi = 5 \times 10^{-3}$.



FD scheme

FV with MPFA + cubic interp.

θ fixed to 0, $\varphi \in [0^\circ, 350^\circ]$; $t = 5 \times 10^{-4}$, $\xi = 10^{-2}$.



Demonstration: visualization settings for production runs

Parameters of a sample high resolution visualization of the whole brain:

- FV scheme with MPFA and cubic interpolation
- diffusion ellipsoid stretching factor: 50
- size of the ROI in model scale: $\Omega = [0, 0.58] \times [0, 0.695] \times [0, 0.34375]$
- numerical grid dimensions $668 \times 800 \times 396$ nodes
- $\xi = 6 \times 10^{-3}$, $\sigma = 5 \times 10^{-4}$, $T = 2 \times 10^{-4}$
- MPI parallel computation requirements: 22 min on 24 cores of AMD Opteron 2427, 23 GB RAM, efficiency > 90%

References

P. Strachota: Implementation of the MR Tractography Visualization Kit Based on the Anisotropic Allen-Cahn Equation. *Kybernetika*, 45 (2009), pp. 657-669.

Obsah

- 1 Úvod
- 2 Univerzální formáty pro ukládání dokumentů
- 3 Typografický systém L^AT_EX
- 4 Nástroje pro tvorbu schémat a obrázků
- 5 Tvorba prezentací v L^AT_EXu
- 6 Tvorba posterů v L^AT_EXu
- 7 Vědecká vizualizace**

Nástroje pro vědeckou vizualizaci

- gnuplot
- ParaView
- VisIt
- MATLAB

Odkazy 1/2



[http:](http://andreas.welcomes-you.com/projects/a0poster/)

[//andreas.welcomes-you.com/projects/a0poster/](http://andreas.welcomes-you.com/projects/a0poster/)



<http://asymptote.sourceforge.net/>



<http://www.cstug.cz/>



<http://www.djvu.org/>



<http://www.inkscape.org/>



<http://www.latex-project.org/>



<http://www.lyx.org/>

Odkazy 2/2



<http://latex-beamer.sourceforge.net/>



[http://tams-www.informatik.uni-hamburg.de/
applets/jfig/](http://tams-www.informatik.uni-hamburg.de/applets/jfig/)



<http://www.wikipedia.org>



<http://www.xfig.org/>