Zpravodaj Československého sdružení uživatelů TeXu

Zdeněk Wagner

Fraktální obrazce v PostScriptu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 13 (2003), No. 1, 45–53

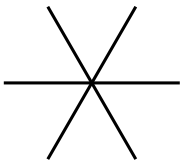
Persistent URL: http://dml.cz/dmlcz/149918

	systém	TEX	formát	editor	prezerač	prog. grafika	ostat. grafika	graf. výst.
55	linux SunOS IRIX Win98	TeXlive	pdfcsplain csplain	Vim	xpdf gv acroread	METAPOST	GIMP	PS PDF

Fraktální obrazce v PostScriptu

Zdeněk Wagner

Fraktály jsou velmi důležité objekty, které nacházejí využití v mnoha vědních oborech. Mají praktický význam i v oblastech, kde bychom to možná ani netušili. Uplatní se při studiu členitosti mořského pobřeží, při popisu povrchu katalyzátorů i při zkoumání podmínek proudění tekutin v potrubí. Je zřejmé, že rozsah jejich použití je značný, a pochopitelně se podrobnému popisu zde věnovat nebudeme. Pro nás budou fraktály jen zajímavým geometrickým útvarem, který lze graficky využít. Předvedeme si, jak lze jednoduché fraktály generovat přímo v PostScriptu a jak byl vytvořen obrázek na obálce tohoto čísla Zpravodaje.



Velmi zjednodušeně lze říci, že fraktál je struktura, která sama sebe pravidelně opakuje ve zmenšené velikosti. Zvětšíme-li tedy libovolnou část fraktálu, najdeme v ní stejný geometrický motiv. Programově lze tedy takový objekt generovat pomocí rekurzivních funkcí. Podle definice se základní fraktální struktura musí opakovat donekonečna. Jenže na vykreslení by počítač musel disponovat nekonečnou velikostí paměti a vykreslování by trvalo nekonečně dlouho. To však není praktické a matematicky dokonalé fraktály nemusí vždy vypadat hezky. Proto ukončujeme vykreslování v určité, poměrně malé hloubce vnoření. Jako první příklad si uvedeme sněhovou vločku, jejíž základní motiv obsahuje šest paprsků vycházejících ze společného středu.

Než se pustíme do vysvětlování, musíme trošku odbočit. Článek obsahuje několik fraktálních obrázků, které sdílejí společná makra. Navíc k ladění a drobným úpravám maker dochází i při psaní článku. Proto si uložíme všechna makra do souboru `fraktaly.ps` a na začátku LATEXového souboru použijeme příkaz

`\special{header=fraktaly.ps}`. Na základě tohoto povelu DVIPS vloží makra do PostScriptového dokumentu a fraktální obrázky je mohou využívat. Struktura souboru maker je následující:

```
%!PS
20 dict /Fraktaly exch def
Fraktaly begin
definice maker
end
```

Vidíte, že jsme si založili vlastní slovník a definice budeme vkládat do něj.

Nyní se můžeme vrátit ke sněhové vločce. Jak jsme již uvedli, budeme na základní motiv pohlížet jako na šest paprsků vycházejících ze společného středu. Paprsek tedy vytvoříme jako úsečku jednotkové délky, na níž budeme aplikovat transformace: posun, zvětšení a otočení. Vykreslíme jej tímto makrem:

```
% <w> <rot> <x> <y> beam
/beam {
  .2 4 index sqrt div setlinewidth
  gsave
    translate rotate dup scale
    newpath 0 0 moveto 1 0 lineto stroke
  grestore
} bind def
```

Parametry makra jsou postupně šířka, definující zvětšení, úhel otočení a souřadnice středu. Všimněte si, že tloušťku čáry neměníme ve stejném poměru.

Paprsek nyní využijeme jako základní kámen k vytvoření hvězdy. Sestavíme z něj rekurzivní makro:

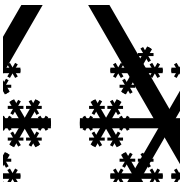
```
% <depth> <w> <x> <y> star
/star {
  0 60 333 {
    4 index 0 gt {
      4 index 1 sub        % depth w x y r depth'
      4 index .3 mul       % depth w x y r depth' w'
      5 index .7 mul dup   % depth w x y r depth' w' 2w' 2w'
      4 index cos mul      % depth w x y r depth' w' 2w' dx
      6 index add exch     % depth w x y r depth' w' x' 2w'
      4 index sin mul      % depth w x y r depth' w' x' dy
      5 index add          % depth w x y r depth' w' x' y'
      star
    } if
    3 index exch  % w x y w r
```

```
    3 index 3 index beam
  } for
  4 {pop} repeat
} bind def
```

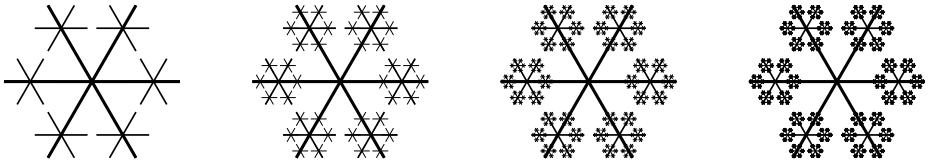
Parametry tohoto makra jsou vyžadovaná hloubka rekurze, šířka (vlastně délka paprsku) a souřadnice středu. V makru snadno najdeme cyklus s krokem 60°. Neděste se horní hranicí 333. Sice při celočíselné aritmetice k zaokrouhlovacím chybám nedochází, ale používání horní meze nad skutečnou požadovanou hranicí je dobrým zvykem. Požadujeme-li nulovou hloubku rekurze, vniřní podmíněný příkaz se nevykoná. Na vrcholu zásobníku máme tedy úhel, který tam vložil operátor `for`. Abychom připravili parametry pro makro `beam`, musíme ze zásobníku zkopírovat šířku a prohodit ji s úhlem. Dále musíme zkopírovat souřadnice středu. Makro `beam` tyto hodnoty ze zásobníku odstraní, takže při dalším průchodu cyklem je zásobník ve stejném stavu. Po skončení cyklu odstraníme parametry operátorem `pop`. Takto jsme vytvořili již dříve uvedený základní motiv.

Při nenulové hloubce rekurze se dostane ke slovu podmíněný příkaz. Na každý paprsek do vzdálenosti 70 % délky nakeslíme další hvězdu zmenšenou na 30 % původní velikosti. Nejprve zkopírujeme hloubku rekurze a zmenšíme ji o jednotku. Dále vypočteme novou šířku a nové souřadnice středu. Tyto hodnoty pak předáme makru `star`. Výsledek pro zvyšující se hloubku rekurze vidíte na obrázku 1. Obrázky byly získány jednoduchým způsobem. Soubor maker nám při zpracování LaTeXového dokumentu vložil DVIPS, takže při hloubce 4 stačí psát:

```
%!PS
%%Pages: 1
%%BoundingBox: 0 0 66 66
%%Page: 1 1
Fraktaly begin
4 33 33 33 star
end
showpage
%%EOF
```

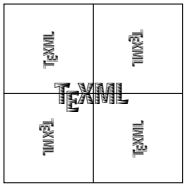


Podívejme se na jedenáctkrát zvětšený výsek spodní části paprsku směřujícího doprava. Získali jsme jej tak, že jsme v příkazu `\includegraphics` nastavili nepovinné parametry na hodnoty `[clip,viewport=54 25 60 31,width=66bp]`. Ač se při použití hloubky rekurze 4 z obrázku 1 zdá, že se malé vločky slijí do jednolité černé plochy, při zvětšení je zřejmé, že tomu tak ve skutečnosti není. Stejný motiv se opakuje donekonečna. I kdybychom použili nekonečnou hloubku rekurze, abychom získali skutečný fraktál, zůstaly by všechny



Obrázek 1: Sněhové vločky s hloubkou rekurze 1–4

vločky „děravé“ a stále by opakovaly svoji strukturu. Tak je předvedena základní vlastnost fraktálů v praxi.



Po této rozcvičce se můžeme pustit do programování textového fraktálu použitého na obálce. Základní strukturou je nápis TEXML. Byl vybrán proto, že hlavní náplní tohoto čísla jsou články o sazbě XML souborů v TEXu. Vnitřní písmeno X bylo vynecháno úmyslně. Text je sázen písmem LexonGothic Xylo ze Střešovické písmolijny. Na obrázku je vidět příklad s hloubkou rekurze 1. Při nulové hloubce bychom ve středu čtverce viděli jen základní text. Každý další rekurzivní krok rozdělí čtverec na čtyři menší a do jejich středu vloží stejný text otočený o 90° ve směru nebo proti směru hodinových ručiček.

Makra jsou v tomto případě složitější. Nejprve nadefinujeme makro pro výběr písma, kde jako jediný parametr zadáme požadovanou velikost:

```
% <size> LX
/LX {/LexonXylo exch selectfont} bind def
```

Samostatně by však takové makro nefungovalo, protože tento font není v PostScriptovém RIPu přítomen. Fontový soubor jsme tedy programem T1ASCII převedli do formátu PFA. Přidáme jej k dokumentu podobně jako fraktální makra, tedy příkazem `\special{header=lexonxyl.pfa}`.

V makru se využívá operátor `selectfont` z PostScriptu Level 2. Máme-li pouze zařízení Level 1, musíme před definici makra `LX` přidat:

```
% Emulation of selectfont by Level 1 operators
/selectfont {exch findfont exch scalefont setfont} bind def
```

Při sazbě textu TEXML budeme vodorovně i svisle posunovat písmeno E tak, jak je to v logu TEX zvykem. Hodnota posunu byla určena empiricky tak, aby to vypadalo hezky. Zde je makro, které jako jediný parametr přijímá velikost písma. Předpokládá se, že aktuální bod sazby již byl nastaven.

```
% <size> texml
/texml {
```

```
  dup LX                    % size
  (T) show                  % size
  neg dup 25 div exch       % -size/25 -size
  dup 4 div dup neg exch    % -size/25 -size size/4 -size/4
  3 -1 roll 10 div exch rmoveto
  (E) show rmoveto (XML) show
} bind def
```

Při dalším zpracování budeme potřebovat šířku a výšku textu, abychom jej mohli umístit na střed. Změříme tedy ohraničovací rámeček textu TEXML. Započítáme pouze korekci na horizontální posuny:

```
% <size> sz <width> <height>
/sz {
  dup LX % size
  gsave newpath 0 0 moveto
  (TEXML) true charpath flattenpath pathbbox % size llx lly urx ury
  3 1 roll sub                               % size llx urx height
  exch 3 1 roll sub                          % size height width
  2 index 25 div sub 3 1 roll 10 div sub grestore
} bind def
```

Nyní můžeme text vysadit na požadované místo. Zadáme úhel otočení, souřadnice středu čtverce a velikost písma. Všimněte si, že svislá velikost nebyla předchozím výrazem určena správně, přestože byl použit operátor `flattenpath`. Proto je do následujícího makra vložena empirická korekce. Text otáčíme vždy o násobek 90°, proto místo skutečného úhlu používáme jen malá celá čísla.

```
% <rot> <x> <y> <size> xput
/xput {
  dup sz                            % rot x y sz width height
  2.5 div neg exch 2 div neg exch   % rot x y sz -h/2.5 -w/2
  6 3 roll                          % sz -w/2 -h/2 rot x y
  gsave translate 90 mul rotate moveto texml grestore
} bind def
```

Nyní můžeme všechny stavební kameny složit do výsledného makra. Je opět rekurzivní, proto si jej nejprve ukážeme a dodatečně vysvětlíme.

```
% <w> <r> <x> <y> <sz> <depth> rxput
% w means half width of the square, x and y is the square's centre
/rxput {
  dup 0 gt {
    5 index 2 div               % w r x y sz depth w/2
    5 index 1 exch sub          % w r x y sz depth w/2 r'
    5 index 2 index sub         % w r x y sz depth w/2 r' x'
```

```
    5 index 3 index add          % w r x y sz depth w/2 r' x' y'
    5 index 2 div 5 index 1 sub  % w r x y sz depth w/2 r' x' y' sz' depth'
    rxput
    5 index 2 div                % w r x y sz depth w/2
    5 index -1 exch sub          % w r x y sz depth w/2 r'
    5 index 2 index add          % w r x y sz depth w/2 r' x'
    5 index 3 index add          % w r x y sz depth w/2 r' x' y'
    5 index 2 div 5 index 1 sub  % w r x y sz depth w/2 r' x' y' sz' depth'
    rxput
    5 index 2 div                % w r x y sz depth w/2
    5 index -1 exch sub          % w r x y sz depth w/2 r'
    5 index 2 index sub          % w r x y sz depth w/2 r' x'
    5 index 3 index sub          % w r x y sz depth w/2 r' x' y'
    5 index 2 div 5 index 1 sub  % w r x y sz depth w/2 r' x' y' sz' depth'
    rxput
    5 index 2 div                % w r x y sz depth w/2
    5 index 1 exch sub           % w r x y sz depth w/2 r'
    5 index 2 index add          % w r x y sz depth w/2 r' x'
    5 index 3 index sub          % w r x y sz depth w/2 r' x' y'
    5 index 2 div 5 index 1 sub  % w r x y sz depth w/2 r' x' y' sz' depth'
    rxput
  } if
  pop xput pop
} bind def
```

Při nulové hloubce rekurze se podmíněný příkaz neprovede. Odstraníme-li nyní hloubku rekurze, máme na vrcholu zásobníku přesně ty parametry, které vyžaduje makro `xput`. Po jeho provedení zbude v zásobníku nepoužitá poloviční šířka čtverce, kterou musíme odstranit. Výsledkem je tedy nápis TEXML uprostřed základního čtverce. Rekurzivní volání se provádí uvnitř podmíněného příkazu. Nejprve zmenšíme šířku čtverce na polovinu. Aby správně fungovala rotace textu, musíme od požadovaného úhlu odečíst rotaci čtverce, z něhož rekurzi voláme. Pak vypočteme souřadnice středu nového čtverce, zadáme poloviční velikost písma a nakonec vložíme hloubku rekurze zmenšenou o jednotku. Postup opakujeme pro všechny menší čtverce.

V příkladech si chceme zobrazit rámeček a v ukázce základního motivu i rozdělení na menší čtverce. Dosáhneme toho jednoduchými makry:

```
% <w> <r> <x> <y> <sz> <depth> frxput
% framed rxput
/frxput {
  gsave
    3 index 3 index translate
    5 index dup scale
    5 index 4000 div setlinewidth 0 setgray
    newpath
    -1 -1 moveto -1 1 lineto 1 1 lineto 1 -1
```

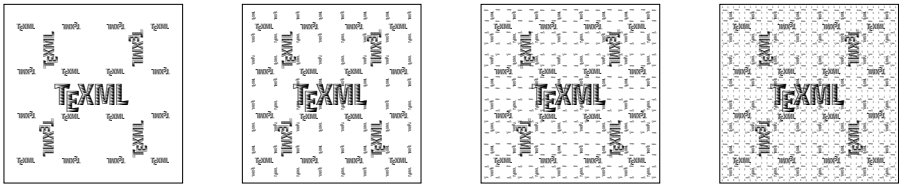
```
    lineto closepath stroke
  grestore
  rxput
} bind def

% <w> <r> <x> <y> <sz> <depth> Frxput
% framed rxput with middle cross
/Frxput {
  gsave
    3 index 3 index translate
    5 index dup scale
    5 index 6000 div setlinewidth 0 setgray
    newpath
    0 -1 moveto 0 1 lineto 1 0 moveto -1 0 lineto
    closepath stroke
  grestore
  frxput
} bind def
```

Nyní již můžeme vykreslit fraktály. Opět jsou vlastní soubory jednoduché, neboť jen otevřeme odpovídající slovník a zavoláme příslušné makro. Základní motiv byl vytvořen makry:

```
%!PS
%%Pages: 1
%%BoundingBox: 0 0 77 77
%%Page: 1 1
Fraktaly begin
33.5 0 33.5 33.5 11 1 Frxput
end
showpage
%%EOF
```

Obrázky s vyšší hloubkou rekurze se liší jen tím, že netiskneme vnitřní dělení čtverce:

```
%!PS
%%Pages: 1
%%BoundingBox: 0 0 77 77
%%Page: 1 1
Fraktaly begin
33.5 0 33.5 33.5 11 5 frxput
end
```



Obrázek 2: Fraktální text s hloubkou rekurze 2–5

```
showpage
%%EOF
```

Hotové fraktály s hloubkou rekurze 2–5 jsou na obrázku 2. Vidíme, že motiv je čtvercový, ale prostor na obálce, vyhrazený pro obrázek, je obdélníkový. Proto jsme pod větší fraktální motiv přidali dva poloviční s hloubkou rekurze o jednotku nižší a otočený nohama vzhůru. Soubor pro použití na obálce pak vypadá takto:

```
%!PS
%%Pages: 1
%%BoundingBox: 0 0 168 252
%%Page: 1 1
Fraktaly begin
84 0 84 168 24 6 rxput
42 2 42  42 12 5 rxput
42 2 126 42 12 5 rxput
end
showpage
%%EOF
```

Na závěr jsme ještě přidali vertikální linky, aby prostor byl využit vizuálně symetricky. Obálka je tedy vytvořena makry (viz styl pro Zpravodaj na stránce `http://bulletin.cstug.cz`):

```
\def\xkr{\kern2.5dd}
\def\xvr{\vrule height 252bp depth 0dd width .3dd}
\def\texml{\includegraphics{texml.eps}}
\Obalka[\Centerbox{\xvr\xkr\texml\xkr\xvr}]
```

Nakonec ještě poznamenáme, že text souborů byl do tohoto článku vkládán příkazem `\verbatiminput`. Tím je zajištěno, že ukázky obsahují přesně stejný kód, který byl při kreslení obrázků použit.

Přestože kód popisovaných maker není složitý, opisování jistě není zábavné. Proto je kód těchto maker zveřejněn na WWW stránkách Zpravodaje, konkrétně na `http://bulletin.cstug.cz/bul20031.shtml` (ovšem bez komerčního střešovického fontu).

Summary: Fractal Images in PostScript

The picture used on the cover of this issue is an example of a fractal image. The article describes the PostScript macro by means of which the picture was created.

Zdeněk Wagner
wagner@cesnet.cz