# plotterart

## Modul maker pro LaTeXovou sazbu estetický produktivní grafiky

Vojtěch Žáček

11. listopadu 2005

**Abstrakt**

Tento dokument popisuje implementaci makra plotterart.

Program od verze 2.0 používá pro vykreslování a manipulaci s grafickými prvky makro mfpic.

Program je součástí bakalářské práce: obor aplikovaná informatika, Fakulta Informatiky, Masarikova Universita.

## 1 plotterart.tex

`\plotterart` Příkaz zahájí plotterart grafiku. První parametr je název grafiky, druhý parametr určuje základní šířku grafiky v bodech. Třetím parametrem je druh použitého pera a čtvrtý parametr je základním rozměrem pera (viz. `\plainipen`).

```
⟨*tex⟩
\def\plotterart #1 #2 #3 #4 {
  \message{*** BEGIN: plotterart #1 #2 #3 #4}
  \opengraphsfile{#1}

  \mfpic[1]{0}{#2}{0}{#2}
  \mfsrc{
    pla_name := #1;
    pla_width := #2;
    pla_pen := #3;
    pla_fat := #4;
  }
  \plainifunc
  \plainiglobal
  \plainipoints
  \plainitrans
  \plainipen
  \plainisym
  \plaperspective
}
```

`\endplotterart` Makro ukončuje definici plotterart grafiky. V místě použití tohoto makra je grafika vykreslena.

```
\def\endplotterart {
```

```
  \message{*** END: endplotterart}
  \endmfpic
  \closegraphsfile
}
```

\fan

Použití: `\fan X10 Y10 X20 Y20 X30 Y30 N`
Makro vykresluje soustavu úseček. Parametry makra definují tři body: $A = (X_{10}, Y_{10}), B = (X_{20}, Y_{20}), C = (X_{30}, Y_{30})$. Poslední parametr určuje počet vykreslených úseček. Makro vykreslí usečky $A$–$D_i$. První bod úsečky je vždy zadaný bod $A$, druhý dod je prvkem množiny $\{B, .., D_{1/i}, .., C\}$, která vznikne rovnoměrným dělením úsečky $B$–$C$. Index $i$ nabývá celých hodnot z intervalu $< 0, N >$.

```
\def\fan #1 #2 #3 #4 #5 #6 #7 {
  \mfsrc{
    x10 := plagetpointx(#1);
    y10 := plagetpointy(#2);
    x20 := plagetpointx(#3);
    y20 := plagetpointy(#4);
    x30 := plagetpointx(#5);
    y30 := plagetpointy(#6);
    for i = 0 step (1 / #7) until 1:
      path pxx;
      pxx =
        z10
        --
        i[z20, z30];
      plotterartdrawall(pxx);
    endfor;
  }
  \hvloc;
}
```

\join

Použití: `\join X11 Y11 X12 Y12 X21 Y21 X22 Y22 N`
Makro vykresluje soustavu úseček. Parametry makra definují čtyři body: $A = (X_{11}, Y_{11}), B = (X_{12}, Y_{12}), C = (X_{21}, Y_{21}), D = (X_{22}, Y_{22})$. Poslední parametr určuje *krok*. Makro vykreslí usečky $E_i - F_i$. První bod leží na spojnici bodů $A$ a $B$ s dělením $N$. Druhý bod leží na spojnici bodů $C$ a $D$ s dělením $N$. (viz. `\fan`)

```
\def\join #1 #2 #3 #4 #5 #6 #7 #8 #9 {
  \mfsrc{
    x11 := plagetpointx(#1);
    y11 := plagetpointy(#2);
    x12 := plagetpointx(#3);
    y12 := plagetpointy(#4);
    x21 := plagetpointx(#5);
    y21 := plagetpointy(#6);
    x22 := plagetpointx(#7);
    y22 := plagetpointy(#8);
    for i = 0 step (1 / #9) until 1:
      path pxx;
      pxx =
        i[(x11, y11), (x12, y12)]
        --
```

```
        i[(x21, y21), (x22, y22)];
      plotterartdrawall(pxx);
    endfor;
  }
  \hvloc;
}
```

\bfan Použití: `\bfan X10 Y10 X20 Y20 X30 Y30 N`

`\bfan` je modifikací makra `\fan`. Nevykresluje soustavu úseček, ale Béziérových křivek.

```
\def\bfan #1 #2 #3 #4 #5 #6 #7 {
  \mfsrc{
    x10 := plagetpointx(#1);
    y10 := plagetpointy(#2);
    x20 := plagetpointx(#3);
    y20 := plagetpointy(#4);
    x30 := plagetpointx(#5);
    y30 := plagetpointy(#6);
    for c=0 step (1 / #7) until 1:
      path pxx;
      pxx =
        z10{curl pla_curl}
        ..(pla_xbezier, pla_ybezier)..
        c[z20, z30]{curl pla_curl};
      plotterartdrawall(pxx);
    endfor;
  }
  \hvloc;
}
```

\bjoin Použití: `\bjoin X11 Y11 X12 Y12 X21 Y21 X22 Y22 N`

Makro je modifikací makra `\join`. Nevykresluje soustavu úseček, ale Bézierových křivek. Tvar křivky lze ovlivnit makry: `\placurl` `\plabezierpoint`.

```
\def\bjoin #1 #2 #3 #4 #5 #6 #7 #8 #9 {
  \mfsrc{
    x11 := plagetpointx(#1);
    y11 := plagetpointy(#2);
    x12 := plagetpointx(#3);
    y12 := plagetpointy(#4);
    x21 := plagetpointx(#5);
    y21 := plagetpointy(#6);
    x22 := plagetpointx(#7);
    y22 := plagetpointy(#8);
    for c = 0 step (1 / #9) until 1:
      path pxx;
      pxx =
        c[(x11, y11), (x12, y12)]{curl pla_curl}
        ..(pla_xbezier, pla_ybezier)..
        c[(x21, y21), (x22, y22)]{curl pla_curl};
      plotterartdrawall(pxx);
    endfor;
  }
  \hvloc;
}
```

\playaxis Vykreslí osu $y$.

```
\def\playaxis{
  \mfsrc{
    draw(pla_half_width, 0)--(pla_half_width, pla_width);
  }
}
```

\plaxaxis Vykreslí osu $x$.

```
\def\plaxaxis{
  \mfsrc{
    draw(0, pla_half_width)--(pla_width, pla_half_width);
  }
}
```

\plabox Vykreslí základní hranice grafiky. Čtverec o straně `pla_width`.

```
\def\plabox {
  \mfsrc{
    draw pplabox;
  }
}
```

\vsym Zapne globální symetrii ve vertikální rovině.

```
\def\vsym{
  \mfsrc{
    glob_h := false;
    glob_v := true;
  }
}
```

\hsym Zapne globální symetrii v horizontální rovině.

```
\def\hsym{
  \mfsrc{
    glob_h := true;
    glob_v := false;
  }
}
```

\nosym Vypne globální symetrii v obou rovinách.

```
\def\nosym{
  \mfsrc{
    glob_h := false;
    glob_v := false;
  }
}
```

\hvsym Zapne globální symetrii v obou rovinách.

```
\def\hvsym{
  \mfsrc{
    glob_h := true;
    glob_v := true;
  }
}
```

`\vloc` Zapne lokální symetrii ve vertikální rovině.

```
\def\vloc{
  \mfsrc{
    loc_h := false;
    loc_v := true;
  }
}
```

`\hloc` Zapne lokální symetrii v horizontální rovině.

```
\def\hloc{
  \mfsrc{
    loc_h := true;
    loc_v := false;
  }
}
```

`\noloc` Vypne lokální symetrii v obou rovinách.

```
\def\noloc{
  \mfsrc{
    loc_h := false;
    loc_v := false;
  }
}
```

`\hvloc` Zapne lokální symetrii v obou rovinách.

```
\def\hvloc{
  \mfsrc{
    loc_h := true;
    loc_v := true;
  }
}
```

`\platransgroup` Nastaví transformační skupinu. 0 - defaultní transformační skupina (zrcadlení kolem os $X$ a $Y$) 1 - otáčení kolem středového bodu $z_{99}$

```
\def\platransgroup #1 {
  \mfsrc{
    tg := #1 * 4;
  }
}
```

`\placurl` Zakřivení Bézierovy křivky. zakřivení:
$>1$ vetší, $=1$ standardní, $<1$ menší.

```
\def\placurl #1 {
  \mfsrc{
    pla_curl := #1;
  }
}
```

`\plarotated` Nastavení rotace všech následujících prvků o úhel $\delta$. Defaultní hodnota je 0.

```
\def\plarotated #1 {
  \mfsrc{
    delta := #1;
  }
}
```

\plascaled Zvětšení v obou osách o hodnotu Ξ. Defaultní hodnota je 1. Při hodnotě $> 1$ dochazí ke zvětšení zvětšení, a při hodnotě $< 1$ ke zmenšení.

```
\def\plascaled #1 {
  \mfsrc{
    chi := #1;
    if pla_perspective:
      pla_fat := pla_fat * chi;
      show pla_fat;
    else:%
    fi;
  }
  \plainipen
}
```

\plabezierpoint Řídící bod Bézierovi křivky $z_{42}$. Defaultně je nastaven na výchozí hodnotu středového bodu $z_{99}$.

```
\def\plabezierpoint #1 #2 {
  \mfsrc{
    pla_xbezier := #1 * pla_width;
    pla_ybezier := #2 * pla_width;
  }
}
```

\plafat Nastavení tloušťky pera.

```
\def\plafat #1 {
  \mfsrc{
    pla_fat := #1 * chi;
  }
  \plainipen
}
```

\plarand Nastavi nahodne promenne TODO … dopsat dokumentaci `\plarand`

```
\def\plarand #1 #2 #3 #4 #5 #6 #7 {
  \mfsrc{
    rand_x := #1;
    rand_y := #2;
    rand_real_x := #3;
    rand_real_y := #4;
    rand_delta := #5;
    rand_fat := #6;
    rand_chi := #7;
  }
}
```

\plaperspective Zapne perspectivu. Zvětšení (`\plascale`) ovlivní i tloušku pera.

```
\def\plaperspective{
  \mfsrc{
    pla_perspective := true;
  }
}
```

\planoperspective Vypne perspektivu.

```
\def\planoperspective{
  \mfsrc{
```

```
    pla_perspective := false;
  }
}
```

\plainiglobal Inicialize globalních proměnných.

```
\def\plainiglobal{
  \mfsrc{
    boolean symetrie;
    boolean glob_h;
    boolean glob_v;
    boolean loc_h;
    boolean loc_v;
    boolean pla_perspective;
    half = .5;
    pla_half_width = half * pla_width;
    tg = 0;
    pla_curl = 1;
    delta = 0;
    chi = 1;
    pla_xbezier = pla_width * half;
    pla_ybezier = pla_width * half;
    pla_xcenter = pla_width * half;
    pla_ycenter = pla_width * half;
    rand_x = 0;
    rand_y = 0;
    rand_real_x = 0;
    rand_real_y = 0;
    rand_delta = 0;
    rand_fat = 0;
    rand_chi = 0;
    boolean pla_cut;
    pla_cut := false;
  }
}
```

\plainipoints Inicializace základních a řídících bodů.

```
\def\plainipoints{
  \mfsrc{
    z1 = (0, 0);
    z2 = (0, pla_width);
    z3 = (pla_width, pla_width);
    z4 = (pla_width, 0);
    z5  = (pla_width * half, pla_width * half);
    z11 = (0, pla_width * half);
    z21 = (pla_width * half, pla_width);
    z31 = (pla_width, pla_width * half);
    z41 = (pla_width * half, 0);
    z99 = (pla_xcenter, pla_ycenter);
    z42 = (pla_xbezier, pla_ybezier);
    path pplabox;
    pplabox = z1--z2--z3--z4--cycle;
  }
}
```

\plainitrans Inicializace pole tranformací.

```
\def\plainitrans{
  \mfsrc{
    transform t[];
    t0 = identity;
    t1 = identity reflectedabout (z11,z21);
    t2 = identity reflectedabout (z31,z41);
    t3 = identity reflectedabout (z11,z21)
                  reflectedabout (z31,z41);
    t4 = identity;
    t5 = identity rotatedaround (z99, 90);
    t6 = identity rotatedaround (z99, 180);
    t7 = identity rotatedaround (z99, 270);
  }
}
```

\plainipen Inicializace pole per.

```
\def\plainipen{
  \mfsrc{
    pen PERA[];
    PERA1 = pensquare scaled pla_fat;
    PERA2 = makepen(fullcircle scaled pla_fat);
    PERA3 = PERA1 transformed
            identity rotated 45;
    PERA4 = PERA3 transformed
            identity rotated 10;
    if ((pla_pen > 0) and (pla_pen < 5)):
      pickup PERA[pla_pen];
    else:%
      pickup defaultpen;
    fi;
  }
}
```

\plainifunc Inicializace základních funkcí.

```
\def\plainifunc{
  \mfsrc{
    def plotterartdraw(expr pa)(expr tr)(expr out) =
      if out:
        t97 := identity rotatedaround (z99, plagetranddelta(delta));
        t98 := identity scaled (plagetrandchi(chi));
        t99 := identity shifted
         (pla_half_width - (chi * pla_half_width)
                  + uniformdeviate rand_real_x,
          pla_half_width - (chi * pla_half_width)
                  + uniformdeviate rand_real_y);
        if pla_cut:
          draw pa
            transformed tr
            transformed t97
            transformed t98
            transformed t99
            cutbefore pplabox cutafter pplabox;
        else:%
          draw pa
```

```
            transformed tr
            transformed t97
            transformed t98
            transformed t99;
        fi;
      else:%
      fi;
    enddef;
  }
  \mfsrc{
    def plotterartdrawall(expr pxxall) =
      plotterartdraw(pxxall, t[tg+0], true);
      plotterartdraw(pxxall, t[tg+1], (glob_h and loc_h));
      plotterartdraw(pxxall, t[tg+2], (glob_v and loc_v));
      plotterartdraw(pxxall, t[tg+3], (glob_h and loc_h
                                       and glob_v and loc_v));
    enddef;
  }
  \mfsrc{
    vardef plagetpointx(expr val)
      = ((val + (uniformdeviate rand_x)) * pla_width)
    enddef;
    vardef plagetpointy(expr val)
      = ((val + (uniformdeviate rand_y)) * pla_width)
    enddef;
    vardef plagetranddelta(expr val)
      = (val + (uniformdeviate rand_delta))
    enddef;
    vardef plagetrandchi(expr val)
      = (val + (uniformdeviate rand_chi))
    enddef;
  }
}
```

\placuttrue

```
\def\placuttrue{
  \mfsrc{
    pla_cut := true;
  }
}
```

\placutfalse

```
\def\placutfalse{
  \mfsrc{
    pla_cut := false;
  }
}
```

\plainitrans Inicializace symetrie. Globalni a lokalni symetrie jsou zapnuty.

```
\def\placlip{
  \mfsrc{
    clip currentpicture to pplabox;
  }
}
```

`\plainitrans` Inicializace symetrie. Globalni a lokalni symetrie jsou zapnuty.

```
\def\plainisym{
  \hvsym
  \hvloc
}
⟨/tex⟩
```

# 2 plotterart.sty

```
⟨*sty⟩
\input plotterart.tex\relax
⟨/sty⟩
```