

POSTSCRIPT

LANGUAGE

JOURNAL

VOL. 3, NO. 2

INTERNATIONAL EDITION

*Je reconnais - toi;
Cette adorable personne c'est toi
sous le grand
oeil de chapeau canotier
Voici
ta bouche
voilà la figure
ton
voici enfin
l'impar
faite image
de ton buste a
doré vu comme
à travers un nuage
ou exquis
un peu
plus bas
c'est ton
cœur
qui
bat*

NOV. DEC. 1991

THIS ISSUE:

- Cover Art: PostScript calligrammes
- Interview: Ned Bunnell of Adobe Systems
- PostScript Security: The Trojan Horse
- MAC corner: Fontographer 3.1
- Programming: 3-D perspectives
- Developer's Corner: Performance issues
- Graphics to the Editor: Contour designs

POSTSCRIPT calligrammes

By Jacques André

“What kind of calligrammes would Apollinaire design today if he had POSTSCRIPT?”

Apollinaire is a French poet who died in 1918. A major part of his oeuvre is in the form of “calligrammes”. Alas, most of them are published in his hand writing. Only few have been typeset, and the result is far from good. Here is an example of one of his original calligrammes, called ‘Le jet d’eau’, taken from ‘Collection de la Pleiade’ published by Gallimard in Paris:

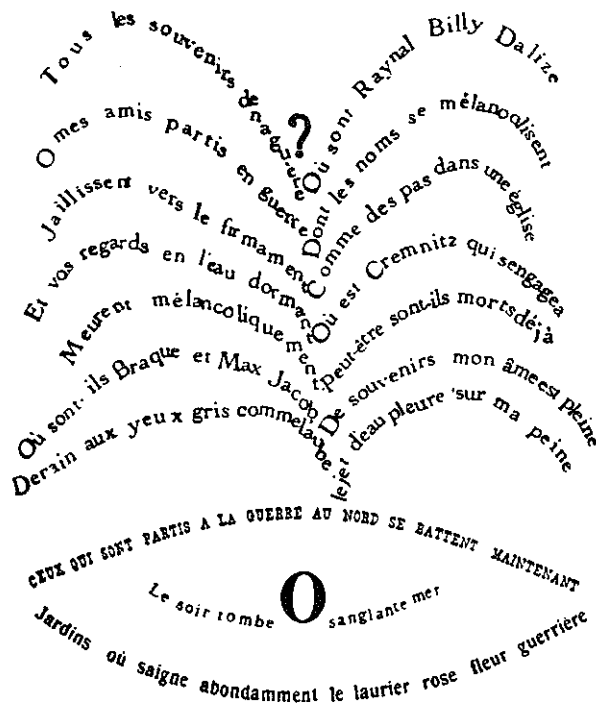


Fig. 1. Le jet d'eau - original version

The main problem with such calligrammes is that classical (photo)typesetting did not easily allow the typesetter to play with character size and orientation.

POSTSCRIPT allows character size and orientation to be modified as follows:

- a character may be displayed at any orientation and size.
- arbitrary text may be written along any path, and each character may be either orthogonal or parallel to this path.
- text can be used to fill an area of any shape.

The figure below shows a POSTSCRIPT version of ‘Le jet d’eau’:

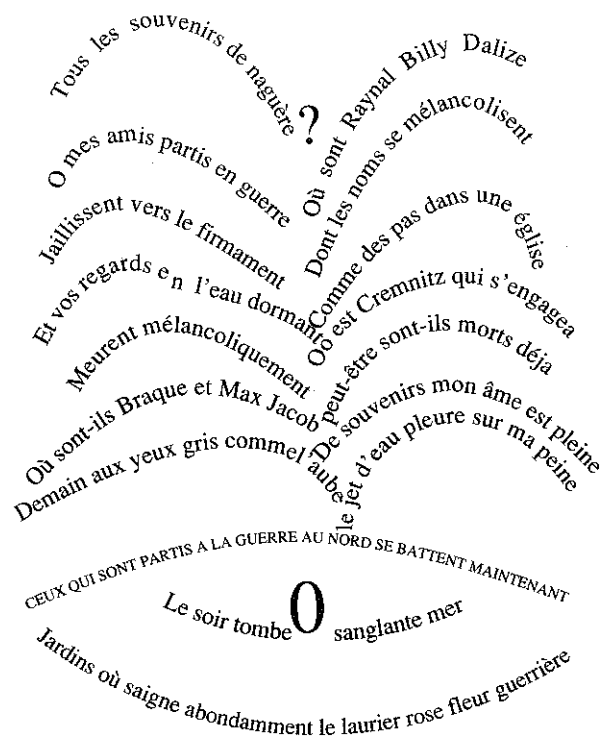


Fig. 2. Le jet d'eau – POSTSCRIPT version

The calligramme below and the one shown overleaf have been drawn using the Adobe blue book (program 11) procedure to print characters along a line. The Bézier control points have been decided "by hand". That is why these drawings are not as perfect as they could be.

Adieu mon Lou mes larmes
 tombent
 Je ne te reverrai plus
 jamais
 Entre nous deux, ma Lou, se dresse
 l'Ombre

Actually the problem is not how to be as close as possible to the manuscript: the problem is "How would Apollinaire have liked to see his calligrammes printed?". In that sense, it is useless to go further in the precision of these examples.

Obviously, a poet could not use native PostScript, but products such as Illustrator, Letrastudio etc. are, on the other hand, possible for them to use.

Does the poet of today know that such tools are available?

"Poètes, a vos plumes? Non, a vos Macs."

Jacques André is the coordinator of the DIDOT project on teaching digital type to both computer scientists and type designers.

Cover art – source listing

The graphic shown on the cover of this issue was produced using the code listed below. The 'skeleton' version of the poem printed at the end of this article was produced by setting the value of test to be true, so that rather than writing text along a path, the path is stroked.

```

%!
% --- Poeme du 9 Fevrier 1915

% --- ACCENTED FONTS ---
/dictlocal 20 dict def

/Reencode {
    dictlocal begin
    /Nouveauvect exch def

    /Nomnouvellepolice exch def

    /Nomplicatedbase exch def

    /Plicatedbasedict Nomplicatedbase findfont def

    /Nouveaudict Plicatedbasedict maxlength dict def
    
```

```

Plicatedbasedict {%forall
    exch dup /FID ne {
        dup /Encoding eq {

            exch dup length array copy Nouveaudict
            3 1 roll put

        } {
            exch Nouveaudict 3 1 roll put
        }
        ifelse
    } {
        pop pop
    } ifelse
} forall

Nouveaudict /FontName
Nomnouvellepolice put
Nouveauvect aload pop
Nouveauvect length 2 idiv {
    Nouveaudict /Encoding get 3 1 roll put
} repeat

Nomnouvellepolice Nouveaudict definefont
pop end
} def
%dictlocal

/CMPvect [
8#200 /Adieresis
8#202 /Ccedilla
8#203 /Eacute
8#205 /Odieresis
8#210 /agrave
8#211 /acircumflex
8#212 /adieresis
8#213 /aacute
8#215 /ccedilla
8#216 /eacute
8#217 /egrave
8#220 /ecircumflex
8#221 /edieresis
8#224 /icircumflex
8#225 /idieresis
8#231 /ocircumflex
8#232 /odieresis
8#235 /ugrave
8#236 /ucircumflex
8#237 /udieresis
8#245 /Acircumflex
8#346 /Ecircumflex
8#347 /Agrave
8#350 /Edieresis
8#351 /Egrave
8#353 /Icircumflex
8#354 /Idieresis
8#357 /Ocircumflex
8#363 /Ucircumflex
8#364 /Ugrave
] def

/Helvetica /F-Helvetica CMPvect Reencode
/Times-Roman /F-Times-Roman CMPvect Reencode
/Times-Italic /F-Zapf CMPvect Reencode
    
```

Cover Art: POSTSCRIPT calligrammes

```
%% pathtext procedure for writing text
%% along a path, taken from the Adobe
%% Blue book, program 11 page 171.
```

```
/pathtextdict 26 dict def
```

```
/pathtext {
  pathtextdict begin
  /offset exch def
  /str exch def
```

```
  /pathdist 0 def
```

```
  /setdist offset def
```

```
  /charcount 0 def
```

```
  gsave
```

```
  flattenpath
```

```
  {
```

```
    movetoproc
```

```
  } {
```

```
    linetoproc
```

```
  }
```

```
  {
```

```
    curvetoproc
```

```
  } {
```

```
    closepathproc
```

```
  }
```

```
  pathforall
```

```
  grestore
```

```
  newpath end
```

```
} def
```

```
  % pathtext
```

```
pathtextdict begin%%%%%%%%%
```

```
/movetoproc {
```

```
  /newy exch def
```

```
  /newx exch def
```

```
  /first newx def
```

```
  /firsty newy def
```

```
  /ovr 0 def
```

```
  newx newy transform
```

```
  /cpy exch def
```

```
  /cpx exch def
```

```
  } def
```

```
  % movetoproc
```

```
/linetoproc {
```

```
  /oldx newx def
```

```
  /oldy newy def
```

```
  /newy exch def
```

```
  /newx exch def
```

```
  /dx newx oldx sub def
```

```
  /dy newy oldy sub def
```

```
  /dist dx dup mul dy dup mul add sqrt def
```

```
dist 0 ne
```

```
{
```

```
  /dsx dx dist div ovr mul def
```

```
  /dsy dy dist div ovr mul def
```

```
  oldx dsx add oldy dsy add transform
```

```
  /cpy exch def
```

```
  /cpx exch def
```

```
  /pathdist pathdist dist add def
```

```
{
```

```
  setdist pathdist le
```

```
{
```

```
  charcount str length lt
```

```
{
```

```
  setchar
```

```
} {
```

```
  exit
```

```
} ifelse
```

```
}
```

```
  /ovr setdist pathdist sub def
```

```
  exit
```

```
}
```

```
  ifelse
```

```
  } loop
```

```
} if
```

```
} def
```

```
  % linetoproc
```

```
/curvetoproc {
```

```
  (ERREUR: curveto apres flattenpath) print
```

```
} def
```

```
/closepathproc {
```

```
  firstx firsty linetoproc firstx firsty movetoproc
```

```
} def
```

```
/setchar {
```

```
  /char str charcount 1 getinterval def
```

```
  /charcount charcount 1 add def
```

```
  /charwidth char stringwidth pop def
```

```
  gsave
```

```
  cpx cpy itransform translate
```

```
  dy dx atan rotate
```

```
  0 0 moveto char show
```

```
  currentpoint transform
```

```
  /cpy exch def
```

```
  /cpx exch def
```

```
  grestore
```

```
  /setdist setdist charwidth add def
```

```
  } def
```

```
  %setchar
```

```
end %%%%%%%%%%% procs
```

Cover Art: POSTSCRIPT calligrammes

```

/test false def
/epsilon {
  test {
    stroke pop pop
  } {
    pathtext
  } ifelse
} def

test {
  /bez {
    /y4 exch def
    /x4 exch def
    /y3 exch def
    /x3 exch def

    /y2 exch def
    /x2 exch def
    /y1 exch def
    /x1 exch def

    newpath x1 y1 moveto
    x2 y2 lineto x3 y3 lineto
    x4 y4 lineto stroke
    newpath x1 y1 moveto
    x2 y2 x3 y3 x4 y4 curveto stroke
  } def
  %bezier

  /cercle {
    /teta2 exch def
    /tetal exch def
    /r exch def
    /y exch def
    /x exch def

    newpath x y moveto
    x y r tetal teta2 arc closepath stroke
    newpath x y r tetal teta2 arc stroke
  } def

  /segment {
    /y2 exch def
    /x2 exch def
    /y1 exch def
    /x1 exch def

    newpath x1 y1 moveto x2 y2 lineto stroke
  } def
} {
  newpath x1 y1 moveto
  x2 y2 x3 y3 x4 y4 curveto
} def

/cerle {
  /teta2 exch def
  /tetal exch def
  /r exch def
  /y exch def
  /x exch def

  newpath x y r tetal teta2 arc
} def

/segment {
  /y2 exch def
  /x2 exch def
  /y1 exch def
  /x1 exch def

  newpath x1 y1 moveto x2 y2 lineto
} def
} ifelse

50 50 translate
/F-Zapf findfont 12 scalefont setfont
gsave 45 185 translate
0 0 moveto 25 rotate (R) show
grestore

gsave 25 125 translate
1 3 scale
0 0 moveto 40 rotate (C) show
grestore

/F-Zapf findfont 14 scalefont setfont
50 210 65 240 90 252 120 235 bez
(econnais - toi) 0 ecrire

/F-Zapf findfont 14 scalefont setfont
30 150 50 190 120 240 190 240 bez
(ette adorable personne c'est toi) 0 ecrire

65 175 110 200 120 200 125 190 bez
(sous le gran) 0 ecrire

125 185 140 170 170 190 200 240 bez
(d chapeau canotier) 0 ecrire

/F-Zapf findfont 12 scalefont setfont
55 170 60 130 85 105 110 110 bez
( V o i c i  l' o v a l e) 0 ecrire
110 110 120 120 130 130 140 170 bez
( de la f i g u r e) 0 ecrire

/F-Zapf findfont 10 scalefont setfont
100 175 115 178 segment%100 -> 95
(oeil) 0 ecrire

88 160 moveto (n) show

```

90 155 moveto (e) show
91 150 moveto (z) show

80 130 100 130 100 130 110 140 bez
(tabouche) 0 ecrire

/F-Zapf findfont 11 scalefont setfont

70 90 moveto gsave 1 3 scale (t) show grestore
(t) stringwidth pop 0 rmoveto (on) show

60 80 moveto (voi) show
50 70 moveto (ci enfin) show
45 60 moveto 2.5 0 (l'impar) ashow
40 45 95 50 segment
(faite image) 0 ecrire
30 30 100 40 segment
(de ton buste a) 0 ecrire
25 18 70 20 70 20 110 30 bez
(dor\216 vu comme) 0 ecrire
25 8 70 10 70 10 110 20 bez
(\213 travers un nuage) 0 ecrire

120 100 moveto gsave 1 3 scale (C) show grestore
120 100 140 105 150 100 160 95 bez

(ou exquis) 0 ecrire
127 92 160 88 segment
(un peu) 0 ecrire
125 80 moveto (plus bas) show
130 70 moveto (c'est ton) show
130 60 moveto (c\372ur) show
130 50 moveto (qui) show
125 38 moveto (bat) show

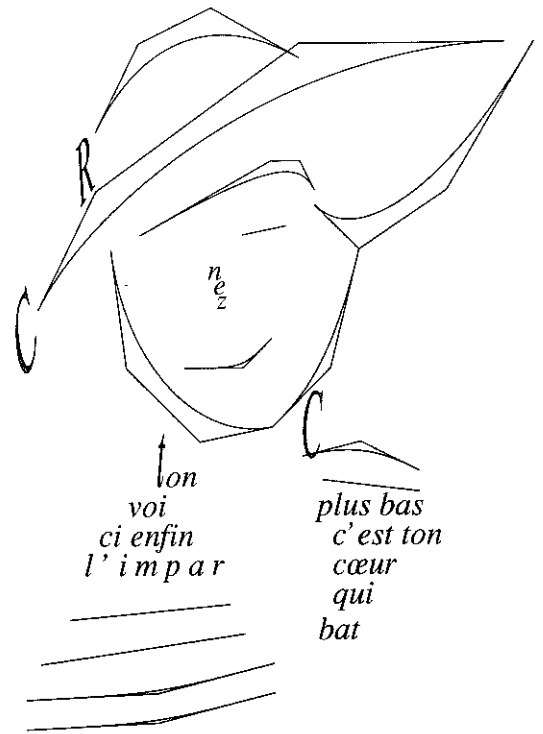


Fig. 3. "Skeleton" - test is true

Jacques André is a regular contributor to the PSLJI, and works for IRISA/INRIA-Rennes, France.

**The British Computer Society
Electronic Publishing Specialist Group**



**The Group to join
for Independent Discussion
of All Aspects of
Electronic Publishing**

Recent Meetings

Publishing from Databases
Mathematics and Non-Latin Typefaces
Paperless Publishing
Colour – The Black Art
CD-ROM and Multimedia Publishing

Next Meeting: 7 February 1991, London

**Document Image
Processing**

Annual Conference: 21–22 March 1991, Oxford

Subscription £10 per annum (£5 to BCS Members)

Further Information from:

BCS EPSG Office
30 Edgerton Road, Huddersfield HD3 3AD
Telephone: 0484 519462



**We're
here
to help
you**

If you're a PageMaker user you'll find plenty of useful information, and a sympathetic ear when you need one, at the PageMaker User Group.

The one thing every PMUG member has in common is the desire to make the best possible use of the program. The user group can help to achieve this aim through our:

- ▲ Bi-monthly magazine
- ▲ Bi-monthly meetings
- ▲ Bulletin board
- ▲ Membership directory
- ▲ Technical support network
- ▲ Co-operative working relationship with Aldus

Why not give us a ring on one of the numbers below, and we'll tell you what we can do to help. Or you can sample the bulletin board by dialling 081 655 0164.

Graham Jones	0635 71802
Steve Dick	081 677 1681
Pat Bitton or	
Angela West	081 691 2735

**PageMaker User Group,
PO Box 43, Newbury, Berkshire RG13 4WH**