The ease of including graphics in TEX documents using 4TEX

Wietse Dol *

Landbouw-Economisch Instituut (LEI-DLO)
P.O.Box 29703
2502 LS Den Haag
The Netherlands
w.dol@lei.agro.nl

Abstract

In this paper we will show how 4TEX can be used to incorporate graphics in TEX documents. Using 4TEX it not only becomes easy to include graphics in TEX files, but also makes it possible to preview and print TEX documents (including graphics) on any machine and any printer. For inserting graphics in TEX documents, 4TEX uses the style file `figures.sty`, the shareware programs TEXCAD and GRAPHIC WORKSHOP, and the freeware programs: HP2XX, BM2FONT, PCLTOMSP, and GHOSTSCRIPT. All these programs are discussed and we also discuss how 4TEX uses the strength of these programs to incorporate graphics in TEX documents.

1 Introduction

TEX has been developed with the idea that it should be possible to have a TEX implementation for every operating system (MS-DOS, VMS, VM-CMS, UNIX etc.). Another feature of TEX is that documents can be freely exchanged between operating systems (because documents are written in standard ASCII). Graphics, however, are machine dependent and the possibility to include graphics in TEX or LATEX depends on the operating system and the DVI-driver you are using. This means that including graphics in TEX or LATEX is often not an easy job.

The solution often used for including graphics is including PostScript pictures in the document using the `\special` command. The `\special` command is ignored/passed on by the TEX compiler but the PostScript DVI-driver will use the `\special` command to insert the PostScript picture at the right place and in the right size in your document. The advantage of this method is that for all operating systems there are PostScript DVI-drivers and that PostScript files are also written in standard ASCII, therefor you can transport text file and graphics to all operating systems. The disadvantage of this method is that you can *only* include PostScript pictures in your document and that you need a PostScript printer to produce output. It is not possible to use the screen previewer to view the DVI-file.

When you have a PC there are other ways to incorporate pictures in TEX documents. Before we will discuss them, we have to know more about the different types of pictures. In principle there are two types of pictures, namely bitmap and vector pictures. A bitmap picture is a matrix with the entries corresponding to points with a color. The dimension of the matrix specifies the height and width of the picture. Because of the fixed matrix, manipulating the picture is difficult and resizing the picture often leads to undesirable results. However, many MS-DOS graphic packages produce pictures in a bitmap format. These bitmap files come in many different types, mostly as a result of different compression and color encoding techniques. Examples of bitmap pictures are: GIF (Compuserve), TIFF, PCX (PC Paintbrush), BMP (Windows 3.x), IFF (Amiga), LBM (Amiga), IMG (Ventura), CUT (Dr Halo), and PCL (Hewlett Packard).

A *vector* picture is specified by a device-independent mathematical description and is therefore easy to manipulate/resize. However, the problem with vector pictures is that most DVI-drivers cannot handle them. Examples of vector pictures are: HPGL (Hewlett Packard Graphics Language), PS (PostScript), and EPS (Encapsulated PostScript).

In the sequel of this paper we will discuss the computer programs TEXCAD, GRAPHIC WORKSHOP, HP2XX, BM2FONT, PCLTOMSP, and GHOSTSCRIPT. We will end this paper by describing how 4TEX combines the strength of all these programs to incorporate graphics in TEX documents.

2 TEXcad

TEXCAD is a drawing program written by G. Horn for producing drawings in LATEX documents. It allows the objects available in the LATEX picture environment to

*The author thanks his former colleagues at the University of Groningen: Maarten van der Vlerk and Erik Frambach for their useful suggestions, support, and their effort in creating 4TEX.

be drawn and edited. Its output is a sequence of LATEX picture commands which can be inserted into a LATEX document to generate the drawing. The advantages of this program are: The text font used in the graphic is the same as the text font used in the main text (i.e. Computer Modern). Using the LATEX picture environment also makes it possible to compile/print/view these graphics on other operating systems. The disadvantages are its user unfriendlyness as compared to other graphical packages, and the limited set of objects of the LATEX picture environment.

TEXCAD is written in TURBO PASCAL V5 and runs on all PC machines. A mouse is not required but strongly recommended. TEXCAD supports the special commands of EMTEX for line drawing (very useful for drawing lines at any angle) but you should not forget to include the style file `emlines2.sty` in the document style declaration of the document. Likewise, if you use *bezier* curves. The problem with these EMTEX specials, however, is that they are machine dependent. For a detailed discussion how to install and use TEXCAD see Horn (1990). For an example of TEXCAD we refere to Figure 5.

3 emTEX and the SPECIAL commands

The `\special` command is ignored completely by the TEX compiler but is executed when running the DVI-driver. Not all DVI-drivers can execute all the `\special` commands. The `\special` command is therefore output device dependent and it is ignored when the DVI-driver does not support that specific `\special` command. The advantage of the `\special` commands is that you can use device-dependent instructions to produce output, e.g., you can use a Postscript printer to include PostScript pictures in your TEX document.

EMTEX has some `\special` commands to include PCX (PC Paintbrush), MSP (Microsoft Paint) and black and white BMP (Windows 3.x) bitmap pictures. The syntax of the `\special` command is:
\special{em:graph [path] <bitmap file>}
where *[path]* is an optional path and *<bitmap file>* is an PCX, MSP, or black and white BMP bitmap file. The upper left corner of the graphic file is located at the reference point of a character. Run length encoded BMP files and 4 color CGA-mode PCX files cannot be used. All non-white pixels of a PCX file are printed (assuming the standard palette). The width of the graphic must not exceed 32760 pixels, the height must not exceed 32766 pixels. The viewer and the printer drivers of the EMTEX package will show and print the bitmap. However, it is not possible to manipulate the picture. This means that different drivers will produce different sized pictures (as a result of the resolution of the device driver and the fixed resolution of the bitmap graphic). For a detailed discussion about `\special` commands see Mattes (1991).

When you want to manipulate the picture (e.g. resizing the picture or color reduction) you can for instance use the shareware program Graphic Workshop. Graphic Workshop is a program for working with bitmapped graphic files. It will handle most of the popular bitmap formats. Graphic Workshop is a simple, menu driven environment which will let you perform the following operations on graphic files:

- View bitmap pictures
- Convert between any two bitmap formats
- Print the bitmap on almost all popular printers
- Dither color pictures to black and white
- Reverse the colors
- Rotate the picture
- Flip the picture
- Scale the picture to any size
- Select a part of the graphic as a new picture
- Reduce the number of colors
- Sharpen and Soften the picture
- Adjust the brightness, contrast and color balance of the bitmaps.

When the picture has more than 32 colors and you use the EMTEX special command to include a graphic an error will appear when viewing or printing the picture. Too many colors will result in too black and unfocused pictures and it is much better to use BM2FONT to incorporate the picture into a TEX document.

The problem with bitmap files is their fixed dimension, i.e. the size of the picture will depend on the DVI-driver used. Suppose the bitmap file `golfer.pcx` has dimensions 550×770 dots. Using the picture and a 300 dpi DVI-driver the picture will have a width of $550/300 = 1.83$inch $= 47$mm and a height of $770/300 = 2.57$inch $= 65$mm. Other DVI-drivers will result in different sizes. The bitmap file is now printed using:

```
\begin{figure}[htbp]                %1
\begin{center}                      %2
 \setlength{\unitlength}{1mm}       %3
 \begin{picture}(47,65)             %4
  \put(0,65){%                      %5
  \special{em:graph golfer.pcx}}    %6
 \end{picture}
\end{center}
\caption{This is an example how to
         use pictures in \emtex}    %7
\end{figure}
%1 for the figure environment see
%  Lamport (1986) pages 59, 176
%  [htbp] lists admissible locations:
%  h=here, t=top of page, b=bottom of
%  page, p=on separate page
%2 center the picture horizontally
%3 we use millimeters as the unit of
%  measurement
%4 for the picture environment see
%  Lamport pages 101-111
%5 normally the picture golfer.pcx
%  will be printed from the left upper
%  corner of the picture box, but it
%  should be printed from the left lower
```

```
%  corner, therefore we need \put(0,65)
%6 \special{...} is the emTeX special
%  command to include a bitmap picture
%7 produces a numbered caption on the
%  place where it is issued
```

This will result in:

Figure 1: *This is an example how to use pictures in* EMTEX

EMTEX also supports the PostScript `\special` command, but it will not view or print a PostScript graphic file, except when you use a PostScript DVI-driver (e.g., DVIPS) to print on a PostScript printer.

There are many macro packages that can help you insert PostScript files in your document, e.g. PSFIG a macro package written by T.J. Darrel. With the help of a PostScript DVI driver, figures are automatically scaled and positioned on the page, and the proper amount of space is reserved. To include a Postscript picture, include the PSFIG style at the top of your document:

```
\documentstyle[11pt,psfig]{article}
```

and, when you wish to include a figure, invoke the macro with, e.g.,

```
\begin{figure}[htbp]
\begin{center}
  \psfig{figure=%
      tiger.ps,width=50mm,height=50mm}
\end{center}
\caption{The use of PSFIG to insert
         a PS picture}
\end{figure}
```

and the result is shown in Figure 2.

Note that spaces in the arguments of the macro are not allowed. For a detailed discussion of all possibilities (e.g., rotation, scaling etc.) we refer to Darrel (1992) and Goossens (1993). The PSFIG-macro will generate some `\special` commands to claim the correct space and size, and with a PostScript DVI driver the picture will be printed correctly.

Figure 2: *The use of PSFIG to insert a PS picture*

4 BM2Font

BM2FONT is a program written by F. Sowa and is used to convert bitmap pictures to TEX fonts. These TEX fonts can be read by the DVI-drivers and are used to view and print pictures. BM2FONT can convert the following bitmap pictures: PCX, GIF, BMP, IFF, LBM, TIFF, IMG, and CUT. For a detailed discussion how BM2FONT works and all the possible parameters we refer to the manual. Note that BM2FONT can produce several TEX fonts (i.e. bitmap fonts (extension `.pk`) and TEX font metric files (extension `.tfm`) and that the bitmap fonts are resolution dependent.

The command syntax of BM2FONT is
bm2font <bitmap file> [options]
The result of *bm2font example.pcx* is one or more font files, but also a file called `example.tex`. This file `example.tex` (written to the current directory) uses the picture fonts and defines a macro called `\setexample` (i.e. consisting of the the word SET and the filename EXAMPLE (without file extension)). The picture is now produced simply by giving the command `\input example.tex` and the command `\setexample` on the location where you want the picture.

We will end this section with an example. Suppose we have a BMP bitmap file `genesis.bmp` and we want to convert this bitmap to TEX fonts for the laserprinter (300 dpi). Running BM2FONT
bm2font genesis.bmp -h300 -v300 -m50 -n50
will result in one TEX font metric file (`genesisa.tfm`), one laserfont (`genesisa.pk`), and the TEX file `genesis.tex` containing the macro `\setgenesis`. The parameters *-h* and *-v* are the horizontal and vertical resolution of the printer, the parameters *-m* and *-n* are the width and height you want the picture to be in millimeters. To produce the picture in the TEX document we can use, e.g.:

```
\begin{figure}[htbp]
\centerline{\input genesis \setgenesis}
%the file genesis.tex contains the
%                     macro \setgenesis
\caption{This is the file genesis.bmp
         converted to \TeX\ fonts}
\end{figure}
```

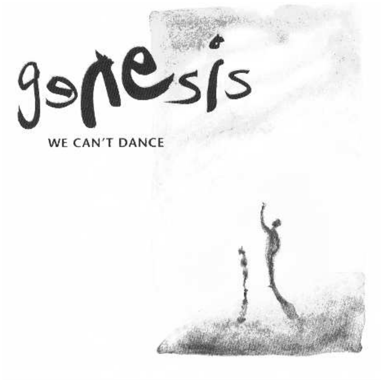

Figure 3: *This is the file genesis.bmp converted to TEX fonts*

BM2FONT gives excellent results and is easy to use. The only disadvantage is that you need to generate picture fonts for every printer you use. For example when you use a 300dpi laserprinter as well as a 600dpi PostScript printer, BM2FONT will give you `.pk` and `.tfm` files with the same name but for different printers. So you have to do your own bookkeeping, which fontfiles you need for the printer you are using at the moment.

Note: if the length of the filename of the picture file is eight characters, the last character will be omitted for the construction of the `.tfm` and `.pk` files (because of the addition of the font numbers `a`, `b`, ...). For instance `scrndump.pcx` will produce `scrnduma.tfm`, `scrnduma.pk`, and `scrndum.tex` and the macro is called `\setscrndum`.

Note: No digits are allowed in the picture filename. E.g., `screen1.pcx` has to be renamed to `screeni.pcx`.

When you have a PCL bitmap file (Hewlett Packard LaserJet and DeskJet graphic output) you can convert this file to a MSP or a PCX bitmap. The conversion is done by using E. Mattes conversion program `pcltomsp`. After conversion you can use the special command to include the graphic (see Section 3) or the program BM2FONT to generate TEX fonts. The syntax is easy, e.g. *pcltomsp -qop graph17.lj graph17*
converts the PCL file `graph17.lj` into the PCX file `graph17.pcx`, and does not display warnings and the program title.

5 Hewlett Packard Plotter files and HP2xx

Some graphics programs produce graphic files in the *HPGL* format (Hewlett Packard Graphic Language). These are vector pictures specifically made for Hewlett Packard plotters. Because we can only use PostScript and certain bitmap pictures in TEX documents we need to convert such files. The extensions often used for HPGL files are `.hpp`, `.plt`, and `.hpg` and are produced by, e.g., the programs MATLAB, GAUSS, and HARVARD GRAPHICS. To convert HPGL plotter files we use the program HP2XX.

HP2XX is a freeware program from H. Werntges and is used to print, view and convert HPGL plotfiles. We refer to the documentation for a detailed discussion of all the possibilities (e.g., rotation, picture size, pencolor, magnification etc.). 4TEX uses HP2XX to convert HPGL files to *PCX* bitmap files and *EPS* (Encapsulated PostScript) files. HP2XX uses no environment variables, it reads and writes the files from the current directory. HP2XX supports the 800×600 super VGA modes (e.g., the Tseng ET4000 and the Trident SVGA). HP2XX is easy to use and produces excellent quality. You can convert HPGL pictures to MF (METAFONT format), CAD (to be used with TEXCAD), EM (EMTEX specials), EPIC (the Enhanced Picture style), IMG-, PBM-, PCL- and PCX-bitmaps and EPS PostScript pictures.

Suppose we have a file `example.hpg`. We can convert this file to a 300 dpi PCX bitmap file `example.pcx` with height 100 mm (width is automatically calculated) using the command
hp2xx -mpcx -d300 -h100 -f example.pcx example.hpg
Instead of converting the picture to a PCX bitmap we can also convert it to an Encapsulated PostScript file (use *-meps* and *-f example.eps* instead of *-mpcx* and *-f example.pcx*). After conversion we can proceed as discussed in the Sections 3 and 4.

As an example we show a Lotus picture that is printed as a HPGL file and then converted to a 300dpi PCX bitmap and a EPS picture using the syntax described above. After conversion we used the style file `figures.sty` (see Section 7) to print this picture:

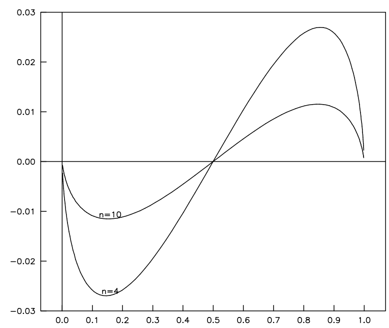

Figure 4: *A HPGL picture converted to PCX or EPS*

6 PostScript and GhostScript

If you want to view, print and manipulate PostScript files and you do not have a PostScript printer (or commercial software), we suggest to use the freeware program GHOSTSCRIPT from Aladdin Enterprises. Using GHOSTSCRIPT you can view and print `.ps` and `.eps` PostScript files on any screen and any printer. GHOSTSCRIPT also supports the Tseng ET4000 and the Trident graphics card for viewing in super VGA mode. You can resize the picture to any length and width. You can also calculate the BoundingBox and convert the Postscript picture to a PCX bitmap. This PCX bitmap can be used with the DVI drivers of EMTEX to view and print the picture in TEX documents without using a PostScript DVI-driver (see Section 3). For a detailed discussion how GHOSTSCRIPT works we refer to the GHOSTSCRIPT documentation. Note that GHOSTSCRIPT is freeware, is regularly updated and gives good results. The only disadvantage is perhaps that it is not very user friendly.

7 The FIGURES style files

The style file FIGURES is a modification of the PSFIG style file and is used in 4TEX. It combines the possibility to print/view PCX pictures with the EMTEX special commands, and to print PostScript files as with PSFIG. By default (or when using the command `\pcx`) FIGURES will try to use PCX picture files. When not found or when using the command `\postscript`, FIGURES will look for Postscript files (`.eps` and `.ps`). This makes it possible to view/print pictures using the EMTEX special commands and to print the pictures in a TEX document on any printer. To include a picture include the FIGURES style option at the top of your document:

```
\documentstyle[11pt,figures]{article}
```

and, when you wish to include the figure (`example.pcx` or `example.eps`), call the macro like this:

```
\putfigure{%
    figure=example,width=2in,height=3in}
```

Note that the extension of the picture file is not specified. All commands defined in the style file PSFIG are also available (see e.g. Goossens (1993)). Some names of the macros of PSFIG are changed, e.g. `\psfig` is changed in `\putfigure` and some extra macros are added, e.g. `\pcx` (use PCX files), `\figurefull` (the same as `\psfull`); `\figuredraft` (the same as `\psdraft`).

8 4TEX

In the preceding sections we have discussed several ways to incorporate graphics in TEX documents. These sections are summarized by a flow diagram (see Figure 5).

The main problem of a user who wants to include a picture is that he/she needs to know which program to use and which parameters and commands one needs before one gets reasonable output. The 4TEX workbench is developed to shield you from these dirty bits. The aim of 4TEX is a simple menu based interface that lets the user choose between all available TEX related programs and give some help wherever needed.

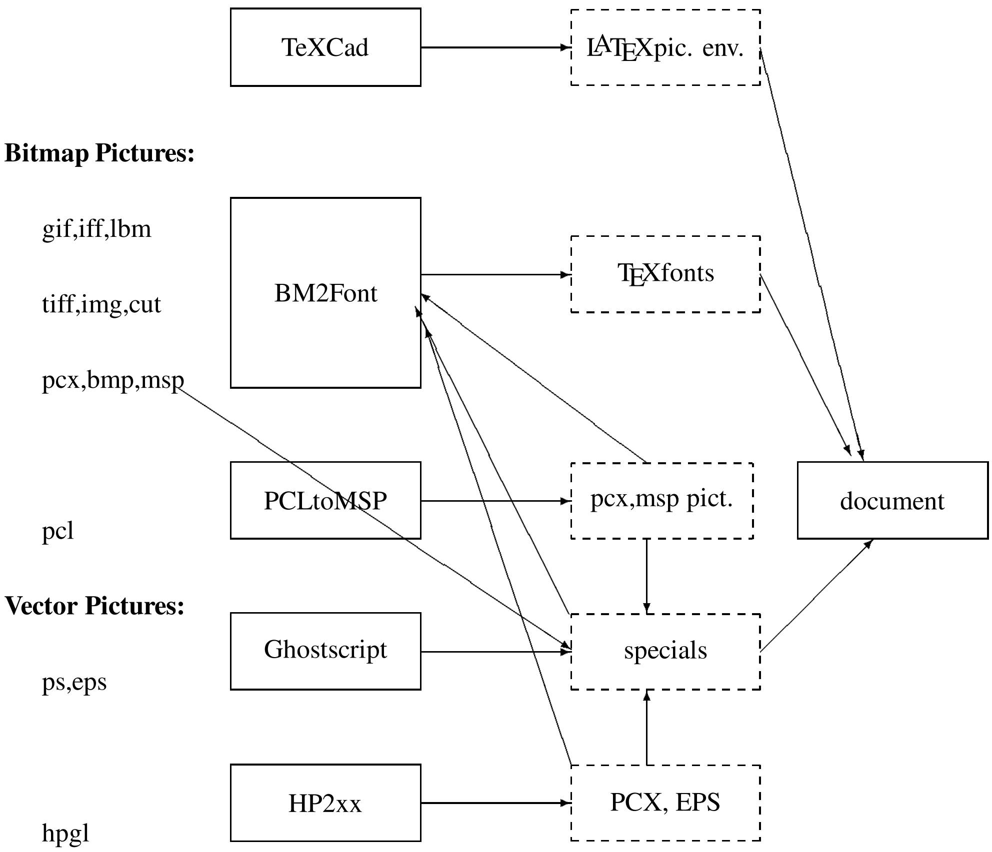

Figure 5: *How to insert graphics? (example of* TEXCAD*)*

The *GRAPHICS* utility of 4TEX helps you incorporating pictures. All the programs discussed above are used, but a user does not need to remember all the (program specific) parameters. Simply choosing from an options list one can specify the parameters. Often this is not necessary because most parameters are set automatically (e.g. the printer resolution). Often the user only needs to specify the size of the picture and then *convert* the picture. Converting means that the correct programs are called and that the result is a TEX file that can be used to insert the picture in your document. When for instance we convert the file `acad.hpp`, 4TEX will end the conversion by telling the user that the picture is inserted simply by adding the style file `figures.sty` and using the statement:

```
\begin{figure}[h]
  \centerline{\input ACAD}
  \caption{your title}
\end{figure}
```

By this way a uniformly approach of inserting pictures is reached, i.e. it does not matter if you use HP2XX, BM2FONT, PCLTOMSP, or GHOSTSCRIPT. After conversion only one simple input statement is enough.

4TEX also makes it possible to view and manipulate the picture. Viewing is also possible in super VGA modes. For instance a postscript picture can be rotated, a BoundingBox can be calculated and 4TEX magnifies the picture so that it will fit the specified size exactly. The same holds for HPGL pictures. Bitmap pictures can be manipulated using GRAPHIC WORKSHOP. 4TEX, however, uses the shareware program `cshow` to view bitmaps.

All the necessary bookkeeping is done by 4TEX. For instance the fonts (`*.pk` and `*.tfm`) are stored in the correct directories and the conversion file is stored on the current working directory.

The conversions of pictures is done in such a way that it allows you to view and print the TEX document with the pictures on any screen or printer. When you want to use a PostScript printer you simple need to include the command `\postscript` in the document. The conclusion therefor must be that with 4TEX the inclusion of graphics has become an easy job.

References

Darrel, T. (1992), PSFIG *1.8 users guide*, available from public file-servers.

Goossens, M. (1993), ‘Postscript en LATEX, de komplimentariteit in de praktijk’, *MAPS Nederlandstalige TEX Gebruikersgroep*, **93.1**, 101–113.

Horn, G. (1990), *TEXcad Version 2.8*, available from public file-servers.

Lamport, L. (1986), *LATEX, a Document Preparation System*, Addison-Wesley, Reading, Massachusetts.

Mattes, E. (1989), *How to run LATEX with* EMTEX, available from public file-servers.

Mattes, E. (1990), DVIDOT *1.3i Benützer Handleitung*, available from public file-servers.

Mattes, E. (1992), EMTEX *manual*, available from public file-servers.

Mattes, E. (1992), DVIDRV *1.4s manual*, available from public file-servers.

Mattes, E. and L. Lamport (1989), *Using LATEX with* EMTEX, available from public file-servers.

Rokicki, T. (1992), DVIPS*: A TEX Driver (5.495)*, available from public file-servers.

Sowa, F. (1991), *Benutzershandbuch BM2FONT*, available from public file-servers.