
Subject: Re: EPS fragment output for latex font interpretation?

Posted by [Paolo Grigis](#) on Wed, 25 Jul 2007 10:25:19 GMT

[View Forum Message](#) <> [Reply to Message](#)

Gernot Hassenpflug wrote:

> Hello all,
>
> I have used IDL to output EPS figures with whatever fonts IDL has
> built-in. Now I would like to use IDL to output EPS fragments (in
> vector format) with markers for the text information, and .tex files
> containing the text corresponding to the markers, with both files to
> be later interpreted by LaTeX with an \input statement. With matlab,
> which I do not currently have a licence to, this was possible using
> the user-contributed laprint program.
>
> With such a splitting of the tasks, it is easy to adjust text size and
> font to what is required in the LaTeX document.
>
> Is there any way to do such a task in IDL (I have 6.2 and 6.3 licences
> at my workplace).
>
> Alternatively, if there is a way to use Computer Modern fonts with
> IDL, I could try to use those in the EPS file directly (then output in
> binary format is fine too).
>
> Many thanks
> Gernot

I've done so in a few occasion, and it worked out fine.

Here's an example with IDL, LaTeX and unix commands, which
I put together with some help from this newsgroup a while ago:

IDL commands to generate a figure with b and t as text markers:

```
N=1000
```

```
x=findgen(N)/(N-1)*10-5
```

```
mu=0.
```

```
sigma=1.
```

```
y=total(exp(-(x-mu)^2/(2*sigma^2)),/cumulative)*(x[1]-x[0])
```

```
set_plot,'PS'
```

```
device,filename='fig1.eps',/encapsulated,xsize=12.,ysize=8.
```

```
!p.font=0
```

```
plot,x,y,yrange=[0.,3.],thick=4,/xstyle,xrange=[-5,5],xtitle='x',ytitle='y'
oplot,!X.crangle,sqrt(2*!Pi)*sigma*[1,1],linestyle=2
```

```
xyouts,0.,1.,'t',/data
xyouts,-3,2.65,'b'
device,/close
set_plot,'X'
```

;END IDL

LaTeX file:

```
\documentclass{article}
\usepackage{geometry}
\usepackage{graphicx}
\usepackage{psfrag}
\pagestyle{empty}
\geometry{paperwidth=12.1cm,paperheight=8.1cm,margin=0pt}
\begin{document}
\psfrag{t}[l][1.35]{$\displaystyle y=\lim_{-\infty}^x
e^{-\left(\frac{t-\mu}{\sqrt{2}\sigma}\right)^2},\mathrm{d}t$}
\psfrag{x}[c][1.75]{$x$}
\psfrag{y}[c][1.75]{$y$}
\psfrag{b}[c][1.35]{$\displaystyle y=\sqrt{2\pi}\sigma$}
\includegraphics[width=11.9cm,height=7.9cm]{fig1.eps}
\end{document}
```

%end latex

Linux command to TEX and transform into eps:

```
latex doc.tex
dvips -o fignew.ps doc.dvi
ps2epsi fignew.ps fignew.epsi
perl -ne 'print unless /^%%BeginPreview/..^%%EndPreview/' < fignew.epsi > fignew.eps
rm fignew.epsi
```

Now you should have an eps file with equations rather than "t" and "b" labels...

Hope this helps,
Paolo Grigis

Subject: Re: EPS fragment output for latex font interpretation?
Posted by [Gernot Hassenpflug](#) on Wed, 25 Jul 2007 11:11:47 GMT

Paolo_Grigis <pgrigis@astro.phys.ethz.ch> writes:

```
> Gernot Hassenpflug wrote:
>> Hello all,
>>
>> I have used IDL to output EPS figures with whatever fonts IDL has
>> built-in. Now I would like to use IDL to output EPS fragments (in
>> vector format) with markers for the text information, and .tex files
>> containing the text corresponding to the markers, with both files to
>> be later interpreted by LaTeX with an \input statement. With matlab,
>> which I do not currently have a licence to, this was possible using
>> the user-contributed laprint program.
>>
>> With such a splitting of the tasks, it is easy to adjust text size and
>> font to what is required in the LaTeX document.
>>
>> Is there any way to do such a task in IDL (I have 6.2 and 6.3 licences
>> at my workplace).
>>
>> Alternatively, if there is a way to use Computer Modern fonts with
>> IDL, I could try to use those in the EPS file directly (then output in
>> binary format is fine too).
```

/../

```
> I've done so in a few occasion, and it worked out fine.
> Here's an example with IDL, LaTeX and unix commands, which
> I put together with some help from this newsgroup a while ago:
```

Many thanks! That is great news, my search of this group had turned up only mention of LaTeX package psfrag in 1997...

```
> IDL commands to generate a figure with b and t as text markers:
>
> N=1000
>
> x=findgen(N)/(N-1)*10-5
>
> mu=0.
> sigma=1.
>
> y=total(exp(-(x-mu)^2/(2*sigma^2)),/cumulative)*(x[1]-x[0])
>
> set_plot,'PS'
> device,filename='fig1.eps',/encapsulated,xsize=12.,ysize=8.
> !p.font=0
```

I'll try this at work in the morning, for use. Is the setting of font to 0 one of the critical tricks?

```
> plot,x,y,yrange=[0.,3.],thick=4,/xstyle,xrange=[-5,5],xtitle ='x',ytitle='y'
> oplot,!X.crange,sqrt(2*!Pi)*sigma*[1,1],linestyle=2
>
> xyouts,0.,1.,'t',/data
> xyouts,-3,2.65,'b'
> device,/close
> set_plot,'X'
```

OK, understood.

```
> ;END IDL
>
> LaTeX file:
>
> \documentclass{article}
> \usepackage{geometry}
> \usepackage{graphicx}
> \usepackage{psfrag}
> \pagestyle{empty}
> \geometry{paperwidth=12.1cm,paperheight=8.1cm,margin=0pt}
```

Excellent! So the below could be put into a separate .tex file and \input into the main document as I am used to doing under Matlab with the laprint program.

```
> \begin{document}
> \psfrag{t}[l][l][1.35]{$\displaystyle y=\int_{-\infty}^x
> e^{\left(\frac{t-\mu}{\sqrt{2}}\sigma\right)^2}\,\mathrm{d} t$}
> \psfrag{x}[c][l][1.75]{$x$}
> \psfrag{y}[c][l][1.75]{$y$}
> \psfrag{b}[c][l][1.35]{$\displaystyle y=\sqrt{2\pi}\sigma$}
> \includegraphics[width=11.9cm,height=7.9cm]{fig1.eps}
> \end{document}
```

Understood.

```
> %end latex
>
>
> Linux command to TEX and transform into eps:
>
> latex doc.tex
> dvips -o fignew.ps doc.dvi
```

Yes, OK.

```
> ps2epsi fignew.ps fignew.epsi
> perl -ne 'print unless /^%%BeginPreview/..^%%EndPreview/' < fignew.epsi > fignew.eps
> rm fignew.epsi
```

Ah, this part is new to me. Any idea what this is repairing, since you've already specified emcapsulated PS in your IDL program. (I understand the perl part after that to remove the preview TIFF image.)

```
> Now you should have an eps file with equations rather than "t" and "b" labels...
>
> Hope this helps,
> Paolo Grigis
```

Fantastic! I'll let you know how this goes.

Best regards,
Gernot

--

Gernot Hassenpflug, NICT, Tokyo

Subject: Re: EPS fragment output for latex font interpretation?

Posted by [Paolo Grigis](#) on Wed, 25 Jul 2007 12:07:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

Gernot Hassenpflug wrote:

```
> Paolo_Grigis <pgrigis@astro.phys.ethz.ch> writes:
```

```
>
```

```
>> Gernot Hassenpflug wrote:
```

```
>>> Hello all,
```

```
>>>
```

```
>>> I have used IDL to output EPS figures with whatever fonts IDL has
>>> built-in. Now I would like to use IDL to output EPS fragments (in
>>> vector format) with markers for the text information, and .tex files
>>> containing the text corresponding to the markers, with both files to
>>> be later interpreted by LaTeX with an \input statement. With matlab,
>>> which I do not currently have a licence to, this was possible using
>>> the user-contributed laprint program.
```

```
>>>
```

```
>>> With such a splitting of the tasks, it is easy to adjust text size and
>>> font to what is required in the LaTeX document.
```

```
>>>
```

```
>>> Is there any way to do such a task in IDL (I have 6.2 and 6.3 licences
>>> at my workplace).
```

```
>>>
```

```
>>> Alternatively, if there is a way to use Computer Modern fonts with
>>> IDL, I could try to use those in the EPS file directly (then output in
```

```

>>> binary format is fine too).
>
> /../
>
>> I've done so in a few occasion, and it worked out fine.
>> Here's an example with IDL, LaTeX and unix commands, which
>> I put together with some help from this newsgroup a while ago:
>
> Many thanks! That is great news, my search of this group had turned up
> only mention of LaTeX package psfrag in 1997...
>
>> IDL commands to generate a figure with b and t as text markers:
>>
>> N=1000
>>
>> x=findgen(N)/(N-1)*10-5
>>
>> mu=0.
>> sigma=1.
>>
>> y=total(exp(-(x-mu)^2/(2*sigma^2)),/cumulative)*(x[1]-x[0])
>>
>> set_plot,'PS'
>> device,filename='fig1.eps',/encapsulated,xsize=12.,ysize=8.
>> !p.font=0
>
> I'll try this at work in the morning, for use. Is the setting of font
> to 0 one of the critical tricks?

```

Yes, I think it is. The point is that latex need to search and replace your text labels, but it cannot find them if they are encoded as anything different than plain text. Using device fonts should ensure that. I am not sure exactly how to do the same using true type font...

```

>
>> plot,x,y,yrange=[0.,3.],thick=4,/xstyle,xrange=[-5,5],xtitle ='x',ytitle='y'
>> oplot,!X.crange,sqrt(2*!Pi)*sigma*[1,1],linestyle=2
>>
>> xyouts,0.,1.,'t',/data
>> xyouts,-3,2.65,'b'
>> device,/close
>> set_plot,'X'
>
> OK, understood.
>
>> ;END IDL
>>
>> LaTeX file:

```

```

>>
>> \documentclass{article}
>> \usepackage{geometry}
>> \usepackage{graphicx}
>> \usepackage{psfrag}
>> \pagestyle{empty}
>> \geometry{paperwidth=12.1cm,paperheight=8.1cm,margin=0pt}
>
> Excellent! So the below could be put into a separate .tex file and
> \input into the main document as I am used to doing under Matlab with
> the laprint program.
>
>> \begin{document}
>> \psfrag{t}[][][1.35]{$\displaystyle y=\int_{-\infty}^x
>> e^{-\left(\frac{t-\mu}{\sqrt{2}\sigma}\right)^2}\,\mathrm{d}t$}
>> \psfrag{x}[c]{}[1.75]{$x$}
>> \psfrag{y}[c]{}[1.75]{$y$}
>> \psfrag{b}[c]{}[1.35]{$\displaystyle y=\sqrt{2\pi}\sigma$}
>> \includegraphics[width=11.9cm,height=7.9cm]{fig1.eps}
>> \end{document}
>
> Understood.
>
>> %end latex
>>
>>
>> Linux command to TEX and transform into eps:
>>
>> latex doc.tex
>> dvips -o fignew.ps doc.dvi
>
> Yes, OK.
>
>> ps2epsi fignew.ps fignew.eps
>> perl -ne 'print unless /^%%BeginPreview/..!/^%%EndPreview/' < fignew.eps > fignew.eps
>> rm fignew.eps
>
> Ah, this part is new to me. Any idea what this is repairing, since
> you've already specified encapsulated PS in your IDL program. (I
> understand the perl part after that to remove the preview TIFF image.)

```

Well, this is not really necessary, don't bother. Since dvips produces a ps file, if you really want an eps you'll have to convert the ps to eps in whatever manner you like (on my machine I just happen to have ps2epsi available, so that's what I used). The reason for that is to have a separate, final eps file with the figure *and* the latex embellishments, which could afterwards be directly included into another (either latex or non-latex) document without need for additional processing.

You're right about removing the preview, just save some disk space.

Ciao,
Paolo

>
>> Now you should have an eps file with equations rather than "t" and "b" labels...
>>
>> Hope this helps,
>> Paolo Grigis
>
> Fantastic! I'll let you know how this goes.
>
> Best regards,
> Gernot

Subject: array subscript conversion
Posted by [Foldy Lajos](#) on Wed, 25 Jul 2007 15:36:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi guys,

according to the manual, array subscripts are converted to long (or long64 on 64 bit systems) before use if necessary, so an explicit conversion should not affect the result.

```
IDL> print, !version
{ x86 linux unix linux 6.3 Mar 23 2006    32    64}
IDL>
IDL> a=lindgen(10)
IDL> print, a[[long(-1ull)]]
      0
IDL> print, a[[-1ull]]
      9
```

Is it a bug or I am missing something?

regards,
lajos

Subject: Re: EPS fragment output for latex font interpretation?
Posted by [Gernot Hassenpflug](#) on Wed, 25 Jul 2007 15:39:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paolo_Grigis <pgrigis@astro.phys.ethz.ch> writes:

```
> Gernot Hassenpflug wrote:
>> Paolo_Grigis <pgrigis@astro.phys.ethz.ch> writes:
>> ./
>>> set_plot,'PS'
>>> device,filename='fig1.eps',/encapsulated,xsize=12.,ysize=8.
>>> !p.font=0
>>
>> I'll try this at work in the morning, for use. Is the setting of font
>> to 0 one of the critical tricks?
>
> Yes, I think it is. The point is that latex need to search and replace
> your text labels, but it cannot find them if they are encoded as anything
> different than plain text. Using device fonts should ensure that. I am
> not sure exactly how to do the same using true type font...
```

No worries, I'm very very happy to get the help I did! Worked beautifully (I temporarily installed a Windows version of IDL under linux at home using wine, and tested, happy to go to bed knowing it will work at the office).

```
>>> ./
>>> ps2epsi fignew.ps fignew.eps
>>> perl -ne 'print unless /^%%BeginPreview/..!/^%%EndPreview/' < fignew.eps > fignew.eps
>>> rm fignew.eps
>>
>> Ah, this part is new to me. Any idea what this is repairing, since
>> you've already specified emcapsulated PS in your IDL program. (I
>> understand the perl part after that to remove the preview TIFF image.)
>
> Well, this is not really necessary, don't bother. Since dvips produces a
> ps file, if you really want an eps you'll have to convert the ps to eps
> in whatever manner you like (on my machine I just happen to have
> ps2epsi available, so that's what I used). The reason for that is to
> have a separate, final eps file with the figure *and* the latex
> embellishments, which could afterwards be directly included into another
> (either latex or non-latex) document without need for additional processing.
```

Ah, yes, understood. In that case of course the fonts are now fixed, no longer modifiable by the document, as I understand it. But I think you are incorrect in one aspect: IDL produced an EPS file when you gave it the /encapsulated modifier for the device. Here the top lines from my file:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 340 226
%%Title: Graphics produced by IDL
```

%%For: gernot@hase, C:\RS\IDL60
%%Creator: IDL Version 6.0, Microsoft Windows (Win32 x86 m32)
...

Many thanks once again,
Gernot

--
Gernot Hassenpflug, NICT, Tokyo
