

Geogebra per \LaTeX

Luciano Battaia^(*)

1 novembre 2009

In questa breve nota propongo alcuni suggerimenti e idee per utilizzare al meglio la capacità di Geogebra di produrre immagini direttamente esportabili in formato PSTricks, e quindi di qualità comparabile al testo prodotto con \LaTeX .

Indice

Premessa	1
1 Il lavoro in Geogebra	1
2 Perfezionamento del codice	3
2.1 Posizionamento delle etichette	5
2.2 Spessore e stile delle linee e dei punti	7
3 L'inserimento delle figure nel documento \LaTeX	9
4 Conclusione	11

Premessa

Geogebra può esportare la vista grafica in file PSTricks oppure PGF/TikZ direttamente utilizzabili. Naturalmente, come per ogni conversione automatica da un formato all'altro, sono quasi sempre necessari piccoli adattamenti o modifiche per soddisfare particolari esigenze o, semplicemente, il proprio gusto estetico. Mi occupo qui in particolare della conversione in formato PSTricks, fornendo alcune indicazioni utili per migliorare il già accettabile risultato che si può ottenere in maniera automatica con Geogebra. Fornisco anche alcune indicazioni pratiche sulle tecniche adatte a inserire le immagini prodotte in un file \LaTeX .

L'idea di questo lavoro è nata da un intervento tenuto nel mese di ottobre 2009 presso il Liceo Scientifico Grigoletti di Pordenone, su invito del collega Nicolò Lomolino, per gli studenti della classe IIIB. Tutti gli argomenti trattati sono quindi proposti a un livello base, e le conoscenze di \LaTeX richieste sono minime.

1 Il lavoro in Geogebra

Geogebra è un software di geometria dinamica e di calcolo algebrico molto sofisticato e potente. La maggior parte della grafica prodotta è esportabile per essere compilata direttamente con

^{*}<http://www.batmath.it>

PSTricks, tuttavia ci sono alcune limitazioni dovute principalmente al tipo di funzioni predefinite nel linguaggio *Postscript* che PSTricks usa. Si ci si limita però alle costruzioni geometriche “con riga e compasso” e a funzioni sufficientemente “standard” non ci sono quasi mai problemi.

Supponiamo dunque di avere costruito alcune figure geometriche e magari di avere inserito qualche grafico di funzione nella finestra di Geogebra: nell’esempio riportato nella figura 1 è proposta un’immagine contenente una circonferenza, la tangente in un suo punto, un triangolo e una parabola.

La figura 1 è ottenuta selezionando una zona della finestra grafica di Geogebra ed esportando direttamente in formato eps⁽¹⁾. In molti casi questa potrebbe essere già una soluzione accettabile, ma presenta lo svantaggio che la figura così ottenuta non è modificabile (anche se la qualità è decisamente buona).

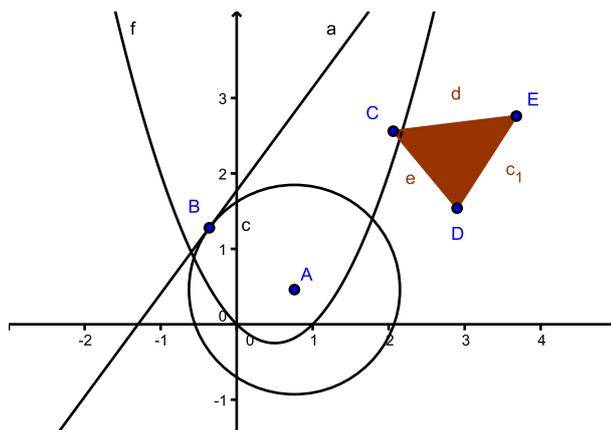


Figura 1 Una finestra grafica di Geogebra

Ritorniamo allora alla nostra finestra grafica di Geogebra e cominciamo con il rendere non visibili tutte le cose che non servono per l’immagine che vogliamo inserire nel nostro documento. Questa operazione si può anche fare successivamente, ma conviene farla, perché la cosa è più veloce, direttamente in Geogebra. In questo caso molto probabilmente non ci interesserà visualizzare le etichette dei vari oggetti costruiti (*c* per la circonferenza, *f* per la parabola, ecc.). Nel caso di una costruzione di geometria con riga e compasso, anche gli assi cartesiani sarebbero di troppo. Controlliamo inoltre la correttezza di tutte le etichette dei punti. I colori che Geogebra usa per la vista grafica non sono tra i più felici: secondo me è meglio, in questa fase, lasciarli come sono e modificarli poi nel codice PSTricks. In molti casi, soprattutto se il documento verrà stampato in bianco e nero, converrà eliminare (con la tecnica che vedremo) tutti i colori, e magari usare solo qualche tono di grigio.

Quando tutto è pronto, selezioniamo la parte di finestra che ci interessa ed esportiamo la *Vista grafica come PSTricks*. Conviene accettare le impostazioni di default proposte nella finestra, adattando magari solo la dimensione del carattere a quella che abbiamo usato nel nostro documento, invece di accettare i 10 pt propostici. Cliccando sul bottone *Genera codice PSTricks* otterremo un file come quello qui di seguito riportato.

```
\documentclass[10pt]{article}
```

¹Geogebra può esportare la finestra grafica (o una sua parte selezionata) in una serie di formati, tra cui **eps** e **pdf**. Come sarà spiegato in seguito, per l’inserimento in un file L^AT_EX, servono sia il formato **eps** che **pdf**. Conviene comunque utilizzare, se si decide di scegliere questa via, il formato **eps** e poi trasformare in **pdf** nel modo che sarà spiegato più avanti.

```

\usepackage{pstricks-add}
\pagestyle{empty}
\begin{document}
\newrgbcolor{zzttqq}{0.6 0.2 0}
\psset{xunit=1.0cm,yunit=1.0cm,algebraic=true,dotstyle=o,dotsize=3pt 0,
linewidth=0.8pt,arrowsize=3pt 2,arrowinset=0.25}
\begin{pspicture*}(-2.92,-1.76)(4.92,3.84)
\psaxes[labelFontSize=\scriptstyle,xAxis=true,yAxis=true,Dx=1,Dy=1,
ticksize=-2pt 0,subticks=2]{->}(0,0)(-2.92,-1.76)(4.92,3.84)
\pspolygon[linestyle=none,fillstyle=solid,fillcolor=zzttqq,opacity=0.1]
(2.06,2.56)(2.9,1.54)(3.68,2.76)
\pscircle(0.76,0.46){1.39}
\psplot{-2.92}{4.92}{(--1.45--1.12*x)/0.82}
\psline[linecolor=zzttqq](2.06,2.56)(2.9,1.54)
\psline[linecolor=zzttqq](2.9,1.54)(3.68,2.76)
\psline[linecolor=zzttqq](3.68,2.76)(2.06,2.56)
\psplot[plotpoints=200]{-2.92}{4.92}{x^2-x}
\psdots[dotstyle=*,linecolor=blue](0.76,0.46)
\rput[b1](0.84,0.58){\blue{A$}}
\psdots[dotstyle=*,linecolor=blue](-0.36,1.28)
\rput[b1](-0.64,1.48){\blue{B$}}
\psdots[dotstyle=*,linecolor=blue](2.06,2.56)
\rput[b1](1.7,2.72){\blue{C$}}
\psdots[dotstyle=*,linecolor=blue](2.9,1.54)
\rput[b1](2.82,1.12){\blue{D$}}
\psdots[dotstyle=*,linecolor=blue](3.68,2.76)
\rput[b1](3.82,2.9){\blue{E$}}
\end{pspicture*}
\end{document}

```

Se non sono state inserite etichette o caselle di testo particolari (per esempio con caratteri accentati) o formule matematiche non standard, basta copiare il file nell'editor L^AT_EX in uso e compilarlo. Se vogliamo evitare rischi, conviene introdurre le righe di codice seguente nel preambolo, subito sotto la riga `\documentclass[10pt]{article}`:

```

\usepackage[latin1]{inputenc} (Questo per Windows!)
\usepackage[applemac]{inputenc} (Questo per Mac!)
\usepackage[italian]{babel}
\usepackage{latexsym}
\usepackage{amsmath}
\usepackage{amssymb}

```

Nel seguito faremo solo riferimento alla versione per Windows.

2 Perfezionamento del codice

Come già accennato, i colori proposti da Geogebra non sono tra i più felici; inoltre Geogebra usa colori diversi a seconda del tipo di oggetto costruito, e a seconda che si tratti di *oggetto libero* oppure *oggetto dipendente*. Naturalmente questo ha poco interesse in un file per la stampa. Secondo me è meglio cancellare tutti i riferimenti ai colori, con l'eccezione del `fillcolor` nella

riga contenete l'istruzione `\pspolygon` (o in righe analoghe che qui non compaiono) in cui si può inserire, per esempio, l'opzione proposta da Geogebra con un `lightgray`, adatto come colore da riempimento. Inoltre `opacity=0.1` è in genere troppo chiara e conviene mettere almeno `opacity=0.3`.

Si otterrà un file come il seguente⁽²⁾.

```
\documentclass[10pt]{article}
\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}
\usepackage{latexsym}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{pstricks-add}
\pagestyle{empty}
\begin{document}
\psset{xunit=1.0cm,yunit=1.0cm,algebraic=true,dotstyle=o,dotsize=3pt 0,
linewidth=0.8pt,arrowsize=3pt 2,arrowinset=0.25}
\begin{pspicture*(-2.92,-1.76)(4.92,3.84)}
\psaxes[labelFontSize=\scriptstyle,xAxis=true,yAxis=true,Dx=1,Dy=1,
ticksize=-2pt 0,subticks=2]{->}(0,0)(-2.92,-1.76)(4.92,3.84)
\pspolygon[linestyle=none,fillstyle=solid,fillcolor=lightgray,opacity=0.3]
(2.06,2.56)(2.9,1.54)(3.68,2.76)
\pscircle(0.76,0.46){1.39}
\psplot{-2.92}{4.92}{(-1.45--1.12*x)/0.82}
\psline(2.06,2.56)(2.9,1.54)
\psline(2.9,1.54)(3.68,2.76)
\psline(3.68,2.76)(2.06,2.56)
\psplot[plotpoints=200]{-2.92}{4.92}{x^2-x}
\psdots[dotstyle=*](0.76,0.46)
\rput[b1](0.84,0.58){A$}
\psdots[dotstyle=*](-0.36,1.28)
\rput[b1](-0.64,1.48){B$}
\psdots[dotstyle=*)(2.06,2.56)
\rput[b1](1.7,2.72){C$}
\psdots[dotstyle=*)(2.9,1.54)
\rput[b1](2.82,1.12){D$}
\psdots[dotstyle=*)(3.68,2.76)
\rput[b1](3.82,2.9){E$}
\end{pspicture*}
\end{document}
```

L'immagine corrispondente è quella della figura 2. Si tratta, come si può ben vedere, di un risultato già accettabile⁽³⁾.

²Cancellando il comando relativo al colore nelle etichette dei punti, per esempio `\rput[b1](0.84,0.58){\blue{A$}}`, rimane una doppia coppia di parentesi graffe, che non dà alcun fastidio: `\rput[b1](0.84,0.58){A$}`; i puristi possono cancellare una delle due coppie di parentesi di graffe, ottenendo `\rput[b1](0.84,0.58){A$}`, che comunque produce lo stesso identico risultato.

³Nel compilare il file precedente, se si usa il formato DVI la resa non è eccellente, in particolare i visualizzatori DVI normalmente usati (per esempio Yap) non riescono a rendere bene l'effetto trasparenza (`opacity`): tutto si risolverà nel file finale in pdf.

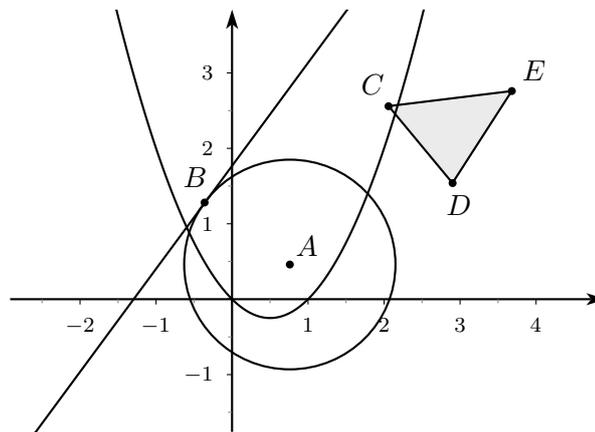


Figura 2 La finestra grafica riprodotta in PSTricks: prima versione

Le cose che possono essere migliorate, e che conviene migliorare vista la relativa facilità del processo, sono:

- il posizionamento delle etichette o l'aggiunta di nuove etichette;
- lo spessore e lo stile delle linee e dei punti;
- il perfezionamento delle etichette messe sugli assi cartesiani.

2.1 Posizionamento delle etichette

Per posizionare le etichette nella figura si possono usare i due comandi fondamentali: `\rput` e `\uput`. In particolare il primo è uno dei comandi di base di PSTricks ed è molto versatile, consentendo anche raffinamenti che in questa sede non intendo affrontare.

La sintassi del comando `\rput` è (semplificando al massimo!):

`\rput[opzioni](x,y){oggetto}`

oppure

`\rput*[opzioni](x,y){oggetto}`

La differenza tra la prima (senza `*`) e la seconda (con `*`) versione è che la seconda consente di sovrascrivere, nascondendo tutto quello che è stato disegnato precedentemente. Il posizionamento dell'oggetto avviene in corrispondenza del punto di coordinate (x, y) tenendo conto dell'opzione scelta, che può essere (anche qui semplificando!):

- `tl` (cioé top left)
- `t` (cioé top)
- `tr` (cioé top right)
- `r` (cioé right)
- `br` (cioé bottom right)
- `b` (cioé bottom)
- `bl` (cioé bottom left)
- `l` (cioé left)
- `c` (cioé center: quest'ultima è il default, si può anche omettere)

Il significato è il seguente: l'oggetto da inserire occupa un certo spazio (una *box* nel gergo di L^AT_EX), di forma in genere rettangolare; questo oggetto viene posizionato in modo che il punto

(detto punto di riferimento) (x, y) si trovi *in altro a sinistra* (`tl`), *in alto al centro* (`t`), ecc. rispetto al rettangolo occupato dall'oggetto.

La sintassi del comando `\uput` è la seguente

`\uput[angolo](x,y){oggetto}`

oppure

`\uput*[angolo](x,y){oggetto}`

Anche qui la versione con `*` sovrascrive eventuali oggetti preesistenti.

A differenza del comando precedente, in questo caso l'etichetta viene posizionata ad una distanza predefinita (ma che potrebbe essere cambiata!) dal punto di riferimento, e in una posizione, individuata dall'angolo indicato, rispetto al punto stesso. Secondo la mia opinione, per le figure geometriche, questo comando è la scelta di gran lunga migliore.

Geogebra sceglie sempre il comando `\rput[b1]`, usando come riferimento un punto quasi sempre diverso da quello a cui l'etichetta si riferisce⁽⁴⁾. Se la scelta di Geogebra soddisfa, si può lasciare tutto così, altrimenti si possono fare delle modifiche "manuali". Ragioniamo su un esempio, relativo al punto B della figura che stiamo costruendo. Il codice proposto da Geogebra è il seguente

```
\psdots[dotstyle=*,linecolor=blue](-0.36,1.28)
\rput[b1](-0.64,1.48){\blue{B}}
```

che abbiamo già semplificato (eliminando il colore) in

```
\psdots[dotstyle=*(-0.36,1.28)
\rput[b1](-0.64,1.48){B}
```

Se vogliamo migliorare la posizione e la visibilità dell'etichetta del punto B possiamo correggere il codice come segue:

```
\psdots[dotstyle=*(-0.36,1.28)
\uput[135](-0.36,1.28){B}
```

ottenendo il seguente effetto:

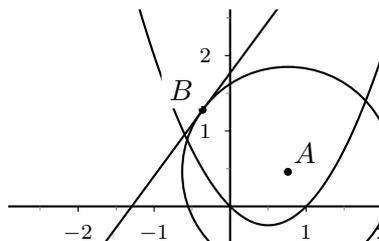


Figura 3 Particolare della finestra grafica riprodotta in *PSTricks*: seconda versione

Da notare che abbiamo sostituito il punto di riferimento generato automaticamente da Geogebra, con le coordinate del punto a cui l'etichetta si riferisce: questo rende più facile determinare l'esatta posizione dell'etichetta rispetto al punto. In maniera analoga possiamo intervenire su altre etichette.

⁴Si osservi che nel file prodotto da Geogebra, ogni punto, inserito con il comando `\psdots`, è immediatamente seguito dalla sua etichetta.

2.2 Spessore e stile delle linee e dei punti

Lo spessore e lo stile delle linee può essere anche deciso in Geogebra. L'effetto a stampa (visibile in particolare nel formato pdf), può essere diverso da quello che appare nella finestra di Geogebra. Per questo motivo consiglio di intervenire manualmente, direttamente sul codice: si possono effettuare diverse modifiche e miglioramenti, ne segnalo solo le più importanti, ragionando sull'esempio che stiamo considerando.

- Innanzitutto nella riga che inizia con `\psset` ritengo opportuno sostituire lo spessore di default delle linee portandolo a 0.5 pt, a differenza del 0.8 che Geogebra propone (di solito) di default.
- Successivamente alcune linee andranno ingrossate per evidenziarle nella figura. La cosa si può fare con l'opzione `linewidth=1pt` (o qualche altra misura di proprio gradimento), da inserire tra le opzioni dei comandi come `\psplot`, `\psline`, `\pscicle`, `\psdots`⁵.

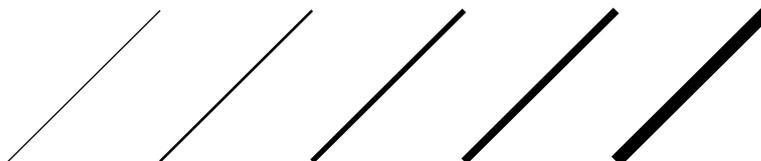


Figura 4 Linee con spessore di 0.5, 1, 2, 3, 5 punti

- Per quanto riguarda lo stile delle linee si possono fare numerose scelte, e anche qui ne segnalo solo una, che consente di avere linee tratteggiate con diversi tipi di tratteggio. L'istruzione relativa, sempre da inserire tra le opzioni, è

`linestyle=dashed,dash=3pt 3pt`

il primo numero dopo `dash=` indica la dimensione del tratto pieno, il secondo quella dello spazio vuoto.



Figura 5 Tratteggi di 2pt 2pt, 3pt 3pt, 4pt 4pt, 5pt 5pt, 5pt 10pt

- Per le dimensioni dei punti una delle opzioni possibili, nella sua versione più semplice, è `dotsize=2pt`, o un'altra misura a scelta.



Figura 6 Punti in dimensioni variabili da 2pt a 10pt

- Anche per lo stile dei punti le opzioni sono innumerevoli. Quelle di uso più comune sono
 - `dotstyle=*`: lo stile normale *pallino pieno*;

⁵Ricordiamo che le opzioni vanno inserite, subito dopo il comando scelto, tra parentesi quadre, e separate da una virgola se ce n'è più d'una. Per esempio nel file prodotto da Geogebra, nella linea `\psdots[dotstyle=*,linecolor=blue]`, la parte `[dotstyle=*,linecolor=blue]` è una successione di 2 opzioni del comando `\psdots`.

- `dotstyle=o` (o è la lettera o minuscola): lo stile *pallino vuoto*;
- `dotstyle=+`;
- `dotstyle=x`;
- `dotstyle=|` (| è la barretta del valore assoluto): lo stile *linea verticale*.

La linea verticale può essere ruotata di un angolo a piacere, di solito 90° per renderla orizzontale, con l'opzione `dotangle=90`.



Figura 7 Alcuni stili di punti

- Se si vogliono solo le *tacche* sugli assi senza i numeri, basta sostituire tutte le opzioni del comando `\psaxes` con `[labels=none]`; se non si vogliono nemmeno le tacche basta sostituirle con `[labels=none,ticks=none]`.
- Naturalmente se si vogliono i nomi degli assi x e y , basterà aggiungerli, alla fine degli stessi e nella posizione voluta, con un comando `\uput`. Segnalo, per concludere, che in certi casi può essere utile inserire una griglia (da togliere quando il disegno è finito!): il comando è `\psgrid`, da mettere subito sotto alla riga che inizia con `\begin{pspicture}`.

Un possibile risultato finale che tiene conto dei suggerimenti dati è proposto nella figura 8.

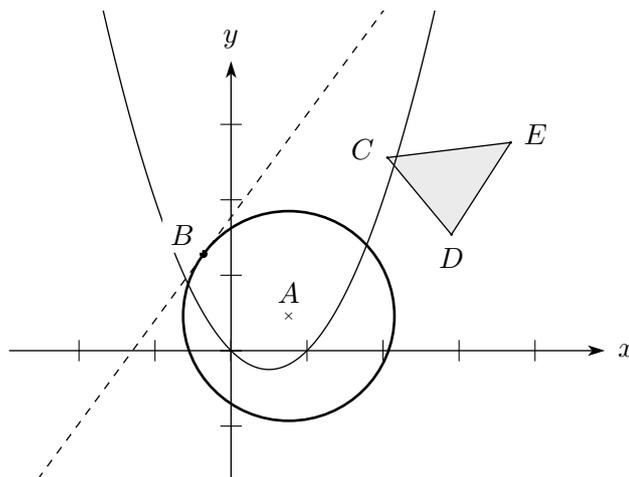


Figura 8 Una possibile figura finale

Il codice che ha prodotto la figura è il seguente.

```
\documentclass[10pt]{article}
\usepackage[latin1]{inputenc}
\usepackage[italian]{babel}
\usepackage{latexsym}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{pstricks-add}
\pagestyle{empty}
\begin{document}
\psset{xunit=1.0cm,yunit=1.0cm,algebraic=true,dotstyle=o,dotsize=3pt 0,
```

```

linewidth=0.5pt,arrowsize=3pt 2,arrowinset=0.25}
\begin{pspicture*}(-2.92,-1.76)(5.5,4.5)
\psaxes[labels=none]{->}(0,0)(-2.92,-1.76)(4.92,3.84)
\pspolygon[linestyle=none,fillstyle=solid,fillcolor=lightgray,opacity=0.3]
(2.06,2.56)(2.9,1.54)(3.68,2.76)
\pscircle[linewidth=1pt](0.76,0.46){1.39}
\psplot[linestyle=dashed,dash=3pt 3pt]{-2.92}{4.92}{(--1.45--1.12*x)/0.82}
\psline(2.06,2.56)(2.9,1.54)
\psline(2.9,1.54)(3.68,2.76)
\psline(3.68,2.76)(2.06,2.56)
\psplot[plotpoints=200]{-2.92}{4.92}{x^2-x}
\psdots[dotstyle=x](0.76,0.46)
\uput[90](0.76,0.46){$A$}
\psdots[dotstyle=*(-0.36,1.28)
\uput*[135](-0.36,1.28){$B$}
\psdots[dotstyle=*,dotsize=1pt](2.06,2.56)
\uput[160](2.06,2.56){$C$}
\psdots[dotstyle=*,dotsize=1pt](2.9,1.54)
\uput[-90](2.9,1.54){$D$}
\psdots[dotstyle=*,dotsize=1pt](3.68,2.76)
\uput[20](3.68,2.76){$E$}
\uput[0](4.92,0){$x$}
\uput[90](0,3.84){$y$}
\end{pspicture*}
\end{document}

```

Si noti che ho sostituito la linea

```
\begin{pspicture*}(-2.92,-1.76)(4.92,3.84)
```

con la

```
\begin{pspicture*}(-2.92,-1.76)(5.5,4.5)
```

e questo al fine di avere spazio a destra e in alto per inserire le etichette x e y sugli assi.

3 L'inserimento delle figure nel documento \LaTeX

Una volta che siamo soddisfatti della nostra figura dobbiamo inserirla nel documento \LaTeX che stiamo componendo. Ci sono diverse strategie per farlo, tra cui anche una che prevede l'inserimento diretto del codice della figura nel file tex . Fin quando non si diventa veramente esperti è comunque meglio seguire una procedura che prevede la costruzione della figura in un file apposito (come quello che produce Geogebra, con gli adattamenti che ho descritto) per poi inserirlo successivamente come immagine nel file tex .

La procedura che consiglio è descritta di seguito.

1. Compilare la figura in PS e visualizzarla, con **GSview** (Ghostscript view).
2. Tramite il menu *File* di **Ghostscript** trasformare l'immagine in formato EPS, rispondendo *Yes* alla richiesta "Have you read help PS to EPS" e controllando che ci sia la spunta su "Automatically calculate Bounding Box".
3. Salvare il file con un nome opportuno (sarà il nome che poi inseriremo nel codice text), *aggiungendo* l'estensione **.eps** che non viene messa da **Ghostscript**.

4. Aprire il programmino `esp2pdf` con cui si trasforma il file `eps` in file `pdf`, mantenendo (in maniera automatica) lo stesso nome (ovviamente non con la stessa estensione). Si tratta di un programmino gratuito (si trova facilmente con un motore di ricerca), molto efficiente e rapido e che produce, a mio avviso, risultati ottimi. Naturalmente se uno ha Adobe Acrobat nella versione `writer` può usare anche quello.
5. A questo punto si hanno due versioni dello stesso disegno: una in formato `eps` e una in formato `pdf`: è proprio quello che serve.

Il preambolo del nostro documento L^AT_EX deve caricare almeno i seguenti pacchetti.

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[italian]{babel}
\usepackage{latexsym}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage{float}
```

Nel punto dove vogliamo inserire la nostra figura mettiamo il seguente codice

```
\begin{figure}[htbp]
\begin{center}
\includegraphics{percorso/nome figura}
\caption{Didascalia}
\end{center}
\end{figure}
```

oppure

```
\begin{figure}[H]
\begin{center}
\includegraphics{percorso/nome figura}
\caption{Didascalia}
\end{center}
\end{figure}
```

Il percorso serve a indicare al compilatore eventuali sottocartelle dove abbiamo piazzato le figure, rispetto a quella dove si trova il file `tex`. Il `nomefigura` *non deve comprendere* l'estensione (`eps` o `pdf`) del file. La *Didascalia* è la didascalia che vogliamo piazzare sotto alla figura.

La differenza tra le due versioni sta solo nell'opzione dopo `\begin{figure}`, che è `[htbp]` nel primo caso, e `[H]` nel secondo caso. Si tratta di richieste fatte al compilatore, in due modi diversi.

- Nel primo caso la richiesta è formulata, a parole, nel seguente modo: *Vedi se puoi mettere la figura qui, se non puoi qui, mettila in cima alla pagina, altrimenti in fondo alla pagina o, infine, in una pagina apposita contenente solo figure. In ogni caso lascio a te compilatore la decisione finale dove mettere la figura.*
- Nel secondo caso la richiesta è formulata, a parole, nel seguente modo: *Metti la figura qui e non discutere.*

Quasi tutti preferirebbero la seconda opportunità. C'è però da tener presente che così potrebbe succedere che in quel punto della pagina la figura non ci può stare perché troppo grande, e allora viene inserita nella pagina successiva, lasciando uno spazio vuoto, più o meno grande, in fondo

alla pagina. Se si usa questa seconda opzione, in genere bisogna agire con aggiustamenti manuali sull'impaginazione⁶.

Usando la prima opzione bisogna tenere conto del fatto che non è prevedibile dove la figura vada a finire (figura *flottante*), e quindi per richiamarla nel testo bisogna usare il meccanismo `\label{etichetta}`, `\ref{etichetta}`. Il codice diventa, più precisamente,

```
\begin{figure}[htbp]
\begin{center}
\includegraphics{percorso/nome figura}
\caption{Didascalia}\label{etichetta}
\end{center}
\end{figure}
```

dove `etichetta` è un nome da noi scelto (esempio: `figura1`) e che deve essere unico in tutto il documento. Quanto dobbiamo richiamare la figura dovremo scrivere una cosa del genere

...con riferimento alla figura `\ref{etichetta}`...

Per un esempio concreto, in questo documento la figura 2 ha, come etichetta,

```
\label{fig:prima_versione}
```

A pagina 4 il richiamo alla figura è fatto con il seguente testo:

L'immagine corrispondente è quella della figura `\ref{fig:prima_versione}`.

Inserita la figura si compila il documento come al solito: mentre si lavora è meglio la compilazione in DVI (e a questo serve l'immagine in formato `eps`), quando tutto è pronto è meglio la compilazione con PDF^LA^TE^X(e a questo serve l'immagine in formato `pdf`).

4 Conclusione

Qualcuno potrà obiettare che il processo è un po' lungo e laborioso. Assicuro che dopo due o tre figure costruite, tutto diventa quasi automatico e, se si costruiscono dei template, è facile ripetere il processo con dei semplici *Copia e incolla*.

Inoltre, e credo sia la cosa in assoluto più importante, la qualità del risultato ottenuto è decisamente superlativa.

Buon Lavoro!

⁶Chi ha provato a inserire immagini in un programma tipo Word o Writer, avrà già sperimentato le difficoltà connesse!!