

PSTricks

ovvero

... la ciliegina sulla torta ...

Luciano Battaia

Una miniguia di avvio per principianti assoluti

Versione del 16 giugno 2008— Work-in-progress!

Sommario

In questa miniguia sono raccolti alcuni materiali e indicazioni forniti durante i corsi di introduzione a PSTricks, tenuti, come attività curricolare, agli studenti del Liceo Scientifico Grigoletti di Pordenone. L'impostazione è dunque quella di un manuale scolastico senza, in particolare, la pretesa di costruire una guida sistematica all'uso di PSTricks, ma solo con l'idea di fornire a chi è interessato un punto di partenza per entrare nell'affascinante mondo di PSTricks. Tutto quello che serve per proseguire si trova nelle numerose guide specializzate reperibili gratuitamente in rete. Esistono poi in commercio anche molti manuali che trattano estesamente tutte le funzionalità di questo package.

Rimando alle note inserite nella mia miniguia di introduzione a \LaTeX per una specificazione delle motivazioni didattiche che mi hanno condotto ad inserire questo software nei curricula scolastici, in particolar modo nei corsi ad indirizzo scientifico.

Indice

| | | |
|-----|--------------------|---|
| 1 | Che cos'è? | 1 |
| 2 | Come si usa | 1 |
| 3 | Installazione | 2 |
| 4 | Principi di base | 2 |
| 4.1 | Punti | 3 |
| 4.2 | Linee e poligoni | 3 |
| 4.3 | Cerchi e archi | 4 |
| 5 | Grafici | 5 |
| 6 | pst-pdf | 6 |
| 7 | Per saperne di più | 8 |

1 Che cos'è?

\LaTeX ha limitate capacità grafiche (e quelle previste non sono semplicissime da usare). Il linguaggio Postscript è invece un linguaggio di descrizione di una pagina da stampare che ha un vasto e completo insieme di comandi grafici. Per un utente \LaTeX è normale produrre output Postscript (per esempio mediante `dvips`, fornito di serie con tutte le implementazioni). Sorge allora la questione dell'inserimento di codice Postscript in un file `tex` in modo da utilizzare tutte le capacità grafiche che il linguaggio possiede.

Questa idea è alla base del pacchetto `PSTricks` di Timothy van Zandt. La cosa essenziale è che non c'è alcuna necessità di conoscere il linguaggio Postscript stesso (se non per alcune fasi nella definizione di certe funzioni matematiche): `PSTricks` è una raccolta di macro comandi, simili a quelli previsti in tutti i pacchetti \LaTeX , che rendono estremamente semplice la produzione di grafici anche molto sofisticati.

Tutte le notizie relative a questo pacchetto e alle sue varie estensioni sono reperibili nel sito web ad esso dedicato <http://www.pstricks.de>. In questo stesso sito potete trovare alcuni spettacolari esempi di immagini prodotte con `PSTricks`.

2 Come si usa

Il problema più grosso nell'uso di questo pacchetto è il fatto che non si può produrre direttamente un file `pdf` con `PDF \LaTeX` , e non sempre il passaggio `dvi`→`ps`→`pdf` è conveniente (il passaggio diretto `dvi`→`pdf` non funziona). Il pacchetto `pst-pdf` risolve quasi completamente il problema, ma il suo uso non è semplicissimo, in particolare in combinazione con `MiKTeX`. Nella sezione 6 gli interessati possono trovare le indicazioni relative al metodo che seguo.

Per la maggior parte dei casi, però, si può usare questo pacchetto semplicemente per produrre grafici da inserire poi in un normale file \LaTeX con il solito `\includegraphics`. In questo caso non ci sono problemi particolari e si può seguire la strategia indicata qui di seguito.

1. Produrre un file minimale contenente solo la figura Postscript. Potete usare il seguente template:

```
\documentclass{minimal}
\usepackage{...} %eventuali package richiesti%
\usepackage{pstricks}
\usepackage{pst-...} %eventuali add-ons di PSTricks%
\pagestyle{empty} %per non far comparire il numero di pagina%
\begin{document}
... codice PSTricks ...
\end{document}
```

2. Compilare con \LaTeX , convertire in `ps` (con `dvips`), aprire Ghostscript e, mediante esso, convertire in `eps` (menu File, PS to EPS).
3. Convertire successivamente il file `eps` ottenuto in `pdf`, preferibilmente mediante l'apposito programma `eps2pdf.exe`, liberamente scaricabile dalla rete. In questo modo si ha a disposizione sia un'immagine `eps` che un'immagine `pdf` e si potrà, nel file \LaTeX principale, compilare sia in \LaTeX che in $\PDF\LaTeX$.

Il processo può apparire complicato e noioso, ma ne vale proprio la pena: i grafici che si ottengono sono di qualità eccezionale (e non si deve pagare alcunchè per costruirli ...).

3 Installazione

L'installazione del package non comporta alcun problema e si fa in modo standard. In MiKTeX se si è scelta la versione "Total" tutto viene installato di default e non c'è bisogno di fare nulla. Consiglio comunque di seguire costantemente il sito www.pstricks.de per controllare eventuali aggiornamenti (molto frequenti, in quanto il package è in costante evoluzione). Comunque se non avete esigenze particolari non occorre preoccuparsi troppo: tutto funziona anche con l'installazione di base.

Esistono anche numerose estensioni del pacchetto e non tutte vengono caricate nelle installazioni standard. Se avete particolari esigenze consultate sempre il già citato sito relativo a PSTricks per ogni utile informazione.

4 Principi di base

Come già accennato, questa non vuole essere una guida di PSTricks, ma solamente un piccolo aiuto per iniziare. Tratterò solamente alcuni comandi veramente di base e la maggiore parte delle spiegazioni saranno deducibili dagli esempi proposti.

PSTricks usa il metodo della coordinate cartesiane per tracciare le figure. Ogni figura deve venire inserita in un box, specificando le coordinate cartesiane del punto in basso a sinistra e in alto a destra del box stesso: (x_s, y_s) e (x_d, y_d) . Il box viene piazzato esattamente dove si trova il punto di inserimento nel file corrente. Tutto il codice va racchiuso inoltre tra `\begin{pspicture}` e `\end{pspicture}`:

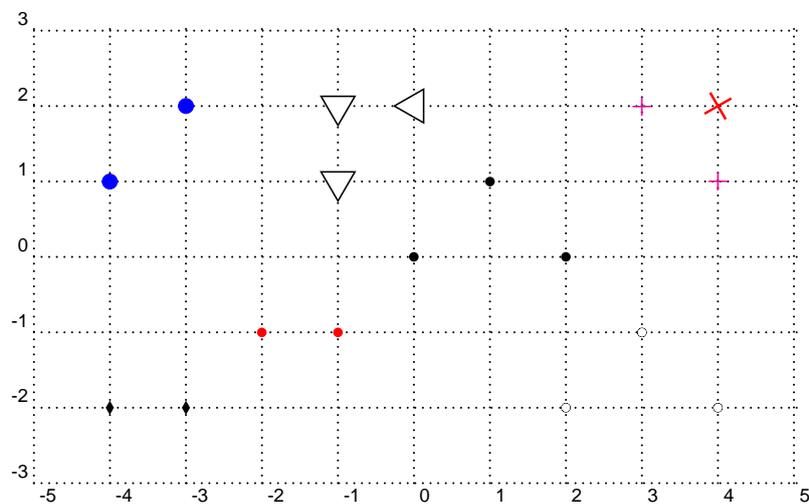
```
\begin{pspicture}(x_s,y_s)(x_d,y_d)
... codice ...
\end{pspicture}
```

La dimensione usata di default è 1 cm, per cui il punto $(1, 0)$ si troverà sull'asse delle x del sistema di assi di default (che non viene tracciato!) a 1 cm dall'origine. Naturalmente tutte le unità possono essere cambiate a piacere.

4.1 Punti

I punti si tracciano con il comando `\psdots`, come nell'esempio seguente (la griglia che vedete sia in questo che negli esempi seguenti è solo per confronto, e non si ottiene con il codice riportato).

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\psdots(0,0)(2,0)(1,1)
\psdots[dotstyle=diamond*](-4,-2)(-3,-2)
\psdots[dotstyle=o](2,-2)(3,-1)(4,-2)
\psdots[linecolor=red](-2,-1)(-1,-1)
\psdots[dotsize=6pt,linecolor=blue](-4,1)(-3,2)
\psdots[dotsize=6pt,linecolor=magenta,dotstyle=+] (3,2)(4,1)
\psdots[dotsize=10pt,linecolor=red,dotstyle=+,dotangle=30](4,2)
\psdots[dotstyle=triangle,dotsize=12pt,dotangle=60](-1,1)(-1,2)
\psdots[dotstyle=triangle,dotsize=12pt,dotangle=90](0,2)
\end{pspicture}
```



Per evitare che vi scervelliate a capire come è fatta la griglia, ecco qui il comando:

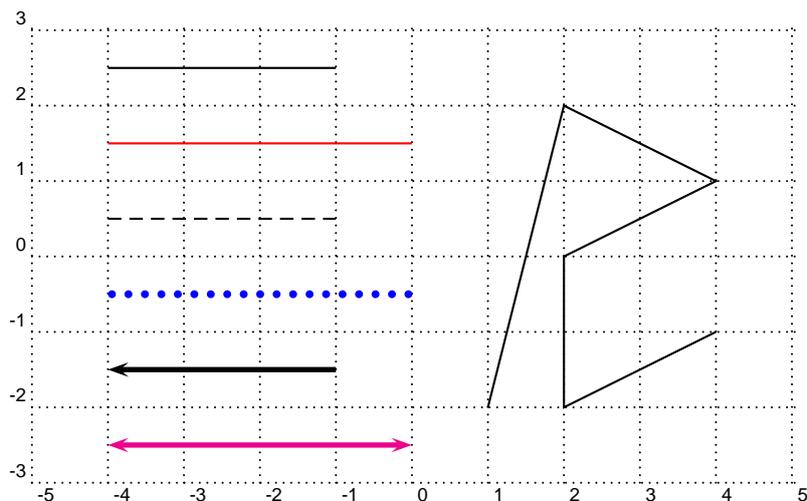
```
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt](-5,-3)(5,3)
```

Osservate che il box indicato per la figura ha uno spazio vuoto di 0.5 cm su tutti i lati, rispetto alla griglia, per consentire di distanziare la figura dal testo.

4.2 Linee e poligoni

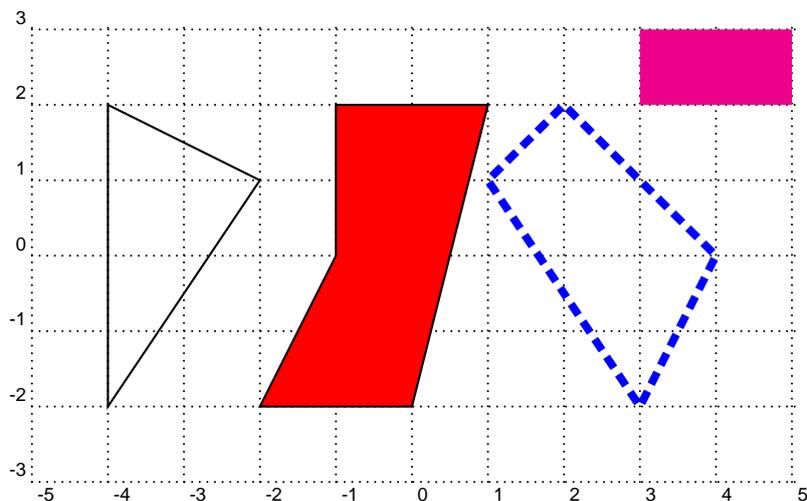
Le linee si tracciano con il comando `\psline(x1,y1)(x2,y2)(x3,y3)...`. Si veda l'esempio.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\psline(-4,2.5)(-1,2.5)
\psline[linecolor=red](-4,1.5)(0,1.5)
\psline[linestyle=dashed](-4,0.5)(-1,0.5)
\psline[linestyle=dotted,linewidth=3pt,linecolor=blue](-4,-0.5)(0,-0.5)
\psline[linewidth=2pt]{<-}(-4,-1.5)(-1,-1.5)
\psline[linecolor=magenta,linewidth=2pt]{<->}(-4,-2.5)(0,-2.5)
\psline(1,-2)(2,2)(4,1)(2,0)(2,-2)(4,-1)
\end{pspicture}
```



Il comando `\pspolygon` traccia una poligonale chiusa, cioè automaticamente l'ultimo punto è connesso con il primo. La variante `\pspolygon*` riempie il poligono con il colore del bordo. Si può comunque riempire con un colore a piacere usando `fillcolor` e `fillstyle`. Si veda l'esempio.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\pspolygon(-4,2)(-4,-2)(-2,1)
\pspolygon[fillcolor=red,fillstyle=solid](-2,-2)(-1,0)(-1,2)(1,2)(0,-2)
\pspolygon[linecolor=blue,linewidth=3pt,linestyle=dashed](1,1)(2,2)(4,0)(3,-2)
\end{pspicture}
```

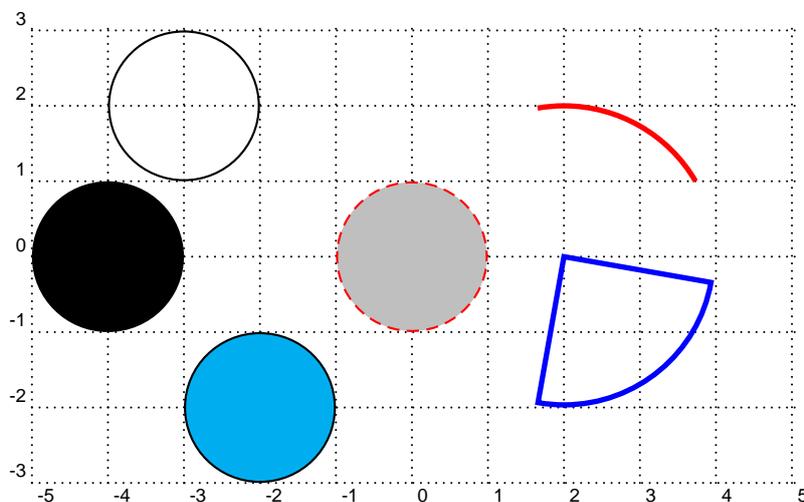


4.3 Cerchi e archi

I cerchi si tracciano con il comando `\pscircle(centro){raggio}` (`\pscircle*(centro){raggio}` fa un cerchio pieno). Per gli archi è previsto il comando `\psarc`, dove si devono specificare il centro, il raggio, l'angolo iniziale e quello finale (in gradi). Per i settori circolari il comando è `\pswedge`. Si veda l'esempio.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\pscircle(-3,2){1}
\pscircle*(-4,0){1}
\pscircle[fillcolor=cyan,fillstyle=solid](-2,-2){1}
\pscircle[fillcolor=lightgray,fillstyle=solid,linestyle=dashed,linecolor=red](0,0){1}
```

```
\psarc[linecolor=red,linewidth=2pt](2,0){2}{30}{100}
\pswedge[linecolor=blue,linewidth=2pt](2,0){2}{260}{350}
\end{pspicture}
```



5 Grafici

I grafici delle funzioni reali si tracciano con il comando `\psplot`, contenuto nel pacchetto aggiuntivo `pst-plot`, che occorre caricare. Lo stesso pacchetto prevede la possibilità di tracciare in automatico gli assi cartesiani.

L'unico problema è che la definizione delle funzioni deve essere fatta usando la sintassi Postscript. Non si tratta comunque di un problema complicato e, in ogni caso, la trattazione di questo tipo di sintassi può costituire un utile argomento di discussione relativamente al complesso problema delle notazioni in matematica.

Senza entrare troppo nei dettagli, mostriamo alcuni esempi di formule PostScript. Occorre prestare la massima attenzione all'ordine e agli spazi, che hanno la funzione di separatori. Le formule sono scritte tra apici, per rendere chiaro l'inizio e la fine: essi non vanno inseriti nel codice.

- $a + b$ si scrive: '*a b add*'
- $a - b$ si scrive: '*a b sub*'
- $a + b + c$ si scrive: '*a b add c add*'
- $a \cdot b$ si scrive: '*a b mul*'
- $\frac{a}{b}$ si scrive: '*a b div*'
- $(a + b)c$ si scrive: '*a b add c mul*'
- $a(b + c)$ si scrive: '*a b c add mul*'
- $\frac{a+b}{c+d}$ si scrive: '*a b add c d add div*'
- $\frac{a}{b+c}$ si scrive: '*a b c add div*'
- $\arctan\left(\frac{a}{b}\right)$ si scrive: '*a b atan*' e fornisce come risultato un angolo in gradi.
- \sqrt{a} si scrive: '*a sqrt*'
- $\sqrt{\frac{a}{b+c}}$ si scrive: '*a b c add div sqrt*'
- a^b si scrive: '*a b exp*'

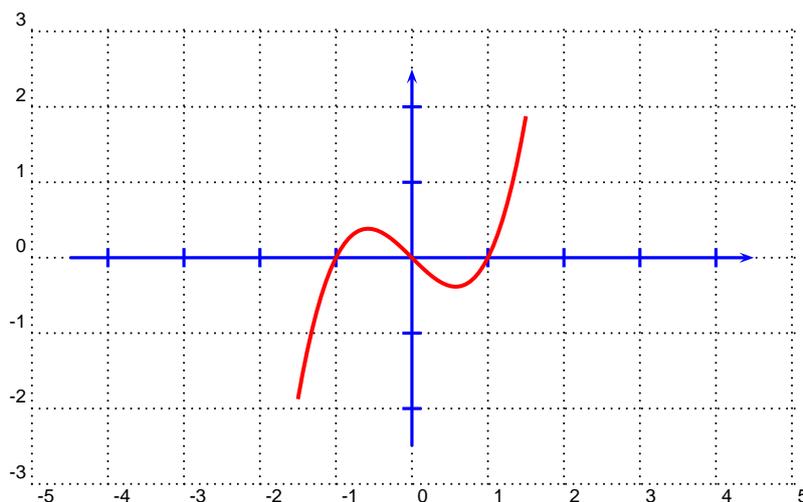
Si noti la mancanza di parentesi e l'uso degli operatori di somma, sottrazione, prodotto, quoziente solo per coppie di valori. Quest'ultima osservazione è fondamentale; è per questo che $a(b+c)$ si scrive 'a b c add mul'. Infatti l'operatore *add* agisce sui due numeri che immediatamente lo precedono (b e c), producendo come un risultato un altro numero, diciamolo d ; a questo punto l'operatore *mul* agisce sui due numeri che lo precedono, ovvero a e d . In sostanza si può evitare l'uso di parentesi proprio perchè ciascun operatore agisce solo come è previsto nella sua definizione originale. La cosa appare evidente quando si considera, per esempio, la somma di tre addendi: $a+b+c$ si scrive 'a b add c add', ad indicare il fatto che si esegue intanto la somma tra a e b e poi il risultato si addiziona con c . Si può anche osservare, come utile esercizio, che la proprietà associativa della somma implica che 'a b add c add' è uguale a 'a b c add add': nel primo caso si sommano prima a e b e il risultato si somma con c , nel secondo caso si sommano prima b e c e il risultato si somma poi con a .

Per esempio la funzione $f(x) = x \sin x$ si scrive 'x x sin mul', cioè (leggendo dal fondo) moltiplica il seno di x per x . La funzione $x^3 - x$ si scrive 'x 3 exp x sub', cioè sottrai x da x elevato al cubo.

Il codice per produrre il grafico di quest'ultima funzione è riportato di seguito.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\psaxes[labels=none,linecolor=blue,linewidth=1.2pt]{->}(0,0)(-4.5,-2.5)(4.5,2.5)
\psplot[linecolor=red,linewidth=1.5pt]{-1.5}{1.5}{x 3 exp x sub}
\end{pspicture}
```

L'intervallo $[-1.5, 1.5]$ è il dominio preso in considerazione per tracciare il grafico.



6 pst-pdf

Come già citato la produzione di un file pdf da un sorgente \LaTeX contenente codice PSTricks non è immediata, in particolare non è possibile utilizzando direttamente \PDF\LaTeX . Il pacchetto *pst-pdf* risolve completamente il problema.

Di seguito potete trovare le indicazioni per far funzionare questo pacchetto sotto MiKTeX. Il funzionamento può essere completamente automatizzato se avete WinEdt, ma non è difficile nemmeno se il vostro editor non è configurabile per l'esecuzione automatica di batch file (in questi casi però occorre usare il prompt di DOS).

Le istruzioni che seguono sono lunghe e dettagliate, ma non scoraggiatevi, tutto il processo si può completare in mezz'ora circa (e vi assicuro che ne vale proprio la pena!).

Fase preliminare.

1. Se avete MiKTeX 2.5 il pacchetto dovrebbe essere già installato. In ogni caso potete reperirlo in un CTAN qualunque oppure in <http://perce.de/LaTeX/pst-pdf/>, e allora dovete seguire la normale procedura che si fa per installare una nuovo pacchetto, ricordandovi sempre di fare un refresh del database di MiKTeX.

2. Scaricate il seguente file batch dal mio sito:

<http://www.batmath.it/latex/down/ps4pdf.bat>

Si tratta di un batch file per DOS, di Lutz Ihlenburg (versione del 9 maggio 2005), modificato da Patrice M egret (versione del 31 maggio 2005), e adatto ad essere usato esclusivamente per il pacchetto `pst-pdf` di Rolf Niepraschk sotto MiKTeX. Mantenete sempre una copia di sicurezza di questo file per evitare di doverlo scaricare nuovamente. Con MiKTeX 2.5 viene distribuito il file originale di Lutz Ihlenburg che, almeno a me, non funziona correttamente.

Ora avete tutto quello che serve per compilare un file L^AT_EX che contiene codice PSTricks. Se non avete WinEdt potete procedere nel modo di seguito indicato.

1. Mantenete il vostro file L^AT_EX che supponiamo abbia nome `miofile.tex`, in una cartella di C: (esempio C:\miacartella). Nella stessa cartella metteteci una copia del file `ps4pdf.bat`.
2. Lavorate normalmente con il file L^AT_EX fin quando non siete giunti alla versione finale, facendo attenzione a compilare con L^AT_EX, convertire in `ps` (con `dvips`) e a visualizzare con `Gosthscript`.
3. Inserite, solo ora, nel preambolo del documento L^AT_EX l'istruzione `\usepackage{pst-pdf}`, preferibilmente dopo quella che richiama i packages di PSTricks.
4. Quanto tutto   pronto aprite il prompt di DOS, andate nella cartella C:\miacartella) e digitate `ps4pdf miofile.tex`. Questa precompilazione dovrebbe produrre un file dal nome `miofile-pics.pdf` che contiene tutte le figure PSTricks del documento (oltre ad alcuni altri file di servizio).
5. A questo punto una compilazione con PDFL^AT_EX dovrebbe regolarmente funzionare e produrre il risultato finale voluto.
6. Un'ultima osservazione. Se qualcosa dovesse andare storto (ma a me non   mai successo) cancellate tutti i file secondari che vengono prodotti dalle varie compilazioni prima di ripetere le operazioni.

Se invece avete WinEdt e volete rendere automatico il processo sopra riportato, dovete configurare l'editor, seguendo i passi di seguito indicati.

1. Piazzate il file `ps4pdf.bat` nella sottocartella `Bin` della cartella `WinEdt`.
2. Vi consiglio di inserire una nuova voce di menu nel gruppo `pdf` di `Accessories`. Per fare questo aprite WinEdt e, tramite il menu `Options`, lanciate il `MenuSetup`. Nella scheda `MainMenu`, fate doppio click sulla voce `&Accessories`
3. Nella finestra che si aprir  posizionate il puntatore in fondo al gruppo denominato `&pdf` (selezionando la linea di separazione marcata con `<=====`) e inserite una `Separator Bar`, usando il primo pulsante nella barra degli strumenti di questa finestra, alla voce `Insert`. Posizionatevi di nuovo in fondo allo stesso gruppo, e, mediante lo stesso pulsante, inserite una `Macro`.

(a) Nella casella `Menu Items` digitate un nome per la macro (per esempio `&ps4pdf`; attenzione l'& iniziale   importante!).

(b) Nella casella `Macro` digitate esattamente il seguente codice:

```
Run('"%B\Bin\ps4pdf.bat" "%N%T"', '%P');
```

(c) Nella casella `Requires File Filter` digitate esattamente il seguente codice:

```
"!M=TeX"
```

(d) Nella casella `Start in` digitate esattamente il seguente codice:

```
%P
```

(e) Chiudendo le finestre aperte dovrete trovare una nuova voce nel menu `Accessories`, dal nome `ps4pdf`.

4. Per produrre un file pdf con un codice contenente istruzioni PSTricks potete ora procedere come segue
 - (a) Lavorate normalmente con il file \LaTeX fin quando non siete giunti alla versione finale, facendo attenzione a compilare con \LaTeX , convertire in ps (con `dvips`) e a visualizzare con Ghostscript.
 - (b) Inserite, solo ora, nel preambolo del documento \LaTeX l'istruzione `\usepackage{pst-pdf}`, preferibilmente dopo quella che richiama i packages di PSTricks.
 - (c) Precompilate usando la voce di menu appena creata. Questa precompilazione dovrebbe produrre un file dal nome `miofile-pics.pdf` che contiene tutte le figure PSTricks del documento (oltre ad alcuni altri file di servizio).
 - (d) A questo punto una compilazione con $\PDF\LaTeX$ dovrebbe regolarmente funzionare e produrre il risultato finale voluto.
 - (e) Un'ultima osservazione. Se qualcosa dovesse andare storto (ma a me non è mai successo) cancellate tutti i file secondari che vengono prodotti dalle varie compilazioni prima di ripetere le operazioni.

Se volete essere proprio pignoli potete creare anche un'immagine per la voce di menu appena introdotta, e inserire un pulsante sulla barra degli strumenti di WinEdt. Dovrete costruirvi a mano un'icona, in formato bitmap, ma non ci si mette più di qualche minuto. Potete anche editarne una di quelle già presenti nella cartella `Bitmaps\Images` di WinEdt, assegnandole il nome, per esempio, `ps4pdf.bmp`. Una volta creata (mantenete le dimensioni di 21×20) piazzatene una copia nella cartella `Bitmaps\Images` e una nella cartella `Bitmaps\Buttons`. Poi eseguite le operazioni indicate di seguito.

1. Nella cartella base di WinEdt rintracciate il file `WinEdt.btn`, apritelo con un text editor (Blocco note di Windows o lo stesso WinEdt) e aggiungete in fondo la riga

```
xxx %B\Bitmaps\Buttons\ps4pdf.bmp,
```

dove `xxx` è il successivo dell'ultimo numero dell'elenco.

2. Nella stessa cartella base di WinEdt rintracciate il file `WinEdt.img`, apritelo con un text editor (Blocco note di Windows o lo stesso WinEdt) e aggiungete in fondo la riga

```
xxx %B\Bitmaps\Images\ps4pdf.bmp,
```

dove `xxx` è il successivo dell'ultimo numero dell'elenco.

3. A questo punto riaprite la scheda `MenuSetup`, successivamente `Accessories`, `&ps4pdf` e, nella casella `Images and Hint`, cliccate prima su una e poi sull'altra delle icone vuote che vi compaiono. Utilizzando la finestra che si apre potete facilmente scegliere l'icona che avete appena creato. Chiudete le finestre che si sono aperte.
4. Cliccate nella zona della "ToolBar", col tasto destro e scegliete la voce `Enter ToolBar Setup`: utilizzando la finestra che si aprirà sarà facilissimo piazzare il pulsante che avete creato dove vi fa più comodo.

7 Per saperne di più

Per procedere oltre consiglio di leggere innanzitutto il manuale dell'Indian \TeX Users Group scaricabile all'indirizzo <http://sarovar.org/projects/pstricks/>. Si tratta di un'ottima presentazione che parte praticamente da zero ed è adatta anche ai principianti. Facilmente consultabile a video per il tipo di formato in cui è presentata.

Successivamente dovrete leggere il testo-manuale di Herbert Voss (il guru attuale di PSTricks) di cui potete trovare tutte le notizie sul già citato sito <http://www.pstricks.de>.

Infine potete scaricare e stampare il manuale originale di Timothy Van Zandt che contiene la descrizione di tutti i comandi previsti. Il manuale non è di facilissima consultazione e per questo consiglio prima la lettura della guida citata sopra.

I numerosi package aggiuntivi hanno poi un loro specifico manuale di solito compreso nel package stesso.