Il sistema PSfrag, versione 3

Michael C. Grant David Carlisle
psfrag@rascals.stanford.edu*

11 Aprile 1998

Tradotto da FLavio Casadei Della Chiesa e Lorenzo Masetti

Indice

1 Cosa è PSfrag?

Molti pacchetti di grafica e disegno producono output nel formato Encapsulated PostScript (EPS), ma pochi possono facilmente produrre equazioni e testi scientifici nei quali TEX è così abile. D'altro canto, la maggior parte dei pacchetti per grafica basati su LATEX non è così espressiva e di facile utilizzo come questi strumenti *stand-alone*.

*Questo indirizzo non è più attivo [N.d.T.]

PSfrag fornisce il meglio dei due mondi, permettendo all'utente di arricchire file Encapsulated PostScript (EPS) con costrutti LaTeX arbitrari. Per compiere questo, l'utente piazza un semplice *tag* di testo nel file grafico, come una sorta di marcatore di posizione. Quindi, utilizzando semplici comandi LaTeX, l'utente ordina a PSfrag di rimuovere quel *tag* dalla figura, e di sostituirlo con un'equazione LaTeX opportunamente ridimensionata, allineata e ruotata. PSfrag permette all'utente di inserire costrutti LaTeX direttamente nello stesso file EPS.

Il Dott. Craig Barrat scrisse la versione originale di PSfrag quando era studente presso la Stanford University. Da allora l'interfaccia è cambiata solo leggermente, ma l'implementazione è stata completamente riscritta. La versione attuale di PSfrag viene mantenuta da Michael Grant e David Carlisle. Si ringraziano i membri della mailing list di PSfrag, e chiunque abbia inviato segnalazioni di bug o suggerimenti.

Nota alla traduzione italiana

Una copia di questo documento e altre traduzioni in italiano di manuali LaTeX sono reperibili presso:

- `http://guild.prato.linux.it`
- `ftp://lorien.prato.linux.it/pub/guild`

e su ogni sito CTAN nella directory `/info/italian`. Per un elenco aggiornato dei siti mirror CTAN consultate `http://www.ctan.org/` alla voce "CTAN mirror".

2 Cosa serve a PSfrag

Per utilizzare PSfrag sono necessari i seguenti tool:

- Una versione recente di LaTeX 2_ε e del pacchetto graphics. PSfrag richiede la versione del 01/12/1997 o superiore di entrambi i pacchetti, comunque è sempre meglio avere le versioni più recenti.
- Se si pensa di utilizzare il pacchetto seminar con PSfrag è necessario assicurarsi di avere la versione del 13/10/1997 o superiore (controllare il paragrafo 8.3).

Le ultime versioni di LaTeX 2_ε, del pacchetto graphics, di PSfrag, e di dvips possono essere trovate sul CTAN, il Comprehensive TeX Archive Network. I siti CTAN ed i loro mirror includono:

Nome	Indirizzo IP	Locazione
`ftp.dante.de`	129.206.100.192	Germania
`ftp.tex.ac.uk`	128.232.1.87	Regno Unito
`ftp.cdrom.com`	165.113.58.253	USA

2.1 Scelta di un *driver* PostScript

PSfrag si basa su qualche "trucco" PostScript per raggiungere i suoi scopi. A causa di tempo e risorse limitate, l'autore non può confermare che PSfrag funzioni correttamente con ogni *driver* PostScript disponibile. Abbiamo tentato di assicurare che col tempo PSfrag funzionerà con ogni *driver* pienamente compatibile con il pacchetto graphics (ovvero uno con il quale viene fornito un file `.def`.)

È stato confermato che PSfrag funziona con i seguenti *driver*:

Driver	Testato da	Compatibilità
Thomas Rokicki's dvips	gli autori	pienamente compatibile
Y&Y's DVIPSONE	gli autori	pienamente compatibile

Aiutateci ad aggiungere campi a questa lista! Se PSfrag funziona con il vostro *driver* fatecelo sapere, in questo modo possiamo aggiungerlo alla lista. Se possibile, verificate l'output sia su stampanti Level 1 che Level 2, in modo da poter fare eventuali distinzioni se necessario. Se PSfrag non funziona, per favore inviate una segnalazione del bug; consultate il paragrafo 9 per sapere come contattarci. Sfortunatamente, non possiamo promettere un rimedio per tutti, ma ci piacerebbe assicurare che i *driver* più popolari rimangano compatibili.

3 Installare PSfrag

Installare i vari file del PSfrag è semplicissimo:

1. Eseguire LaTeX su `psfrag.ins` per estrarre `psfrag.sty` e `psfrag.pro`
2. Installare `psfrag.sty` nella locazione usuale per le macro di LaTeX. In distribuzioni basate su kpathsea tipo teTeX, questa locazione è determinata dalla variabile `TEXINPUTS`.
3. Installare `psfrag.pro` dove il *driver* PostScript va a cercare i file di intestazione. Per sistemi basati su kpathsea tipo teTeX, questo è indicato dalla variabile `DVIPSHEADERS`. Per dvips, in particolare, la scelta più sensata è la stessa directory dei file `tex.pro` e `special.pro`.
4. Se si possiede una vecchia versione di PSfrag, bisogna cancellare, se esistono, i seguenti file: `ps2frag.ps`, `ps2frag`, `ps2psfrag` (i vari script di eleborazione), e `epsf.sty` (quello fornito da PSfrag, *non* quello di dvips!). Gli amministratori di sistema dovrebbero sostituire `ps2frag` con uno script che avverta gli utenti dell'aggiornamento.

4 Utilizzo

Qua sotto c'è una veloce descrizione dell'utilizzo di PSfrag:

- Usare il comando `\includegraphics` definito dai pacchetti graphics e graphicx per aggiungere figure EPS. Se si utilizza il comando `\epsfbox` di `epsf.sty`, allora `epsf.sty` deve essere caricato dopo `psfrag.sty`. Altri pacchetti basati su `graphics.sty`, tipo graphicx oppure epsfig, non soffrono di questa restrizione.
- Caricare `psfrag.sty` con un comando `\usepackage`.
- Assicurarsi che le figure EPS contengano un semplice *tag* in ciascuna delle posizioni in cui si vuole mettere un comando LaTeX. Bisogna utilizzare una sola parola composta da lettere non accentate e numeri. Si è cercato di permettere l'utilizzo di un *tag* di testo più arbitrario, ma il meccanismo non è infallibile; si veda il paragrafo 8.1.
- Per ogni *tag* nel file EPS, aggiungere un comando al documento LaTeX per specificare in che modo questo *tag* debba essere sostituito, nel modo seguente:

  `\psfrag{`*tag*`}[`⟨*posn*⟩`][`⟨*psposn*⟩`][`⟨*scala*⟩`][`⟨*rot*⟩`]{`*testo LaTeX* `}`

Il *tag* verrà sostituito con del testo LaTeX. Per esempio: in un programma di disegno tipo xfig, piazzare il testo

```
xy
```

in un punto particolare. Per sostituirlo con $x + y$, si può usare la macro

```
\psfrag{xy}{$x+y$}
```

Tutte le chiamate a comandi `\psfrag` che precedono `\includegraphics` (o comandi equivalenti) nello stesso ambiente o in ambienti circostanti verranno utilizzate solo per un dato file EPS; in questo modo possono essere definiti sia comandi `\psfrag` "globali" che locali ad una certa figura.

Qualsiasi testo che non viene menzionato in un comando `\psfrag` non verrà sostituito, quindi PostScript e LaTeX possono essere liberamente mescolati. Visualizzando l'output con un visualizzatore DVI tipo xdvi o dviwin, una lista verticale delle sostituzioni effettuate verrà piazzata a sinistra di ciascuna figura. Essa permette di controllare la formattazione delle sostituzioni, e comunque scompare nel documento PostScript finale. Sfortunatamente i *driver* DVI non sono capaci di piazzare le sostituzioni effettuate da PSfrag sopra le figure e quindi per vedere il risultato è necessario stampare oppure utilizzare un visualizzatore PostScript tipo GhostView.

Questa versione del PSfrag dovrebbe funzionare in modo adeguato nella modalità di compatibilità LaTeX2.09.

5 Comandi e ambienti

`\psfrag{`*tag*`}[`⟨*posn*⟩`][`⟨*psposn*⟩`][`⟨*scala*⟩`][`⟨*rot*⟩`]{`*sostituzione*`}` `\psfrag*{`*tag*`}[`⟨*posn*⟩`][`⟨*psposn*⟩`][`⟨*scala*⟩`][`⟨*rot*⟩`]{`*sostituzione*`}`

La macro `\psfrag` definisce una "`{`*sostituzione*`}` LaTeX" che deve essere piazzata al posto dei `{`*tag*`}` PostScript. Il comando deve essere inserito prima della chiamata ad `\includegraphics`, o comandi equivalenti, ed ha effetto su tutte le occorrenze del `{`*tag*`}` nella figura.

Un comando `\psfrag` rimarrà attivo fino alla chiusura dell'ambiente in cui è stato invocato. Si possono perciò definire `\psfrag` globali, che vengono applicati ad ogni figura, o, ad esempio, un `\psfrag` all'interno di un ambiente `figure` in modo che venga applicato ad un solo file EPS.

Gli argomenti opzionali di posizionamento `[`⟨*posn*⟩`]` e `[`⟨*pspons*⟩`]` specificano come vengono allineati il *bounding box* del testo LaTeX ed il *bounding box* del testo PostScript, rispettivamente. Alcuni strumenti per disegnare potrebbero chiamare questi oggetti *control point* o *alignment point*.

`[`⟨*posn*⟩`]` il punto di riferimento di LaTeX. La sintassi di questo argomento è identica a quella del comando `\makebox`. Possono essere scelte fino a due lettere, una dalla lista {`t`,`b`,`B`,`c`} (in cima, in fondo, sulla *baseline*, al centro) e l'altra dalla lista {`l`,`r`,`c`} (a sinistra, a destra, al centro). Se una delle due manca, viene utilizzato `c` (centro). Assieme, esse specificano uno dei 12 punti di ancoraggio; se l'argomento viene interamente omesso, allora viene assunto `[Bl]`—ma nota che fornendo `[]` si intende posizionamento centrato.

Quando utilizzato nella modalità di compatibilità LaTeX2.09, l'allineamento predefinito è `[bl]`, questo per supportare i vecchi documenti. Di solito non dovrebbe essere una differenza significativa.

[⟨*psposn*⟩] Il punto di riferimento del testo PostScript. I possibili argomenti sono identici a quelli del [⟨*posn*⟩], lo stesso dicasi per il valore predefinito, `[Bl]` (`[bl]` nella modalità di compatibilità LaTeX2.09).

La sostituzione LaTeX può essere opzionalmente ridimensionata e ruotata rispetto al suo punto di riferimento:

[⟨*scala*⟩] Fattore di scala. È meglio utilizzare il cambiamento della dimensione dei font nel testo LaTeX piuttosto che il fattore di scala, ma si può utilizzare il fattore di scala per aggiustare la loro dimensione. Il valore predefinito è `[1]`.

[⟨*rotn*⟩] Rotazione aggiuntiva attorno al punto di riferimento, in gradi. La rotazione "nominale" del testo LaTeX combacia con quella del testo PostScript che deve sostituire; la rotazione totale è questo valore "nominale" più [⟨*rotn*⟩]. Il valore predefinito è `[0]`.

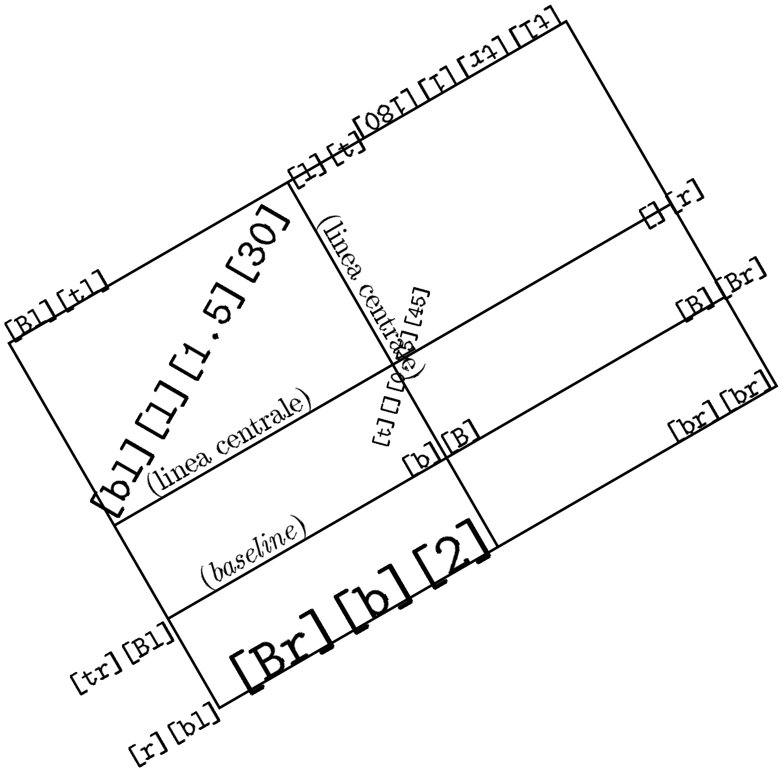

Figura 1: Una dimostrazione delle varie opzioni per il comando `\psfrag`.

La figura 1 illustra varie combinazioni di argomenti. Se si visualizza questo documento con un visualizzatore DVI tipo `xdvi`, allora le sostituzioni PSfrag dovrebbero essere allineate alla sinistra della figura e, se il visualizzatore è capace di mostrare file EPS, si dovrebbe vedere una grande `gA` ruotata. Se si stampa il documento, o se lo si visualizza con un visualizzatore PostScript come GhostView, allora le sostituzioni dovrebbero essere sovrapposte su una rappresentazione grafica del *bounding box*, linee centrali e *baseline* del *tag* `gA`. (La rappresentazione grafica dei *bounding box* è fornita solo in modalità debug.)

Se è già stata definita una sostituzione per `{`*tag*`}`, allora il comando `\psfrag` senza asterisco lo sostituirà senza avvertire. La versione asteriscata, comunque, aggiungerà una nuova sostituzione alla lista. Utilizzando la versione asteriscata, un solo pezzo di un testo PostScript può innescare molte sostituzioni. Non mi viene in mente il motivo per cui molti utenti utilizzano la versione asteriscata del comando, comunque nella figura 1 è stata utilizzata a titolo esemplificativo.

`\begin{psfrags} \end{psfrags}`

Si può utilizzare l'ambiente `psfrag` per delimitare la visibilità delle chiamate a `\psfrag`. Come detto sopra, i comandi `\psfrag` persistono fino all'uscita dal più vicino ambiente circostante. *Qualsiasi* ambiente: `center`, `figure`, etc., quindi non dovrebbe essere necessario utilizzare questo ambiente, che non ha altri effetti sul documento.

5.1 Inglobare operazioni PSfrag nei file EPS

```
\tex[⟨posn⟩][⟨psposn⟩][⟨scala⟩][⟨rot⟩]{LaTeX text}
\psfragscanon  \psfragscanoff
```

PSfrag 3.0 supporta i comandi "incorporati" `\tex` che si trovano anche nella precedente versione di PSfrag. Usati propriamente sono uno strumento potentissimo, ma sono stati deprecati a causa della necessità di un passo di *pre-processing*. Diversamente dalle precedenti versioni di PSfrag, il supporto per il comando `\tex` deve essere esplicitamente dichiarato, come descritto qua sotto.

Come si può vedere, la sintassi del comando `\tex` è molto simile a quella del comando `\psfrag`, solo che, invece di inserire il comando `\tex` nel file LaTeX, il comando `\tex` è incorporato nel file EPS stesso. In altre parole, il comando diventa il *tag* di sostituzione.

Per esempio, si può piazzare il testo

```
\tex[bl][bl]{$\alpha$}
```

in un punto particolare del file EPS per fare in modo che il LaTeX lo sostituisca con α. Molti utenti PSfrag trovano questa funzionalità utile per etichettare gli assi cartesiani, titoli e legende nei grafici MATLAB.

Il vantaggio di questo approccio è che i cambiamenti possono essere fatti direttamenti nel file EPS senza dover modificare alcun comando `\psfrag` nel file LaTeX. (È comunque necessario ricompilare il file LaTeX in questi casi.)

Ci sono comunque svantaggi e avvertimenti in questo tecnica, che includono:

- Per cambiare le etichette create dai comandi `\tex` si deve modificare la figura; invece se si utilizza `\psfrag`, bisogna solamente modificare il documento, il che può essere meno faticoso. (Si deve comunque eseguire LaTeX in entrambi i casi.)
- Dato che i comandi `\tex` sono lunghe stringhe di caratteri, essi possono alterare il *bounding box* in modo indesiderato. Questo problema può essere mitigato riducendo la dimensione del font della stringa `\tex`, dato che non ha effetto sulla dimensione della sostituzione del PSfrag.
- Il comando `\tex` non è supportato da documenti PostScript compressi.
- Il motore TeX deve scandire il file EPS per trovare queste stringhe, e questo potrebbe aumentare il tempo di eleborazione del documento. (Ad essere onesti, non abbiamo ancora incontrato un caso in cui questo ritardo diventa rilevante.)
- Importante: ogni volta che un file viene scandito da PSfrag, si genera un file `nomefile.pfg`, dove `nomefile` è il nome del file principale del documento LaTeX. Ogni file con lo stesso nome verrà sovrascritto senza avvertimento.

Questa funzionalità non viene da tempo abilitata in automatico, eccetto nella modalità di compatibilità LaTeX 2.09. In questo modo, per i documenti LaTeX, bisogna attivarla in uno dei seguenti modi:

- Per attivare la scansione per una singola figura, far precedere i comandi `\epsfbox` o `\includegraphics` con uan chiamata a `\psfragon`. La scansione verrà terminata all'uscita dall'ambiente circostante; alternativamente, si può disattivarla esplicitamente con una chiamata a `\psfragoff`.
- Per attivare la scansione per un intero documento, usare l'opzione `scanall` nel comando `\usepackage` per il pacchetto PSfrag.

Lo *scanner* `\tex` continuerà ad essere supportato in questa forma; così se si trovano applicazioni nelle quali è preferibile utilizzare il comando `\tex`, lo si utilizzi senza paura!

6 Opzioni del pacchetto

Le opzioni per PSfrag sono solo quattro. Qualsiasi altra opzione non gestita da PSfrag viene inoltrata a `graphics.sty`.

`209mode` (solo in modalità nativa LaTeX 2_ε) fa sì che PSfrag si comporti esattamente come se la modalità di compatibilità LaTeX 2.09 fosse attivata. Di conseguenza, l'allineamento predefinito è `bl`, e la ricerca dei comandi `\tex` è attivata in tutti i file EPS. Questa opzione è utile per convertire vecchi documenti LaTeX 2.09 in LaTeX 2_ε.

La versione di PSfrag per LaTeX2.09 generava un file ausiliario per ogni figura EPS, che conteneva informazioni importanti per la sostituzione. Questi file non sono più usati e possono essere cancellati.

`2emode` (solo in modalità di compatibilità LaTeX 2.09) obbliga PSfrag a rimanere in modalità LaTeX 2_ε anche in presenza di un documento LaTeX 2.09, ovvero il perfetto contrario dell'opzione `209mode`. Quando questa opzione è attivata, l'allineamento predefinito è `Bl`, e la ricerca dei comandi `\tex` è disattivata di *default*.

`scanall` Attiva la ricerca dei comandi `\tex` per *default*. Si consiglia di usare questa opzione se la maggior parte delle figure fa uso di comandi `\tex` incorporati.

`debug` attiva alcune caratteristiche di debug di PSfrag. Inserisce del codice aggiuntivo nel file EPS per disegnare il i bordi dei *bounding box* di ogni porzione di testo che viene sostituita. Probabilmente è utile solo per gli sviluppatori di PSfrag.

7 Un esempio

Nell'esempio che segue mostriamo come usare PSfrag assieme al pacchetto MATLAB. I seguenti comandi MATLAB generano i grafici di una funzione seno e di una funzione coseno, piazzano dei semplici *tag* e sostituzioni `\tex` nella figura, e salvano il risultato in un file EPS, `example.eps`.

```
t = 0:.1:10;
plot(t,sin(t),t,cos(t));
axis('square'); grid;
title('\tex[B][B]{Grafico di $\sin(t)$ and $\cos(t)$}');
```

```
xlabel('\tex[t][t]{$t$}');
ylabel('\tex[B][B]{$\sin(t)$, $\cos(t)$}');
text(t(30),sin(t(30)),'p1');
text(t(60),sin(t(60)),'p2');
text(t(90),sin(t(90)),'p2');
tt=text(t(50),cos(t(50)),'p3');
set(tt,'HorizontalAlignment','center','VerticalAlignment',...
    'bottom','Rotation',atan2(-sin(t(50))*10,2)*180/pi);
print -deps example
```

(In MATLAB, il comando `text` allinea di *default* il testo a sinistra e al centro, come l'argomento `[l]` in `[`$\langle$*psposn*$\rangle$`]`.)

Il codice riportato sotto include il file `example.eps` nel documento LaTeX, dimensionando la figura ad una larghezza di 3.5 pollici. Vari comandi `\psfrag` sono usati per sostituire i *tag* `p1`, `p2` e `p3` nella figura, e il comando `\psfragscanon` è usato per avvisare PSfrag che deve analizzare `example.eps` cercando i *tag* `\tex`.

```
\begin{figure}[tbh]
    \unitlength=1in
    \begin{center}
        \psfragscanon
        \psfrag{p1}[l]{\begin{picture}(0,0)
            \put(0.15, 0.2){\makebox(0,0)[l]{$\sin(t)$}}
            \put(0.1,0.2){\vector(-1,-2){0.1}}
            \end{picture}}
        \psfrag*{p1}[][l]{$\ast$}
        \psfrag{p2}[][l]{$\ast$}
        \psfrag{p3}{$\cos(t)$}
        \includegraphics[width=3.5in]{example.eps}
     \end{center}
     \caption{Un esempio \textsf{psfrag}.}
\end{figure}
```

Notare l'uso di un ambiente `picture` all'interno della sostituzione per `p1`.

Il risultato di questi due passi è mostrato nella figura 2.

7.1 Ridimensionamento della figura

Ci sono due modi di ridimensionare figure EPS con il pacchetto graphics, e ciascuno di essi ha un effetto diverso sulle sostituzioni PSfrag. Chi è solito usare lo stile `epsf.sty` è abituato a uno solo di questi comportamenti.

Usando le macro `\scalebox` o `\resizebox` di `graphics.sty`, le sostituzioni PSfrag saranno ridimensionate assieme alla figura. Questo effetto è illustrato nella figura 3. L'esempio della figura 3 usa il seguente comando per portare le dimensioni della figura a una larghezza di 3.5 pollici:

```
\resizebox{3.5in}{!}{\includegraphics{example.eps}}
```

Al contrario, il codice usato per ottenere la figura 2 usa la parola chiave `width=` di `graphicx.sty`:

```
\includegraphics[width=3.5in]{\includegraphics{example.eps}}
```

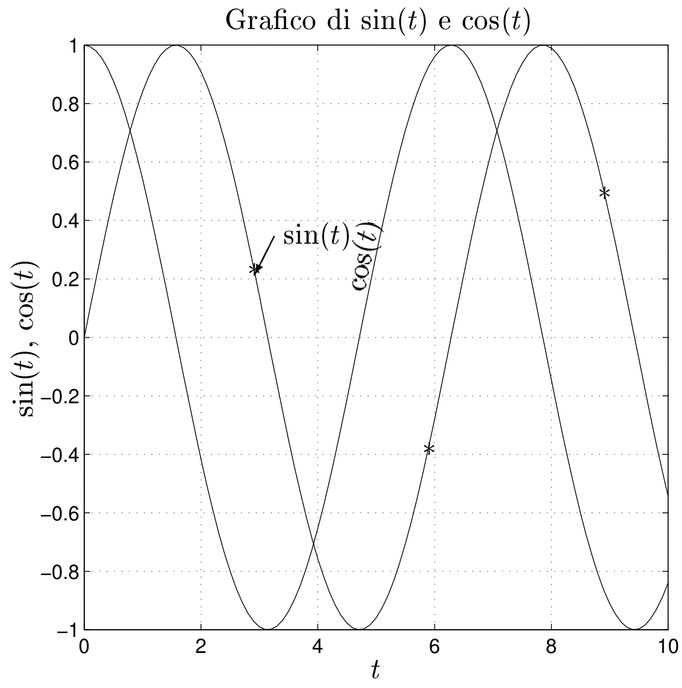

Figura 2: Un esempio PSfrag.

La figura 2 illustra anche il comportamento che si può ottenere usando le macro `epsf.sty`: `\epfxsize`, `\epsfysize`, etc. In questi casi, il testo PSfrag non viene ridimensionato con la figura.

Si vede chiaramente che il testo nella seconda figura è più piccolo che nella prima. Questo succede perché `\resizebox` usa dei trucchi di PostScript per ridimensionare tutti i contenuti del suo argomento mantenendo le proporzioni. Dal momento che i comandi `\psfrag` non sono realmente composti fino a quando non si arriva all'interno del comando `\includegraphics`, anch'essi vengono ridimensionati.

In `graphicx.sty`, le coppie di valori chiave `width=`, `height=`, and `scale=` ridimensionano la figura senza ridimensionare il testo che viene sostituito, purché siano forniti prima di un valore per la rotazione attraverso il parametro `angle=`. Naturalmente le macro `\resizebox` e `\scalebox` sono ancora disponibili in `graphicx.sty` quindi è possibile mescolare i due comportamenti finché si ottiene il risultato desiderato.

Si veda la documentazione di graphics per maggiori dettagli.

Se non si è ancora sicuri su questa distinzione, provare entrambi i metodi per ridimensionare le figura finché non si trova una soluzione soddisfacente.

8 Errori frequenti, problemi noti, e bug

PSfrag è privo di bug.

Beh, naturalmente stiamo scherzando. PSfrag usa alcuni trucchetti PostScript per raggiungere i suoi obiettivi. Quindi non ci meraviglieremmo se qualcuno dovesse trovare dei bug. Se si riscontra qualche problema, si controlli che non sia elencato qui sotto; se non lo è, lo si segnali alla mailing list PSfrag (paragrafo 9).

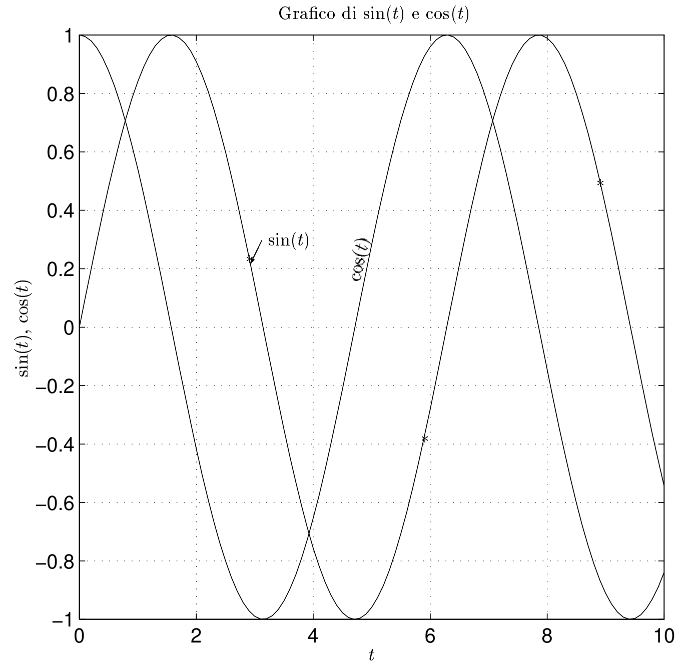

Figura 3: Lo stesso esempio PSfrag della figura 2, usando `\resizebox` per impostare la larghezza.

8.1 Usare correttamente i *tag* di PSfrag

Uno dei problemi in cui gli utenti incorrono più di frequente con PSfrag è che alcuni *tag* vengono rimpiazzati correttamente, ma non tutti. Se possibile, quindi, bisognerebbe progettare le figure con il funzionamento di PSfrag in mente, seguendo questa regola:

> Quando si aggiunge un frammento di testo (un *tag*) in una figura per essere sostituito da PSfrag, usare una sola parola, contenente solo lettere non accentate e numeri.

PSfrag è stato progettato per funzionare in questo modo; seguire questa regola vi dà la garanzia pressoché totale che PSfrag funzioni come promesso. Naturalmente non è sempre possibile seguire questa regola, e un piccolo numero di pacchetti grafici dà costantemente dei problemi. In tutti i casi, questi problemi possono essere risolti capendo in che modo PSfrag cerca questi *tag*.

Il formato PostScript usa cinque comandi per visualizzare del testo—`show`, `ashow`, `kshow`, `widthshow` e `awidthshow`—anche se, in molti casi, un file EPS definirà delle abbreviazioni di questi comandi. PSfrag, in realtà, intercetta questi comandi e controlla se contengono dei *tag* da sostituire. Quando la stringa corrisponde ad un *tag* noto, PSfrag si immagina dove il *tag* *sarebbe* stato piazzato, e inserisce in quel punto l'opportuna sostituzione. Se invece non trova nessuna corrispondenza, PSfrag lascia che il comando `*show` si comporti normalmente.

Le stringhe visualizzate dai comandi `*show` sono delimitate da parentesi, che svolgono un ruolo simile a quello dei doppi apici nel linguaggio `C`. Per esempio:

`(Una prova.) show` ha l'effetto di visualizzare `Una prova.`

Le parentesi che non devono essere interpretate come delimitatori e alcuni altri caratteri speciali devono essere preceduti da un *backslash* (\) in una stringa PostScript. Per esempio:

```
(x = \(0,1]) show            visualizza           x = (0,1]
```

Tenendo conto di questo, ecco la regola per i *tag* di PSfrag:

> Il *tag* passato al comando `\psfrag` dev'essere scritto esattamente come deve comparire nel comando `*show` del file EPS e senza le parentesi delimitatrici.

In altre parole, PSfrag funzionerà solo se la stringa nel comando `\psfrag` è in tutto e per tutto uguale a quanto si trova nel file EPS. Se le stringhe che si vogliono sostituire contengono dei backslash, come nell'esempio di `x = \(0,1]`, allora si deve aggiungere un backslash anche al comando `\psfrag`. PSfrag, inoltre, può sostituire solo stringhe intere, non parti di esse. Quindi se il file EPS contiene

```
(Voglio sostituire XXX qui) show
```

il comando `\psfrag` sbaglia se gli si passa solo `XXX`.

Volendo, si può usare un semplice editor di testo per controllare; i file EPS, infatti, sono (quasi sempre) solo dei semplici file ASCII.

Sfortunatamente, alcuni pacchetti grafici visualizzano il testo passando ogni carattere singolarmente a un comando `show`. In altre parole, usando questi strumenti di disegno per inserire la stringa "prova" in una figura, verrà prodotto un codice di questo tipo:

```
(p) show (r) show (o) show (v) show (a) show
```

Se lo strumento che utilizzate per produrre le figure è di questo tipo, ci scusiamo; usare PSfrag diverrà molto più scomodo—sarà possibile usare solo *tag* composti di un solo carattere. Strumenti di questo tipo impediscono anche l'uso del comando `\tex`.

8.2 Problemi nell'utilizzo di alcune figure xfig

PSfrag non funziona con le figure create con xfig che usano il *pattern fill* ("riempimento con *pattern*"). Quando si vuole colorare/riempire un poligono, xfig fornisce varie scelte possibili: toni di grigio, colori semplici, o alcuni riempimenti più complicati come ad esempio il tratteggio, *checkers* (scacchiera), etc. Purtroppo, l'uso di *pattern fill* in una figura poi elaborata da PSfrag produce dei file PostScript che non verranno stampati.

Per fortuna esistono dei metodi per aggirare questo problema:

1. Non usare riempimenti a pattern nelle figure create con xfig e sostituirli con colori (o con toni di grigio). Per i dettagli consultare la documentazione di xfig.
2. Aprire il file EPS incriminato (generato da fig2dev o dal comando "export" di xfig) con il proprio editor di testo preferito. Cercare la definizione del comando `PATfill`; all'interno di questa subroutine, sostituire `show` com `oldshow` (c'è una sola occorrenza).

Un messaggio per tutti gli *hacker* che modificano il codice PostScript: sia PSfrag che xfig ridefiniscono il comando PostScript `show`. `oldshow` è il comando dove xfig sposta la "vecchia" versione di `show`. Se riuscite a determinare perché questo aggiustamento funziona, e convincete gli sviluppatori di xfig ad attuare questa modifica, o se avete da suggerire un modo per correggere PSfrag, per favore fatelo.

8.3 Problemi utilizzando vecchie versioni del pacchetto seminar

Un pacchetto molto diffuso, seminar, è stato, per un certo periodo, incompatibile con PSfrag 3.0. Questo è dovuto al fatto che PSfrag fa affidamento su alcune caratteristiche della routine di output di LaTeX 2ε , mentre il pacchetto seminar usa ancora una con molti punti in comune con LaTeX 2.09.

La migliore soluzione a questo problema è di assicurarsi di avere la versione più recente del pacchetto seminar che si può trovare in ogni sito CTAN, ad esempio nello stesso posto in cui avete trovato PSfrag. Una pagina web per il pacchetto seminar si trova all'indirizzo `http://www.tug.org/applications/Seminar/`. La versione 13/10/1997 sembra aver corretto il problema.

Se per qualche ragione siete costretti ad usare una versione più vecchia, esiste una soluzione temporanea, specifica per dvips: aggiungere il comando `\special{header=psfrag.pro}` subito prima di `\begin{document}` nel sorgente LaTeX.

9 La mailing list di PSfrag

Esiste una mailing list Majordomo dedicata alla manutenzione di PSfrag[1]. La mailing list *non* serve a sostituire questo manuale o a risparmiarvi una piccola dose di congetture. È invece il luogo ideale per segnalare bug, idee per lo sviluppo, e così via. Chiunque voglia aiutare l'evoluzione di PSfrag si può iscrivere, mandando un'email a

`majordomo@rascals.stanford.edu`

con la riga `subscribe psfrag` nel testo dell'email.

Segnalazioni di bug, idee, etc. dovrebbero essere dirette a

`psfrag@rascals.stanford.edu`.

Se trovate un bug, per favore forniteci i file necessari (un file LaTeX , le figure EPS, etc. in modo che possiamo provare da soli! Provate a fornirci il pù piccolo esempio indipendente che presenta il bug in questione. Se questo non è possibile, scriveteci due righe.

[1]Attenzione, al momento della traduzione del manuale la mailing list di PSfrag non è più attiva. [N.d.T.]