

ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Trieste

INFN/TC-95/24

28 agosto 1995

Stefano Ansoldi:

Introduzione a SEMINAR & PSTricks due stili LATEX per lucidi e grafica PostScript

Introduzione
a
SEMINAR & PSTricks
due stili L^AT_EX
per lucidi e grafica PostScript

Stefano Ansoldi

*Dipartimento di Fisica Teorica dell'Università,
Strada Costiera 11, 34014-Trieste, Italy,*

E-mail address: `ansoldi@vstst0.ts.infn.it`

25 agosto 1995

Contenuto

Contenuto.	ii
Elenco delle figure.	iii
Elenco delle tavole.	v
Elenco degli esempi.	vii
1 Introduzione.	1
2 SEMINAR.	5
2.1 Documentazione e <i>style-options</i> .	5
2.2 Come cominciare.	6
2.3 Intestazione e incorniciatura.	8
2.4 Overlays.	16
2.5 Appunti.	19
2.6 Esempio: come impostare un <i>MAIN file</i> .	24
2.7 Che fare se non si ha PSTricks ?	27
3 PSTricks.	29
3.1 Documentazione e <i>style-options</i> .	29
3.2 PSTRICKS.STY .	32
3.2.1 Alcuni comandi speciali.	37

3.3	PST-PLOT.STY.	40
3.4	MULTIDO.STY.	44
3.5	PST-NODE.STY & PST-COIL.STY.	47
3.6	GRADIENT.STY.	52
3.7	PST2EPS.STY.	54
3.8	CHARPATH.STY, TEXTPATH.STY, COLORTAB.STY.	55
A	Installare Seminar.	57
B	Installare PSTricks.	59
	Indice.	61

Elenco delle figure

2.1	Il più semplice lucido <i>landscape</i> ed ...	8
2.2	... il più semplice lucido <i>portrait</i> .	13
2.3	Elaborazione del precedente lucido <i>landscape</i> e ...	14
2.4	... di quello <i>portrait</i> .	15
2.5	Esempio di utilizzo del comando <code>\overlay</code> : si osservi che può essere utilizzato praticamente ovunque, in modo testo, in modo matematico ed anche in figure!!!	18
3.1	Risultato dell'esempio dei comandi di PSTricks ; si noti la semplicità consentono di controllare anche i più piccoli particolari dei disegni.	35
3.2	Risultato del secondo esempio dei comandi di PSTricks .	36
3.3	Esempio di utilizzo di <code>\pscustom</code> e <code>\psclip</code> in un <code>ps-picture-environment</code> .	39
3.4	Esempio di utilizzo del comando <code>\parametricplot</code> .	43
3.5	<i>Output</i> del comando <code>\psaxes</code> di <code>pst-plot</code> .	43
3.6	Semplice output di un comando <code>\multido!</code>	45
3.7	Risultato di un esempio un pò più complesso sul comando <code>\multido</code> .	46
3.8	Diagramma di commutazione ottenuto sfruttando le possibilità offerte da <code>pst-node</code> .	51
3.9	Utilizzo del qualificatore <code>gradient</code> .	53

Elenco delle tavole

2.1	Files di documentazione e di <i>style-options</i> nella distribuzione di Seminar	12
2.2	Comandi che influenzano la stampa delle note.	23
3.1	Files di documentazione e di <i>style-options</i> nella distribuzione di PSTricks	31

Elenco degli esempi

1	Seminar: lucidi elementari	7
2	Seminar: cornici ed intestazioni	8
3	Seminar: overlay	17
4	Seminar: appunti I°	20
5	Seminar: appunti II°	21
6	Seminar: <i>MAIN file</i>	24
7	PSTricks: grafica I°	34
8	PSTricks: grafica II°	34
9	PSTricks: clipping	38
10	PST-plot: parametricplot	41
11	PST-plot: assi cartesiani	41
12	Multido: multido semplice	44
13	Multido: multido evoluto	44
14	PST-node: diagramma di commutazione	49
15	Gradient: gradient	52

Capitolo 1

Introduzione.

SEMINAR & **PSTRICKS** sono due pacchetti che aumentano considerevolmente le possibilità, peraltro tutt'altro che limitate, di \LaTeX . Il primo di essi è stato pensato esplicitamente per agevolare la preparazione di lucidi per conferenze o seminari, il secondo invece consente di utilizzare direttamente in \LaTeX (e \TeX !) le potenzialità proprie del linguaggio **PostScript** rendendo particolarmente agevole l'esecuzione di disegni e grafici da inserire nei propri lavori.

I pacchetti sono molto sofisticati, in grado di soddisfare praticamente tutte le esigenze, e vengono forniti con numerosi files di stile che consentono di sfruttarne appieno le potenzialità in maniera molto immediata.

È praticamente impossibile conoscere completamente tutte le molteplici funzioni di questi pacchetti, senza averli usati più e più volte. Per questo si cercherà di precisare soprattutto quello che è *possibile fare* piuttosto che *come farlo*. Poi ognuno potrà approfondire la conoscenza, tramite i manuali e la pratica, delle funzioni del pacchetto che gli risultano più utili, tralasciando completamente le altre ma ottenendo comunque risultati più che soddisfacenti.

Prima di presentare le caratteristiche salienti di questi stili, si esporranno alcuni problemi che si possono avere utilizzandoli e che potrebbero scoraggiare un primo approccio verso di essi:

- i pacchetti sono molto “*vasti*” ed impararne tutte le caratteristiche in maniera sistematica può essere lungo e non molto divertente;
- non è possibile passare il file sorgente ad amici o colleghi che non possiedono i pacchetti.

Entrambe le difficoltà però possono essere aggirate. In primo luogo, come già detto, non è necessario imparare tutte le funzioni, ma solo quelle che interessano per il nostro scopo.

In secondo luogo, l'installazione¹ dei pacchetti non è particolarmente problematica e quindi si possono spedire a chiunque li voglia senza particolari problemi (ad esempio **Seminar** richiede semplicemente di avere alcuni files in una delle directory di input di \LaTeX).

Inoltre è sempre possibile con un minimo di tempo in più, soprattutto in relazione ai files in **PSTricks**, organizzare il sorgente in modo da poter processare separatamente le figure, e poterle poi inserire come si fa di solito, tramite macro come *psfig*, o *epsfig* spedendo poi questa seconda versione.

Questi lievi svantaggi sono compensati da vantaggi considerevoli. In primo luogo si ha un consumo di spazio su disco molto minore in quanto tutti i documenti vengono conservati come files di testo. Inoltre è più comodo apportare correzioni alle figure all'interno del file \LaTeX ed in più si ha un notevole guadagno estetico perchè lo stile dei disegni è uguale a quello del testo. A queste "pignolerie" si aggiunge naturalmente il fatto che sfruttando le potenzialità offerte dai pacchetti è possibile ottenere risultati di molto superiori a quelli normalmente ottenibili in altri modi ed in molto meno tempo.

Per concludere è importante osservare che i programmi non sono solo stili \LaTeX , ma possiedono una tale gamma di *style-options*, che le necessità più comuni possono essere soddisfatte anche sapendo usare solo superficialmente i pacchetti, il che può essere ottenuto facilmente studiando i files di esempio.

Tre semplici precisazioni:

- come si vedrà in seguito, certe opzioni di **Seminar** per funzionare richiedono che sia installato **PSTricks**;
- **PSTricks** definisce i nuovi comandi tramite istruzioni `\special` che di solito non sono riconosciute da *previewer per .dvi*, come *XDVI*; per vedere ciò che realmente verrà stampato è quindi necessario " $\text{\LaTeX}are$ " il file e poi processarlo anche con *DVIPS*, ad esempio, osservando quindi il risultato con *GV* o equivalenti visori **PostScript**;
- se si è installato **Seminar**, ma non **PSTricks**, si possono ugualmente ottenere buoni risultati utilizzando le possibilità aggiunte

¹Vedi appendici A-B

Capitolo 2

SEMINAR.

2.1 Documentazione e *style-options*.

Seminar è uno stile \LaTeX che consente di preparare in maniera molto efficace *lucidi* per conferenze/seminari e degli eventuali *appunti* di accompagnamento.

I files della distribuzione si dividono in due categorie:

1. files di documentazione :

SEM-USER.TEX è il file sorgente del manuale di **Seminar**;
SEMSAMP1.TEX e **SEMSAMP2.TEX** sono due files di esempi;
FANCYBOX.TEX è il file sorgente del manuale di **Fancybox**;

2. files di *style-options*, cioè :

semrot opzione che consente di ruotare automaticamente le trasparenze *portrait* (o *landscape*) in modo da poter stampare tutti i lucidi contemporaneamente;

semcolor attiva la possibilità di utilizzare colori nella stesura dei lucidi;

semlayer assieme a *semcolor* consente di usare l'opzione *overlay*;

xcomment opzione per scegliere come gestire la stesura delle note che accompagnano le trasparenze;

slidesec opzione che introduce comandi di sectioning & heading;

fancybox opzione che facilita la gestione di *LR*-boxes in \LaTeX , soprattutto in relazione alla creazione di *frames*;

semhelv e *semcmss* opzioni che predefiniscono certi tipi di font ma funzionano solo sotto il **NFSS**.

Tra le opzioni precedenti *semrot*, *semcolor* e *semlayer* caricano il pacchetto **PSTricks**, che quindi deve essere installato.

2.2 Come cominciare.

Sfruttando appieno le possibilità offerte da **Seminar** è possibile semplificare notevolmente la stesura di lucidi per seminari e conferenze con la possibilità di scrivere, al contempo, degli appunti di accompagnamento ed ottenendo un manuale già bello e pronto per essere distribuito; è possibile creare sia trasparenze “*landscape*” che “*portrait*”, dimensionandole a piacere e fornendole di cornici ed intestazioni personalizzabili. Nel seguito vengono descritte le principali caratteristiche, che è indispensabile conoscere per cominciare ad usare il pacchetto. Per informazioni più complete si consiglia di consigliare direttamente il manuale di **Seminar**.

Come già accennato **Seminar** è uno stile \LaTeX . Ciò significa che per usarlo si deve adoperare la solita struttura, cioè

```
\documentstyle{seminar}
```

seguito da

```
\begin{document} ... My document ... \end{document}
```

I principali *nuovi* comandi aggiunti al \LaTeX standard da **Seminar** sono

```
\begin{slide}[-,-] ... \end{slide}
```

```
\begin{slide*}[-,-] ... \end{slide*}
```

dei quali il primo serve a generare un lucido *landscape*, ed il secondo a generare un lucido *portrait*¹. Per default il documento viene **processato** in *landscape-mode* e quindi quando si spedisce alla stampante l'output di *DVIPS* si deve usare il comando

```
PRINT/PARAMETER=(PAGE_ORIENT=LANDSCAPE)
```

o altri equivalenti, dipendenti dalla stampante che si usa; cioè il documento deve essere **stampato** in *landscape-mode*.

Alternativamente è possibile **processare** il documento in *portrait-mode* inserendo la *style-option* *portrait*, con il comando

```
\documentstyle[... ,portrait, ...]{seminar}
```

La scelta dipende dai gusti, comunque quando si processa il file, un messaggio di avvertimento ricorda se è necessario stampare il documento in *landscape-mode*.

Ecco quindi il più semplice file **Seminar** che si può scrivere:

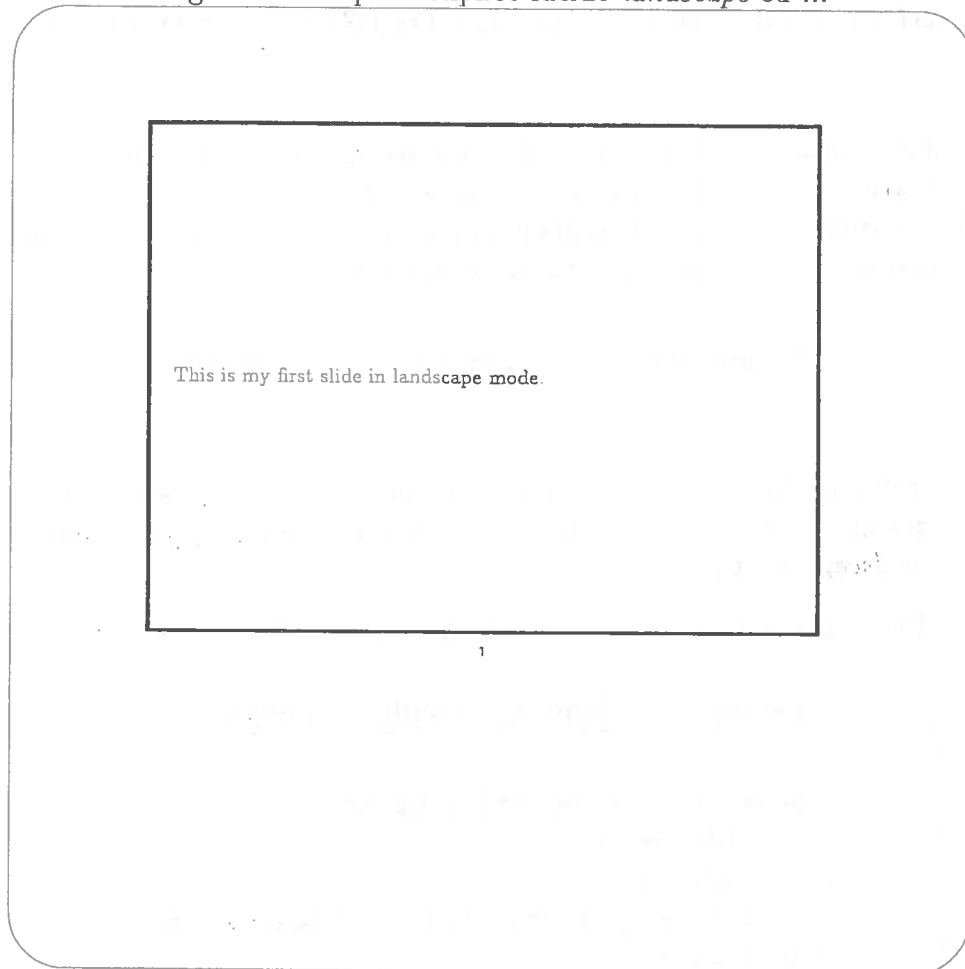
Esempio 1 (Seminar: lucidi elementari)

```
\documentstyle[semrot]{seminar}
\begin{document}
\begin{slide}[-,-]
  This is my first slide in landscape mode.
\end{slide}
\begin{slide*}[-,-]
  This is my first slide in portrait mode.
\end{slide*}
\end{document}
```

¹[-,-] è un parametro opzionale che consente di definire delle dimensioni per il lucido diverse da quelle di default.

Con esso si ottengono due lucidi, il primo *landscape* ed il secondo *portrait*. Nelle figure (2.1)-(2.2) c'è il risultato, che a dire il vero non sembra molto incoraggiante; però in breve, sulla stessa struttura base, si potranno aggiungere interessanti sviluppi. Si noti comunque che il materiale è centrato verticalmente, cosa che viene fatta in maniera automatica. Anche la gestione degli spazi, sempre verticali, è automatica entro certi limiti, in modo da consentire l'ottimale posizionamento dell'output su una trasparenza.

Figura 2.1: Il più semplice lucido *landscape* ed ...



2.3 Intestazione e incorniciatura.

Partendo dalla configurazione precedente, l'aggiunta di una cornice e delle intestazioni superiore ed inferiore è una cosa molto rapida, come si vede dall'esempio che segue:

Esempio 2 (Seminar: cornici ed intestazioni)

```
\documentstyle[fancybox,semrot]{seminar}
\newpagestyle{prova}
    {Dipartimento di Fisica Teorica ...}
    {\hfil It is not very difficult ...}
\slidesmag{2}
\rotateheaderstrue

\begin{document}

\slideframe[...]{shadow}
\pagestyle{prova}

\begin{slide}
    This is my first slide in landscape mode.
\end{slide}

\begin{slide*}
    This is my first slide in portrait mode.
\end{slide*}

\end{document}
```

Il comando da usare per l'incorniciatura è

$$\backslash\text{slideframe}[param]\{name\}$$

in cui le possibilità di default per *name* sono

1. **none**
2. **plain**

oppure aggiungendo l'opzione *fancybox* una delle seguenti:

1. **shadow**
2. **double**

3. `oval`

4. `Oval`

L'aggiunta dell'opzione `semcolor` consente anche:

1. `scplain`

2. `scdouble`

3. `scshadow`

Visto che la maggior parte dei tipi di cornici valide sono dimensionabili a piacere, l'argomento opzionale [*param*] consente di passare un parametro al comando che definisce la cornice².

Le intestazioni si ottengono invece con il comando

```
\pagestyle{name}
```

dove *name* può essere:

1. `empty`

2. `plain`

3. `headings`

4. `myheadings`

5. `align` che mette dei “+” agli angoli della pagina.

I primi 4 `pagestyles` di questo elenco funzionano come nel \LaTeX standard, ma si possono anche definire degli stili personali, come nell'esempio (2), sfruttando il comando

```
\newpagestyle{name}{header}{footer}
```

²Ad esempio con l'opzione `Fancybox` si aggiunge il comando `\shadowbox[stuff]`, che consente di generare una cornice con un'ombra di larghezza `shadowsize`; Volendo modificare questa larghezza per le ombre dei lucidi, si può ad esempio utilizzare il comando `\slideframe[\setlength{shadowsize}{24pt}]{shadow}`. Questo è utile usando `PSTricks` (vedi seguito!) perchè il parametro opzionale può essere usato per cambiare alcuni attributi `PostScript`.

in cui *name* è il nome del nuovo stile di pagina da usare in `\pagestyle`, *header* è il contenuto dell'intestazione superiore e *footer* è il contenuto dell'intestazione inferiore.

Il comando

$$\backslash\text{slidesmag}\{n\}$$

dove $-5 \leq n \leq 9$, consente di gestire l'ingrandimento dei font in base alla quantità di informazioni da inserire sui lucidi e va usato solo nel *preamble*. Il default è $n = 4$.

Sempre nel *preamble* si può usare il comando

$$\backslash\text{rotateheaderstrue}$$

che consente di ruotare le intestazioni in modo da averle nella posizione corretta sia per lucidi *landscape* che per lucidi *portrait*.

Le figure (2.3)-(2.4) mostrano il risultato, già diverso da quello precedente!!!

FILES DI DOCUMENTAZIONE:

- SEM-USER.TEX
- SEMSAMP1.TEX
- SEMSAMP2.TEX
- FANCYBOX.TEX

FILES DI OPZIONI:

1. semrot
2. semlayer
3. semcolor
4. xcomment
5. slideseq
6. fancybox
7. semhelv
8. semcmss

Tavola 2.1: Files di documentazione e di *style-options* nella distribuzione di **Seminar**.

Figura 2.2: ... il più semplice lucido *portrait*.

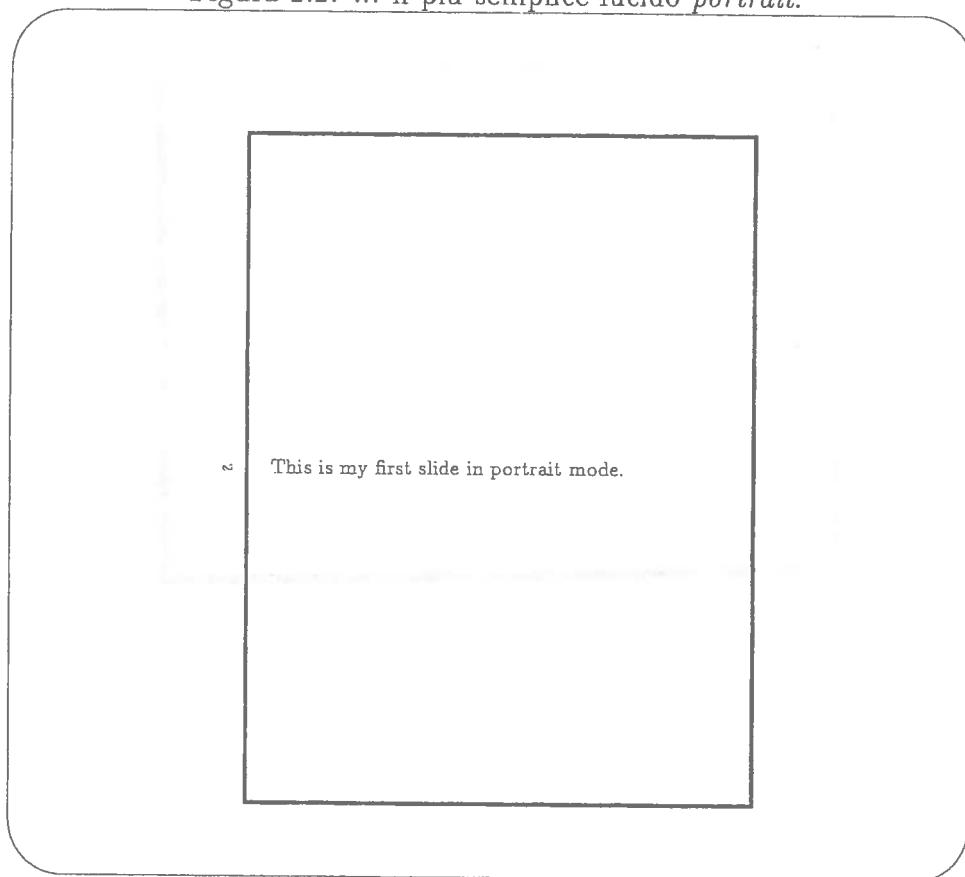


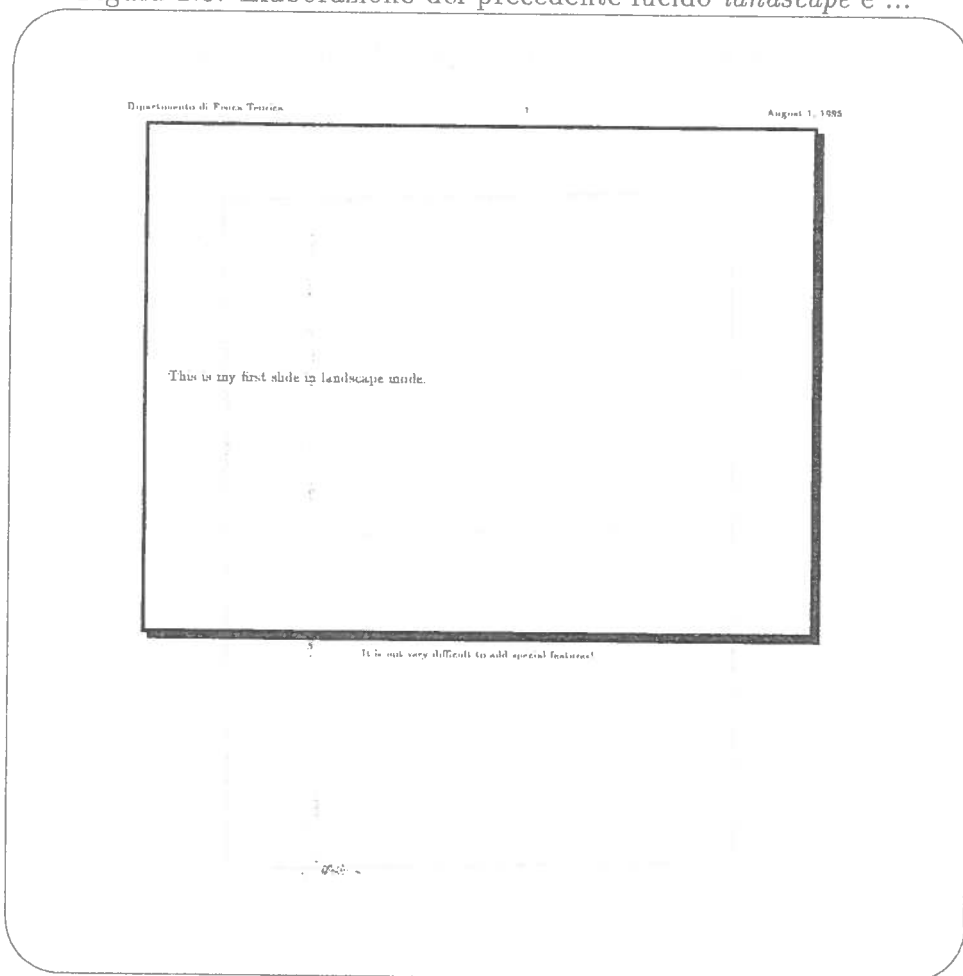
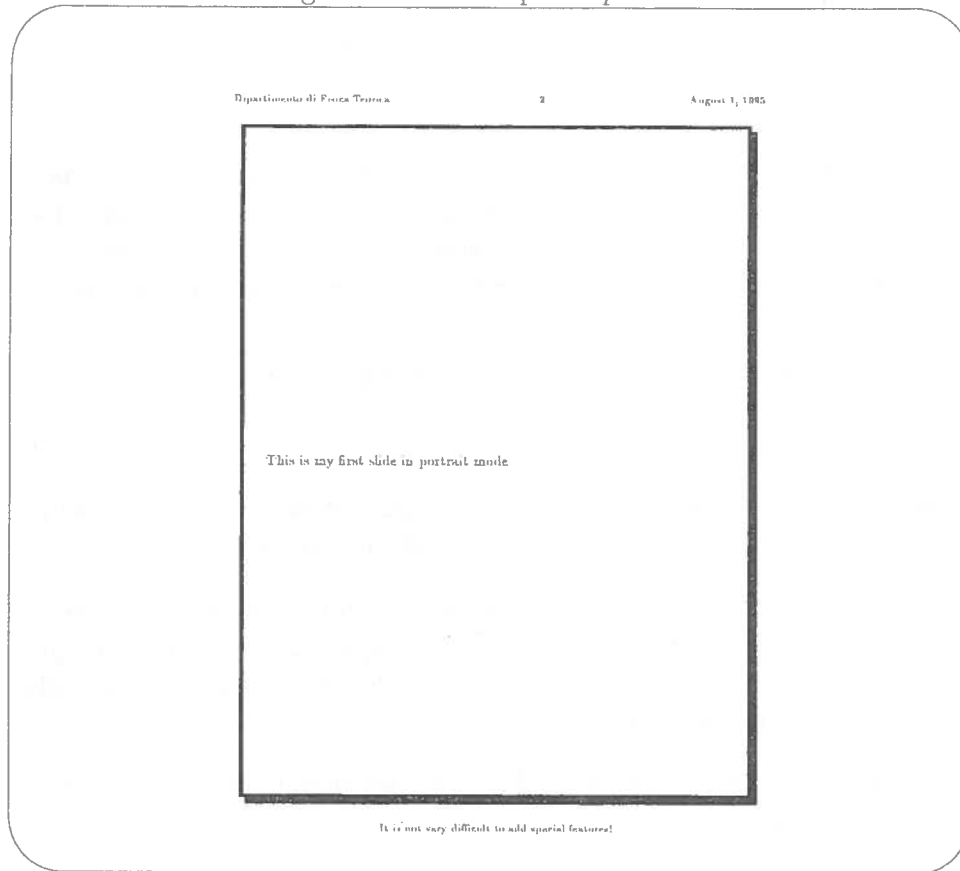
Figura 2.3: Elaborazione del precedente lucido *landscape* e ...

Figura 2.4: ... di quello *portrait*.



2.4 Overlays.

Una delle possibilità più interessanti fornita da **Seminar** è certamente quella di creare delle trasparenze adatte per aggiungere gradualmente delle informazioni durante l'esposizione. I comandi che consentono questo sono:

$$\backslash\text{overlay}\{n\}\{\dots\}$$

$$\backslash\text{begin}\{\text{overlay}\}\{n\} \dots \backslash\text{end}\{\text{overlay}\}$$

in cui n può assumere un valore intero tra 0 e 9. Tali comandi possono essere messi praticamente ovunque nel documento, in testo, formule matematiche e figure, come mostrato nell'esempio (3). Per usarli bisogna utilizzare l'opzione `semcolor` unitamente a `semcolor`, cioè

$$\text{documentstyle}[\dots, \text{semcolor}, \text{semcolor}] \{\text{seminar}\}$$

Tutti i $\text{T}_{\text{E}}\text{X}$ -*groups* che contengono `\overlay0` finiscono su un primo lucido, quelli con `\overlay1` su un secondo e così via.

Come si può capire dal nome stesso l'opzione `semcolor` consente qualcosa in più, cioè di preparare delle trasparenze a colori. Ciò può ottenersi anche se non si ha una stampante **PostScript** a colori, tramite una fotocopiatrice a colori³.

³Il modo per utilizzare dei colori nella stesura dei lucidi consiste di due passi:

- definire i colori che si vogliono usare tramite il comando

$$\backslash\text{colorlayers}\{\text{colore1}, \text{colore2}, \text{colore3}\}$$

- in seguito quando si vorrà usare il colore *colore1* si avrà a disposizione il comando $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ `\colore1` e così per gli altri colori definiti; i comandi dei nuovi colori funzionano come i comandi per cambiare tipo di font, e quando si stampa il file, si ottengono per ogni lucido con più colori tante copie, ciascuna con le parti di testo dello stesso colore; facendo più passaggi del lucido tramite fotocopiatrice a colori si ottengono allora dei lucidi colorati.

Di seguito è mostrato una parte di sorgente \LaTeX che produce come risultato in fase di stampa i quattro lucidi di figura (2.5). È un esempio di utilizzo del comando `\overlay` che consente di apprezzarne l'estrema versatilità:

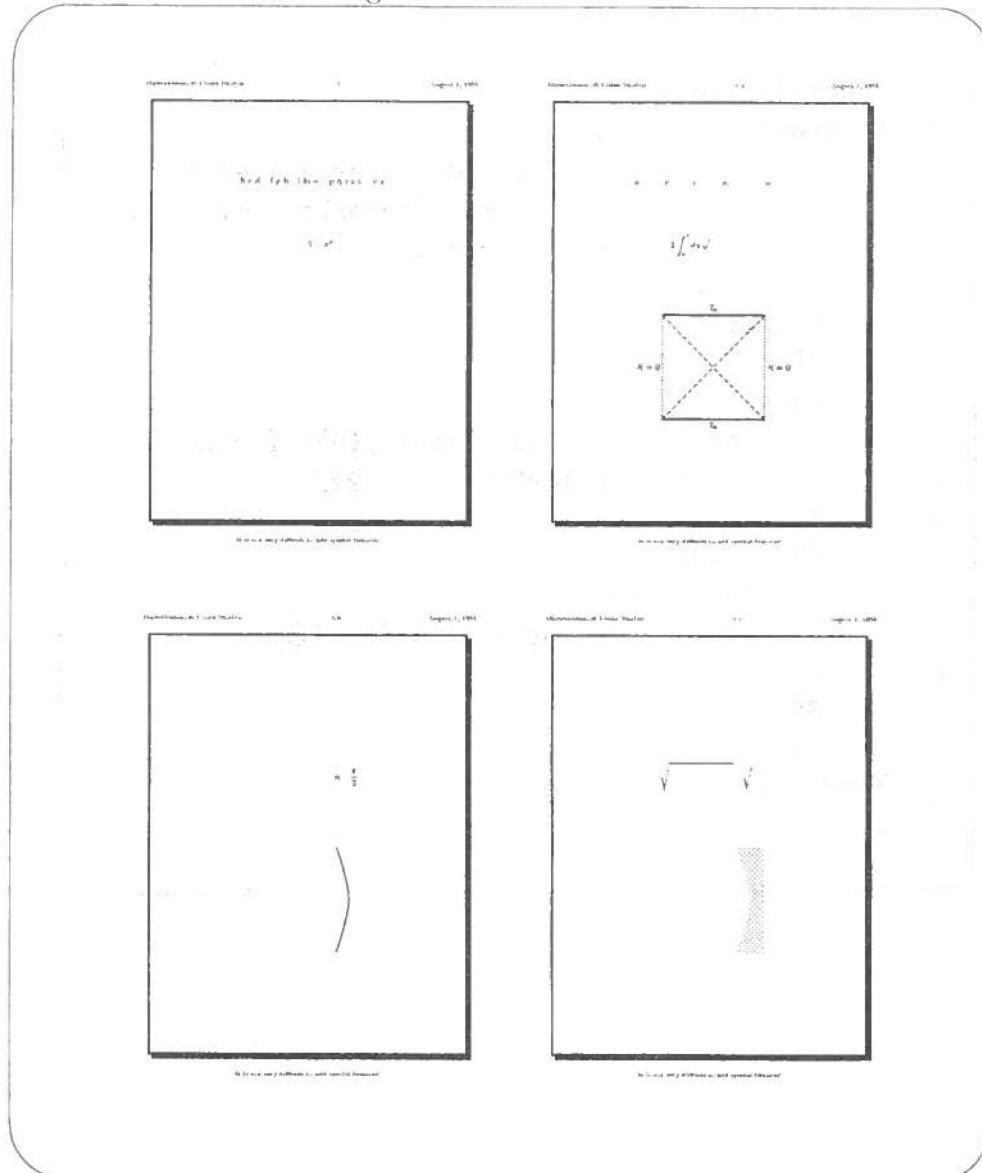
Esempio 3 (Seminar: overlay)

```

...
\begin{slide*}
\centerline{\overlay0
  {\overlay1 a} b c d {\overlay1 e} f g h
  {\overlay1 i} l m n {\overlay1 o} p q r
  s t {\overlay1 u} v z
  }
\vskip 1 cm
$$
{\overlay3
  \sqrt{\overlay1 2 \int_{0}^{1} dx
  \sqrt{\overlay0 1 - x^{2}}}}
}
{\overlay3
  {\overlay2 =}
  \sqrt{\overlay 2 \frac{\pi}{2}}}
}
$$
...
\end{slide*}
...

```

Figura 2.5: Esempio di utilizzo del comando `\overlay`: si osservi che può essere utilizzato praticamente ovunque, in modo testo, in modo matematico ed anche in figure!!!



2.5 Appunti.

Oltre alle possibilità già descritte, un'altra delle peculiarità di **Seminar**, che può renderlo oltremodo utile nella preparazione di lucidi, è il fatto che, con poco lavoro in più, è possibile affiancare alla stesura delle trasparenze quella degli appunti di accompagnamento; questi possono fungere da promemoria per chi espone o possono essere stampati insieme ai lucidi per lasciare delle note a chi segue. Le *style-options* disponibili sono:

- notes;
- notesonly,notesonly*;
- slidesonly;
- article;
- portrait;
- 10pt, 11pt, 12pt, ...;
- (*xcomment*).

Tra queste la principale è

notes

da inserire al solito in

```
\documentstyle[... ,notes, ...]{seminar}
```

che predispone il pacchetto ad accettare l'introduzione di appunti. Aggiungendo `notesonly` si può evitare di stampare i lucidi e stampare solo gli appunti⁴, mentre aggiungendo `slidesonly` si stampano solo le trasparenze. L'opzione `article` definisce in maniera particolare la formattazione del documento e `portrait` lo stampa in orizzontale, a forma di manuale.

⁴`notesonly*` consente di aggiungere una segnalazione al posto di ogni lucido.

Solo *xcomment* non deve essere inserito tra le *style-options* perchè viene utilizzato per default a meno che non si ponga `\noxcomment` nel *preamble*.

Ci sono molti semplici comandi che influenzano l'output (alcuni sono riportati nella tavola (2.5)), ed i più significativi sono quelli del grado di ingrandimento dei lucidi

`\slidesmag{n}`

e del testo

`\articlemag{n}`

perchè è quasi sempre necessario aggiustare l'ingrandimento delle note e quello dei lucidi, in modo da avere un corretto output in cui trasparenze e note non si sovrappongono. Trovare il giusto compromesso tra leggibilità e ordine comunque di solito non è difficile.

Il comando

`\ptsize{n}`

consente di scegliere il numero di punti del font (*assicurarsi che ciò che si sceglie sia realmente disponibile!*), cosa che può essere fatta anche a livello di *style-option*; invece

`\slideplacement{name}`

permette di decidere dove mettere i lucidi (al pari di ciò che accade in \LaTeX con il comando `\begin{figure}...\end{figure}`)⁵.

Per aggiungere gli appunti ci sono le due possibilità illustrate nei due esempi che seguono:

⁵Vedi la tavola (2.5) per le possibilità del qualificatore *name* del comando `\slideplacement{name}`.

Esempio 4 (Seminar: appunti I°)

```

\documentstyle[... ,notes]{seminar}
\begin{document}
\begin{slide*}
...
\end{slide*}
Qui si possono scrivere le note,
blah, blah, blah, ...
\begin{slide}
...
\end{slide}
...
\begin{allversions*}
  Scrivere qui tutto cio' che non
  si vuole venga tolto quando si
  usa l'opzione slidesonly.
\end{allversions*}
...
\end{document}

```

Esempio 5 (Seminar: appunti II°)

```

\documentstyle[... ,notes]{seminar}
\noxcomment
...
\begin{document}
...
\begin{note}
  Aggiungere qui gli appunti, ecc.
\end{note}
...
\end{document}

```

Il primo metodo (esempio (4)) consiste, una volta aggiunta l'opzione `notes`, nello scrivere gli appunti fuori dagli *slide-environments*. Quando si vogliono processare solo i lucidi si aggiunge l'opzione

slidesonly

e \LaTeX nel processare il file ignorerà tutto quello che è all'esterno degli *slide-environments*.

Se ci fosse la necessità di aggiungere qualcosa fuori da essi, che deve essere assolutamente processato, si ha a disposizione l'*environment*

```
\begin{allversions*} .... \end{allversions*}
```

che è consigliato *solo* per introdurre cambiamenti di stile e font, non per introdurre del testo.

Nell'esempio (5) è mostrato il secondo modo per aggiungere gli appunti, che consiste nel mettere nel *preamble* il comando `\noxcomment`. Si ha allora a disposizione un nuovo *environment*

```
\begin{note} .... \end{note}
```

all'interno del quale va messo il testo \LaTeX degli appunti. Eventuali comandi di font, ecc., possono essere tranquillamente messi fuori dagli *slide-environment* ed anche dai *note-environment*, e vengono processati sempre.

- `\slidesmag{n}`
- `\articlemag{n}`
- `\ptsize{n}`
- `\slideplacement{name}` dove name può essere:
 1. float
 2. float*
 3. onepercol
 4. twopercol
 5. here
 6. here*

Tavola 2.2: Comandi che influenzano la stampa delle note.


```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{document}

\begin{allversions*}
  \raggedslides[Opt]           % Lucidi ‘giustificati’
  \slideframe{scshadow}       % Stile delle cornici.
  \pagestyle{mypagestyle}     % Stile di pagina.
  \setlength{\fboxsep}{10pt}  % Separazione di 10 pt.
                              % per i riquadri.
  \setlength{\unitlength}{1cm} % L’unita’ e’ il cm.
\begin{allversions*}

\input{sempst_00.tex}         % Ricordarsi di terminare
\input{sempst_01.tex}         %      questi files
\input{sempst_02.tex}         %      con \endinput.
...
\end{document}

```

2.7 Che fare se non si ha PStricks?

Un solo avvertimento. Se non si possiede il pacchetto PStricks non è possibile usare l’opzione

semrot

cioè non si possono stampare contemporaneamente i lucidi *landscape* e quelli *portrait*. Allora il da farsi è:

1. aggiungere dapprima nel *preamble* il comando

```
\landscapeonly
```

e stampare gli slide *landscape*;

2. togliere `\landscapeonly` e aggiungere

```
\portraitonly
```

mettendo tra le *style-options* `portrait`

```
\documentstyle[... ,portrait]{seminar}
```

e stampare così gli slide `portrait`.

Questo procedimento non crea problemi nella numerazione nè nei riferimenti.

Capitolo 3

PSTricks.

3.1 Documentazione e *style-options*.

PSTricks è uno stile molto più vasto di Seminar (come si vede dai files presenti nella distribuzione, tavola (3.1)) e consente di utilizzare istruzioni PostScript direttamente in L^AT_EX & T_EX. Più precisamente definisce dei nuovi comandi, che consentono di usare le potenzialità del PostScript senza dover conoscere approfonditamente il linguaggio di programmazione.

La distribuzione contiene non solo lo stile PSTricks, ma anche altri mini stili aggiuntivi, che sono elencati nella tavola (3.1) e verranno descritti brevemente nel seguito uno per uno (con l'eccezione dei meno interessanti). Nel complesso il pacchetto si presenta come una potente interfaccia grafica. Le potenzialità che si acquisiscono vanno ben al di là della semplice possibilità di tracciare linee, cerchi, spezzate: un'ampia gamma di qualificatori consente infatti di gestire tramite semplici parametri, il colore del testo, lo spessore delle linee, l'aggiunta di speciali terminatori (freccie e simili), il tracciamento di assi cartesiani, la costruzione di diagrammi di flusso, ecc., ecc..

Per questo non è conveniente imparare "in massa" tutte le possibilità offerte dal pacchetto, ma, una volta apprese le conoscenze di base (cioè la filosofia del programma e la struttura fondamentale della sintassi), si può cercare di volta in volta come risolvere le necessità grafiche che si hanno.

Nel seguito vedremo alcuni esempi in cui verrà evidenziato lo spirito di questo pacchetto, in modo che chiunque voglia usarlo abbia già le

idee di base necessarie per poter affrontare il manuale senza perdere tempo. Prima però si descrivono brevemente alcuni dei files presenti nella distribuzione (vedi tavola (3.1)):

- Files di documentazione:

PST-USR%.PS e **PST-QUICK.PS** costituiscono il manuale di **PSTricks**;

SAMPLES.TEX e **TP-TEST.TEX** sono files \LaTeX di esempi, generali nel primo, specifici per l'opzione `textpath` nel secondo;

PST-TEST.TEX e **FILETEST.DAT** sono files \TeX di esempi, il secondo dei quali utilizza i files **FILETEST.DAT** e **FILETEST.RAW**;

CHARTEST.TEX, **DENIS1.TEX**, **DENIS2.TEX** sono files \TeX di prova per l'opzione `charpath`.

- Files di opzioni: tutti questi files, nella doppia versione (cioè sia con l'estensione `.TEX` che `.STY`) sono files di *style-options* che devono essere "caricati" da \TeX o \LaTeX ¹.

¹Per quanto riguarda \TeX viene caricato il file con l'estensione `.TEX` tramite un comando `\input` mentre in \LaTeX viene sfruttato il file `.STY` quando si aggiunge la *style-options* nel `\documentstyle`.

1. FILES DI DOCUMENTAZIONE:

- PST-USR%.PS
- PST-QUICK.PS
- SAMPLES.TEX (\LaTeX)
- PST-TEST.TEX (\TeX)
- FILETEST.TEX (\TeX)
- CHARTEST.TEX (\TeX)
- DENIS1.TEX (\TeX)
- DENIS2.TEX (\TeX)
- TP-TEST.TEX (\LaTeX)

2. FILES DI OPZIONI:

- `pstricks`
- `pst-plot`
- `multido`
- `pst-coil`
- `pst-node`
- `gradient`
- `pst2eps`
- `charpath`
- `textpath`
- `colortab`

Tavola 3.1: Files di documentazione e di *style-options* nella distribuzione di **PSTricks**.

3.2 PSTRICKS.STY.

Il primo stile di cui si parlerà è il principale, cioè PSTRICKS. Questo stile aggiunge la gran parte dei comandi grafici:

- comandi per lavorare a colori:
 - `\black`, `\darkgray`, `\gray`, `\lightgray`, `\white`,
 - `\red`, `\green`, `\blue`, `\cyan`, `\magenta`, `\yellow`
- un comando per settare vari parametri PostScript:
 - `\psset`²
- comandi per tracciare spezzate e/o curve aperte e/o chiuse ed altri oggetti geometrici e per disegnare griglie di riferimento che aiutino nell'esecuzione del disegno:
 1. `\psline`, `\pspolygon`, `\psframe`, `\pscircle`, `\pswedge`, `\psellipse`, `\psarc`
 2. `\psbezier`, `\parabola`, `\pscurve`, `\psccurve`, `\psdots`
 3. `\grid`
 4. `\pscustom`, `\psclip`, `\endpsclip`³

La sintassi quasi generale è

$$\backslash\text{command} [\text{opt.}](x_1, y_1) \dots (x_n, y_n)\{arg\}$$

²Ad esempio:

`\psset{yunit=1pt}` fissa in un punto l'unità di misura lungo x ;

`\psset{xunit=1cm}` fissa in un centimetro l'unità di misura lungo y ;

`\psset{runit=1in}` fissa in un pollice l'unità di misura "radiale".

Per completezza si ricorda che si può settare anche l'unità di misura della coordinata angolare, che unita a quella radiale, definita prima in `\psset` consente di definire un sistema di coordinate polari nel foglio! Ciò si fa con il comando,

$$\backslash\text{degrees}[13]$$

il quale ad esempio stabilisce che gli angoli vengono misurati in *tredicesimi* di angolo giro. Consultare il manuale di PStricks per ulteriori informazioni.

³Questi comandi sono trattati in maniera più estesa con un esempio loro dedicato nella sottosezione (3.2.1).

in cui tra parentesi quadre vanno gli argomenti opzionali. Le parentesi tonde delimitano sempre una coppia di coordinate (solo in un paio di casi queste ultime sono considerate parametri e vanno tra parentesi graffe), mentre le parentesi graffe sono riservate ad eventuali argomenti (cioè non sono necessariamente presenti in ogni comando), ad esempio nei comandi che ruotano qualcosa, ecc.

Importante è il comando

$$\backslash\text{psset}\{arg\}$$

che consente di settare vari parametri PostScript per le istruzioni che seguono. È comunque sempre possibile settare questi parametri utilizzando la possibilità di introdurre gli argomenti opzionali nel comando grafico.

Un altro comando che può essere utile è

$$\backslash\text{psgrid}[opt.](x_1,y_1)(x_2,y_2)(x_3,y_3)$$

che traccia una griglia con angoli opposti (x_2,y_2) e (x_3,y_3) ed origine (se si vogliono le label) in (x_1,y_1) . Numero delle suddivisioni, spessore delle linee principali e secondarie, tipo e posizione delle labels, colore, possono essere controllati tramite vari parametri.

Di seguito è riportata la sintassi di altri comandi:

- $\backslash\text{psline}*[par]\{arr\}(coor1)\dots(coorn)$
- $\backslash\text{pspolygon}*[par](coor1)\dots(coorn)$
- $\backslash\text{psframe}*[par](coor1)(coor2)$
- $\backslash\text{pscircle}*[par](centro)\{raggio\}$
- $\backslash\text{pswedge}*[par](centro)\{raggio\}\{ang1\}\{ang2\}$
- $\backslash\text{pscurve}*[par]\{arr\}(coor1)\dots(coorn)$
- $\backslash\text{psdot}*[par](coor1)\dots(coorn)$

I parametri grafici (indicati con *par* nelle righe precedenti) non influenzano solo lo spessore e il colore delle linee, ma anche il tipo (ad esempio

si può usare `[linestyle=solid]` o `[...=dotted]` o `[...=dashed]`) personalizzando ad esempio il pattern (provare `[linestyle=dashed, dash=15pt 5pt]`). È addirittura possibile scegliere se avere gli spigoli arrotondati o vivi (parametro `linearc`) e così via.

Le frecce agli oggetti possono essere aggiunte, ove permesso, sostituendo ad `arr` vari simboli, `{>}` `{<}` `{>>}` `{|->}` ad esempio, nelle più svariate combinazioni.

Degno di nota è il parametro `[showpoints=true]` che pone dei *mark* nelle coordinate che si assegnano, in modo da poter correggere più facilmente il disegno.

L'asterisco("`*`") in tutti i comandi è **opzionale** e consente di eseguire un `fill` automatico del `path` del colore `fillcolor` e nello stile `fillstyle` entrambi settabili tramite `\psset`.

Di seguito sono riportati alcuni esempi di utilizzo di questi comandi, il cui risultato grafico compare nelle figure (3.1)-(3.2).

Esempio 7 (PSTricks: grafica I°)

```
\pspolygon[fillcolor=black,fillstyle=crosshatch]
          (0,3)(2,6)(5,6)(7,3)(5,0)(2,0)
\pscicle[fillcolor=darkgray,fillstyle=solid]
          (3.5,3){3}
\psline[linecolor=lightgray]{>}(0,3)(8.5,3)
\psline[linestyle=dashed,dash=3pt 1pt,%
        linecolor=white]{|->>}(3.5,0)(3.5,6)
```

Esempio 8 (PSTricks: grafica II°)

```

\begin{pspicture}(-.5,-.5)(8.5,6.5)
  \psgrid[subgriddiv=1,griddots=10,%
          gridlabels=7pt](8,6)
  \psline[linewidth=2pt,lineararc=10pt]{>-<}%
        (7.8,5.8)(0.2,5.8)(0.2,0.2)(7.8,0.2)
  \psbezier[linewidth=1pt,showpoints=true,%
            linecolor=darkgray]{|-|}%
        (0.5,0.5)(0,6)(8,0)(7.5,5.5)
  \psccurve[linewidth=3pt,linecolor=gray]%
        (4,0)(2,4)(6,4)
  \psdots[dotstyle=square]
        (0,0)(0,6)(8,0)(8,6)
  \psdots[dotstyle=pentagon](4,3)
\end{pspicture}

```

Figura 3.1: Risultato dell'esempio dei comandi di **PSTricks**; si noti la semplicità consentono di controllare anche i più piccoli particolari dei disegni.

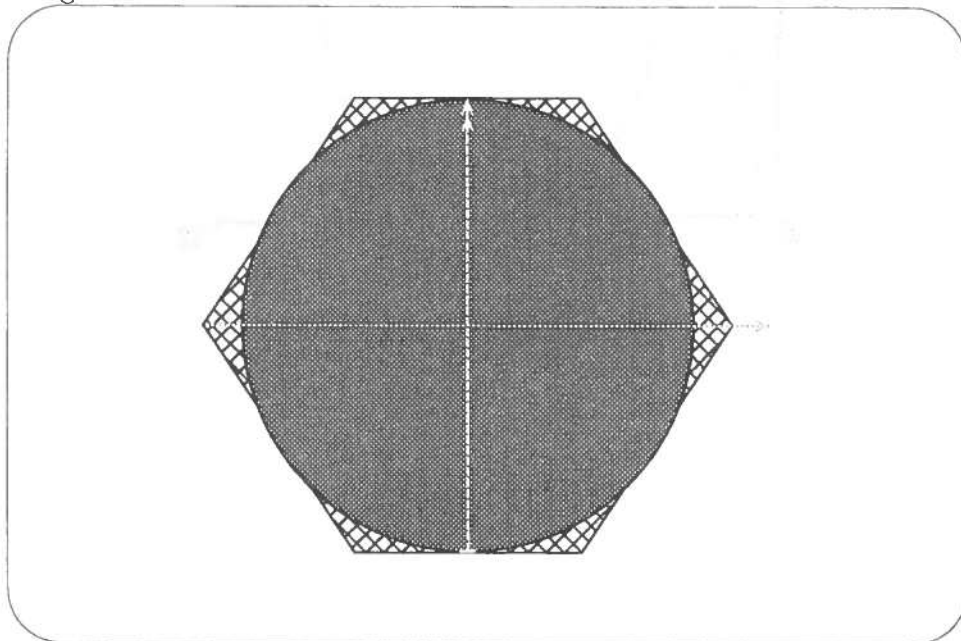
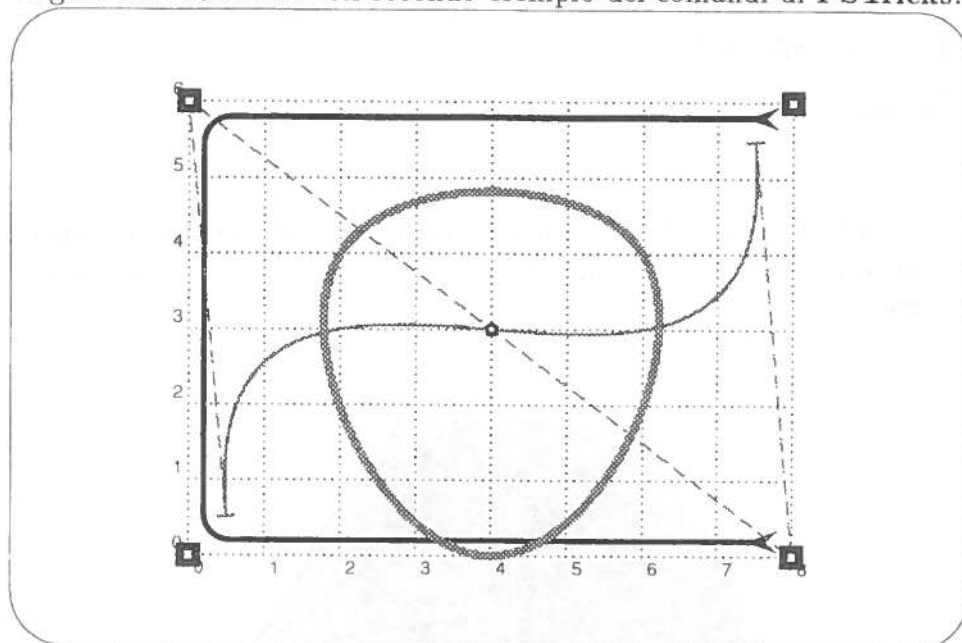


Figura 3.2: Risultato del secondo esempio dei comandi di PSTricks.



Tutti i comandi sono descritti con maggior precisione nella guida di **PSTricks** ma ce ne sono alcuni che meritano una particolare attenzione e di cui si parlerà nella prossima sottosezione.

3.2.1 Alcuni comandi speciali.

Un comando che si presta a svariati utilizzi è

```
\begin{pspicture}[bas](coor1)(coor2)
...
\end{pspicture}
```

simile al comando \LaTeX

```
\begin{picture}(coor1)(coor2) ... \end{picture}
```

Ecco il significato dei vari parametri:

bas è la frazione dell'altezza da allineare con \TeX -*currentpoint* (quindi un numero compreso tra 0 e 1);

(*coor1*)(*coor2*) sono le coordinate degli angoli inferiore sinistro e superiore destro del disegno.

All'interno dell'*environment* si possono usare tutti i comandi grafici cui si è accennato in precedenza, ed in più il comando

```
\rput*[ref]{rot}(x,y){stuff}
```

che consente di posizionare il *LR*-box *stuff* nella posizione (*x,y*), ruotato di un angolo *rot*; con l'eventuale settaggio [*ref*] si può al solito determinare la *reference-point* di *stuff* (che per default è il centro), cosa che torna molto utile anche per associare delle labels alle varie parti del disegno. È possibile così generare semplici (e complessi) disegni, sfruttando i comandi grafici già visti, oppure personalizzare grafici e disegni ottenuti con altri programmi (ad esempio lo *stuff* di un `\rput` può tranquillamente essere un file **.PS**).

Nell'esempio che segue si coglie l'occasione per descrivere in maggior dettaglio due interessanti comandi che possono essere usati all'interno

di un `pspicture`-*environment*, cioè

```
\pscustom[opt.arg.]{graphic commands}
```

e

```
\psclip ... \endpsclip
```

dei quali il primo serve a definire dei propri oggetti grafici ed il secondo ad eseguire un *clipping*. All'interno di `\pscustom` si possono usare i più svariati comandi grafici (fare riferimento al manuale per conoscerli tutti). Ma ecco l'esempio cui si accennava:

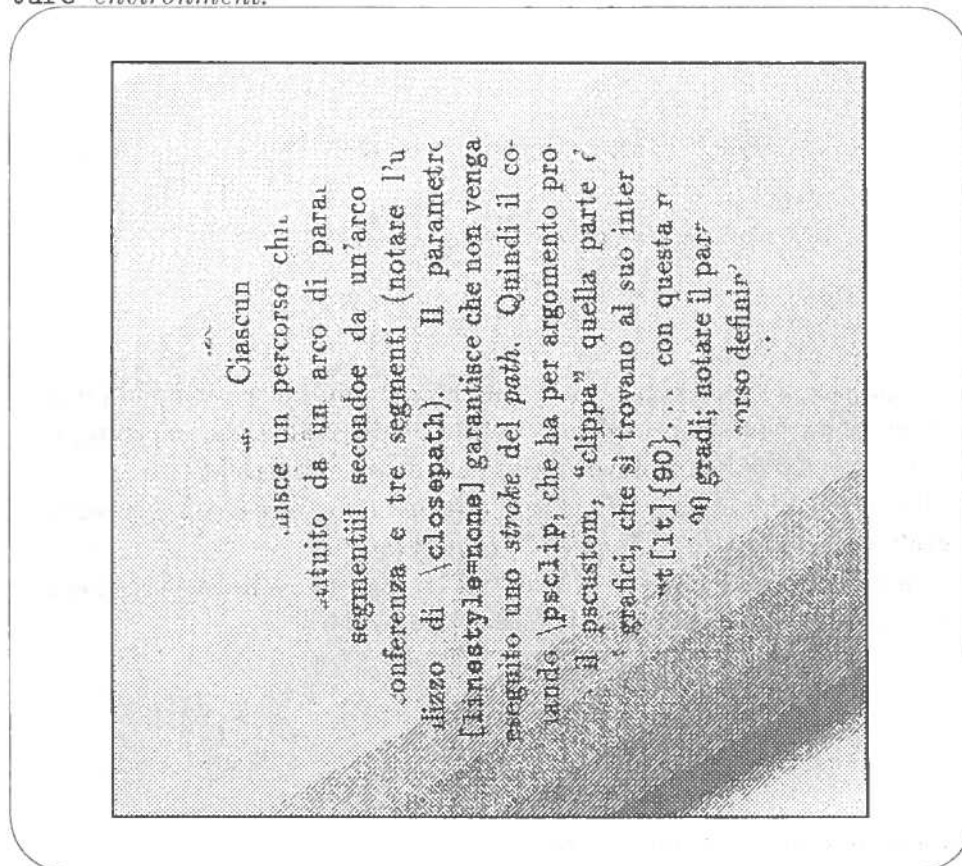
Esempio 9 (PSTricks: clipping)

```
\begin{pspicture}(0,0)(10,10)
\psframe[...](10,10)
\psclip{%
  \pscustom[linestyle=none]{%
    \parabola(1,7)(5,1)
    \rlineto(0,3) \rlineto(-8,0)
    \closepath}
  \pscustom[linestyle=none]{%
    \psarc(5,5){4}{0}{180}
    \rlineto(0,-5) \rlineto(8,0)
    \closepath}}
  \rput[1t]{90}(1,1){\parbox[t]{8cm}{%
    Questo \ 'e un esempio...
    ...
    ...altri effetti speciali.}}
\endpsclip
\end{pspicture}
```

In esso si mostra un utilizzo di `\psclip` insieme a `\pscustom`. Ciascun comando `\pscustom` definisce un percorso chiuso, il primo costituito da un arco di parabola e tre segmenti, il secondo da un'arco di circon-

ferenza e tre segmenti (notare l'utilizzo di `\closepath`). Il parametro `[linestyle=none]` garantisce che non venga eseguito uno *stroke* del *path*. Quindi il comando-*environment* `\psclip`, che ha per argomento proprio il `\pscustom`, "clippa" quella parte di oggetti grafici, che si trovano al suo interno (cioè l'`\rput{90}`... con questa minipage ruotata di 90 gradi; notare il parametro `leftop`) secondo il percorso definito dai due `\pscustom`. Una caratteristica importante di `\psclip` è che se si inseriscono, come argomenti più percorsi chiusi, il *clipping* avviene nell'intersezione, che è proprio il caso di questo esempio. Questi comandi possono essere utilizzati ad esempio per ombreggiare delle regioni dalla forma strana o per altri effetti speciali. Il risultato dell'esempio precedente è nella figura (3.3)

Figura 3.3: Esempio di utilizzo di `\pscustom` e `\psclip` in un *pspicture-environment*.



3.3 PST-PLOT.STY.

Uno degli ulteriori stili aggiunti dal pacchetto è PST-plot che è pensato per consentire di graficare semplici funzioni direttamente sul proprio lavoro, con un minimo di conoscenza del linguaggio PostScript. Alle possibilità offerte da comandi tipo

$$\backslash\text{fileplot}[par]\{file\}$$

che consentono di stampare direttamente i dati presenti in *file* sotto forma di grafico⁴ se ne aggiungono due ulteriori particolarmente interessanti:

$$\backslash\text{psplot}*[par]\{x_{min}\}\{x_{max}\}\{ps-func\}$$

$$\backslash\text{parametricplot}*[par]\{t_{min}\}\{t_{max}\}\{ps-func\}$$

dove x_{min} , x_{max} (risp. t_{min} , t_{max}) sono gli estremi della variabile (risp. parametro) e *ps-func* è il codice **PostScript** del grafico da tracciare, in cui x e t rispettivamente devono comparire come variabile.

Con il primo comando si grafica una funzione reale di x e con il secondo un grafico parametrico in funzione del parametro t .

par consente di selezionare il “tratto” di tracciatura, a linea, punti, ecc. (cioè *curve*, *dots*, *ccurve*...), tramite il parametro

$$\text{plotstyle}$$

Importante è anche il parametro

$$\text{plotpoints}=\textit{num}$$

⁴Sul manuale di **PSTricks** spiegazioni più dettagliate.

che consente di scegliere il numero di punti da tracciare. Ecco un esempio, in cui si vedono applicati anche l'*environment* `pspicture` ed il comando `\psgrid`:

Esempio 10 (PST-plot: parametricplot)

```
\psset{xunit=3cm,yunit=3cm}
\begin{pspicture}(3.25,3)
  \psgrid
  \rput{0}(1.5,1.5){\parametricplot%
                    [plotstyle=ccurve,%
                    plotpoints=99,%
                    linewidth=2pt]%
                    {0}{360}{t sin 5 t mul cos}
  }
  \end{pspicture}
\psset{xunit=1cm,yunit=1cm}
```

Il risultato è quello di figura (3.4).

Tracciare funzioni è utile ma spesso servono anche degli assi cartesiani per completare il grafico. Si può ottenere rapidamente ciò tramite il comando

```
\psaxes[par]{arr}(coor1)(coor2)(coor3)
```

(*coor2*) e (*coor3*) indicano la posizione degli angoli inferiore sinistro e superiore destro del rettangolo che contiene gli assi, mentre (*coor1*) designa la posizione dell'origine (esempio nella figura (3.5)). Anche per questo comando vi sono numerosi parametri:

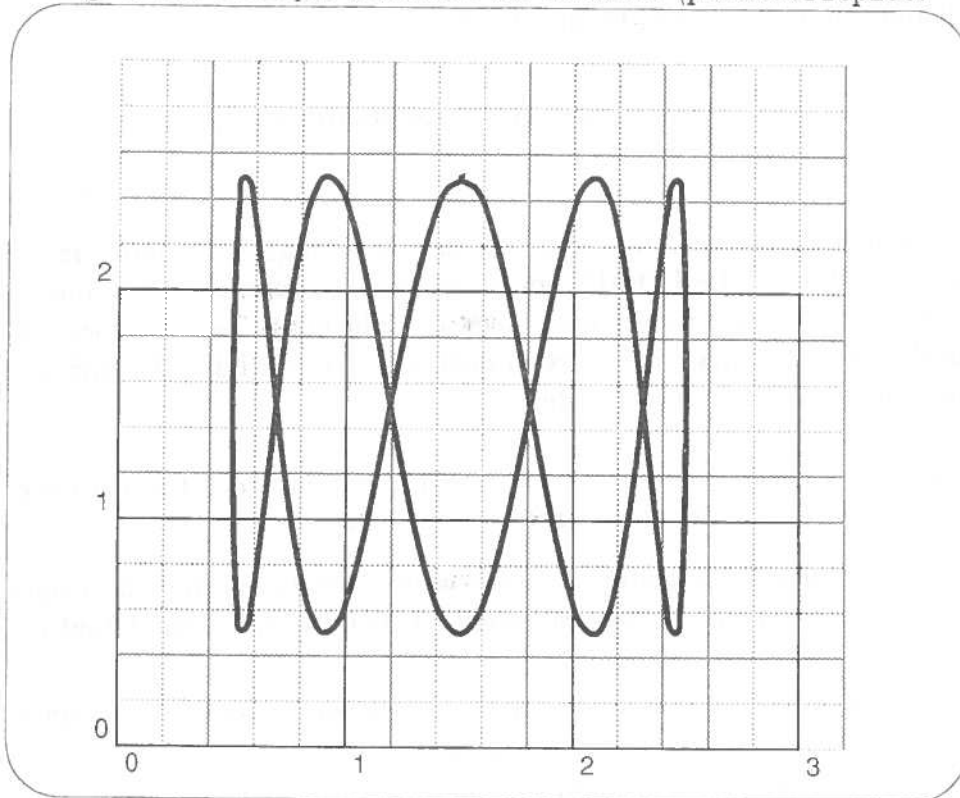
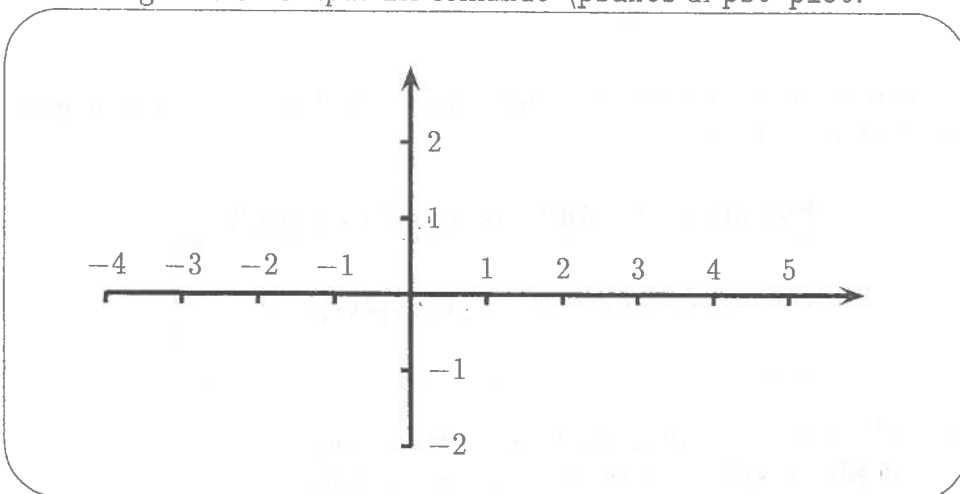
1. `labels=none/x/y/all` consente di scegliere se mettere i numeri;
2. `ticks=none/x/y/all` consente di scegliere se mettere le tacche;
3. `tickstyle=full/top/bottom` consente di scegliere se le tacche devono estendersi dal lato delle label, dal lato opposto o da entrambi.

Ecco un esempio,

Esempio 11 (PST-plot: assi cartesiani)

```
\psaxes[linewidth=1.5pt,  
        tickstyle=top,  
        labels=all]{>}(4,2)(0,0)(10,5)
```

che in fase di output produce il sistema di assi cartesiani di figura (3.5).

Figura 3.4: Esempio di utilizzo del comando `\parametricplot`.Figura 3.5: *Output* del comando `\psaxes` di `pst-plot`.

3.4 MULTIDO.STY.

Questa *style-option* consente di eseguire dei **DO-LOOP** in \LaTeX ! Tramite il nuovo comando introdotto

$$\backslash\text{multido}\{variables\}\{repetitions\}\{stuff\}$$

è possibile creare dei loop, inserendo dei comandi grafici (e non!) in cui le variabili del **DO-LOOP** compaiono come parametri; si può quindi stampare più volte un disegno, magari cambiandone solo il colore o la posizione, o stampare dei *mark* su degli assi, e cose simili. Il significato dei vari parametri è il seguente:

variables è una *control-sequence* in \LaTeX che inizi per `d o n o i o r`, nella forma *valore iniziale + incremento*:

1. `d` stà per distanza e i parametri devono essere delle *length* per \LaTeX ; in fase di *output* si ottiene un valore in *sp* (output significa “*stampa*”);
2. `n` stà per numero e i parametri devono essere numeri con le parti decimali lunghe uguali;
3. `i` stà per intero ed i parametri devono essere numeri interi;
4. `r` stà per reale ed i parametri devono essere numeri reali;

repetitions è il numero di ripetizioni da eseguire;

stuff è la sequenza \LaTeX da ripetere, che può contenere le variabili utilizzate in *variables*.

Per capire meglio il principio di funzionamento di questo comando ecco un esempio molto semplice:

Esempio 12 (Multido: multido semplice)

$$\backslash\text{multido}\{\backslash\text{int}=1+2\}\{10\}\{\backslash\text{int}, \backslash }$$

con l’altrettanto semplice risultato in figura (3.6).

Un pò più complesso è invece l’esempio seguente:

Esempio 13 (Multido: multido evoluto)

```

\begin{pspicture}(-4.5,-4.5)(4.5,4.5)
  \newgray{mygray}{0}
  \psset{fillstyle=solid,fillcolor=mygray}
  \multido{\nm=1.0+-0.15,\dim=4cm+-.7cm}{5}
    {\newgray{mygray}{\nm}
     \rput{0}(0,0){\pscircle(0,0){\dim}}}
\end{pspicture}

```

Osserviamo dapprima che un'istruzione `multido` può anche apparire all'interno di una figura, come nel caso in esame. La prima riga dopo il `\begin{pspicture}`, cioè

```
\newgray{mygray}{0}
```

serve solo a definire il colore `mygray` in modo da non ottenere un errore alla successiva istruzione,

```
\psset{fillstyle=solid,fillcolor=mygray}
```

che lo richiama e serve a fissare i parametri **PostScript** per le successive istruzioni: in particolare si fissa lo stile per il *filling* ed il relativo colore. Si arriva così al `multido`:

Questo ripete per cinque volte il disegno di un cerchio di raggio `\dim` che risulterà *fillato* grazie alla precedente istruzione `\psset` del colore `\mygray`, che viene ridefinito prima di ogni tracciatura dal noto comando `\newgray`. Ad ogni ciclo le variabili `\nm` e `\dim` vengono aumentate del rispettivo incremento (-0.15 e -0.7 cm rispettivamente): si noti che

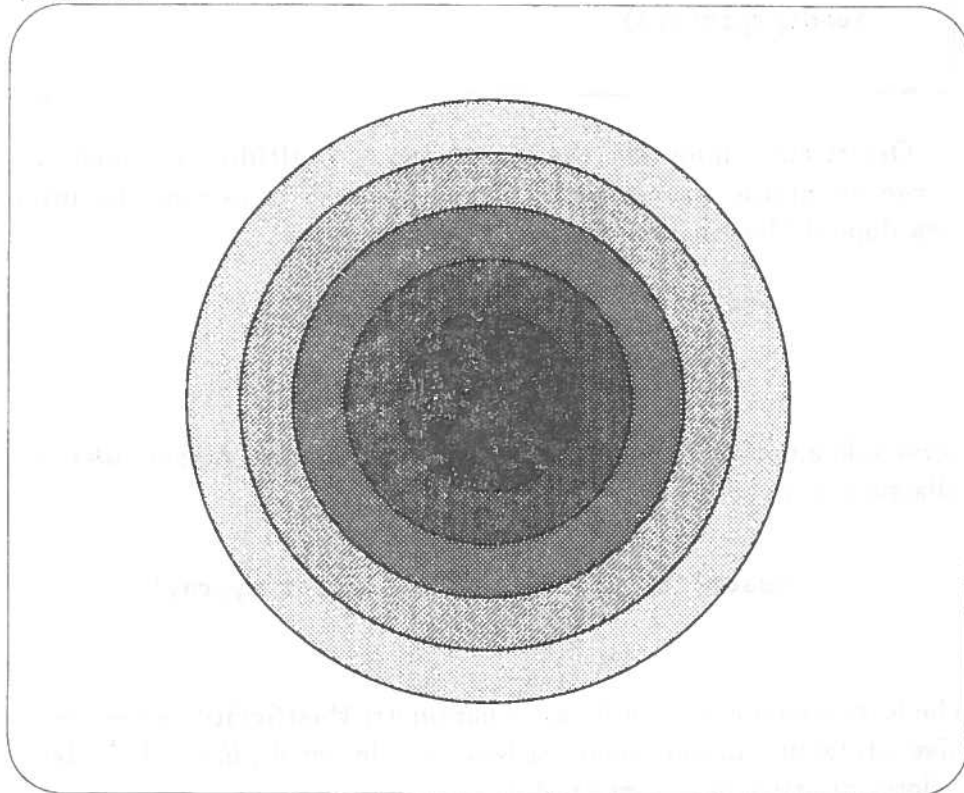
Figura 3.6: Semplice output di un comando `\multido`!

1, 3, 5, 7, 9, 11, 13, 15, 17, 19,

poichè il secondo parametro del comando `\newgray` è un numero, tale è il *tipo* di `\nm`, mentre `\dim` è una distanza, come deve essere il raggio del cerchio.

Il risultato è nella figura (3.7).

Figura 3.7: Risultato di un esempio un pò più complesso sul comando `\multido`.



3.5 PST-NODE.STY & PST-COIL.STY.

Questa interessante *style-option*, PST-NODE.STY, semplifica la stesura di diagrammi di flusso e di tutti i diagrammi in cui si vogliono connettere tramite frecce vari *item* (ad esempio nei diagrammi che spiegano la dipendenza tra i vari capitoli di un libro o anche nei diagrammi matematici, ecc.).

Le istruzioni introdotte si dividono in tre tipi:

1. istruzioni di assegnazione, che consentono di stampare un *box* di grafica e/o testo e di associargli un nome, con il quale in seguito potrà essere richiamato; queste istruzioni si distinguono perchè incorniciano diversamente (o nient'affatto) il *box* o perchè fissano dei diversi *punti di riferimento* per le successive connessioni;
2. istruzioni di connessione, che consentono di connettere tra loro due nodi. Hanno la sintassi

$$\backslash nc \dots \{arr\} \{node1\} \{node2\}$$

ed, ovviamente, connettono il nodo *node1* con quello *node2* in vari modi (curve spezzate, ecc.); al solito vi sono dei parametri che consentono di personalizzare le varie caratteristiche, spessore, lunghezza dei vari tratti, curvatura, ecc.;

3. istruzioni di *labeling*, per assegnare dei nomi alle connessioni; in generale una di queste istruzioni si presenta come

$$\backslash \dots put [ref] \{rot\} (pos) \{lab\}$$

con il seguente significato dei simboli:

ref è il punto di riferimento della label che va posto alle coordinate indicate;

rot è l'angolo di rotazione;

pos è la posizione in coordinate;

lab è la label;

le istruzioni che iniziano con lettera maiuscola usano il default per i primi tre parametri, così come `\mput`.

Si elencano di seguito i vari comandi dividendoli nelle categorie appena menzionate.

Definizione di un nodo: assegnazione di un nome e posizionamento di un oggetto (grafico e non):

1. `\rnode`, `\Rnode`
2. `\pnode`
3. `\cnode`, `\circlenode`, `\cnodeput`
4. `\ovalnode`

Connessioni tra nodi : i nodi vengono richiamati con il nome usato nel comando di *assegnazione*:

1. `\ncLine`, `\ncbar`, `\nc`
2. `\nccurve`, `\nccircle`, `\ncloop`
3. `\ncangle`, `\ncangles`, `\ncdiag` `\ncdiagg`
4. `\nccoil`, `\nczigzag`

“labels” di nodi: labeling delle connessioni:

1. `\put`, `\aput`, `\bput`
2. `\mput`, `\Aput`, `\Bput`

Si rimanda al manuale di **PSTricks** per ulteriori informazioni sull'utilizzo specifico dei vari comandi. Dall'esempio che segue si può però già imparare il funzionamento dei vari comandi e di alcuni loro parametri:

Esempio 14 (PST-node: diagramma di commutazione)

```

$$
\begin{array}{cccccc}
\Rnode{a}{X} & & \Rnode{b}{Y} & & \QQ & & \\
\Rnode{c}{\tilde{Y}} & & & & \QQ & & \\
\Rnode{d}{\tilde{X}} & & & & & & \\
\|[3cm] & & & & & & \\
\Rnode{aa}{X/\ker(\psi)} & & & & & & \\
\Rnode{dd}{\tilde{x}/\ker(\tilde{\psi})} & & & & & & \\
\|[2cm] & & & & & & \\
\end{array}
\psset{nodesep=5pt} \everypsbox{\scriptstyle}
\ncLine{->}{a}{b}\Aput{\psi}
\ncLine{<<-}{c}{d}\Aput{\tilde{\psi}}
\ncLine{->}{a}{aa}\Bput{\phi}
\ncLine{->}{d}{dd}\Aput{\tilde{\phi}}
\ncLine[linestyle=dashed,dash=2pt 2pt]{%
{<->}{aa}{b}\Aput{\chi}\Bput{\sim}
\ncLine[linestyle=dashed,dash=2pt 2pt]{%
{<->}{dd}{c}\Aput{\tilde{\chi}}\Bput{\sim}
\ncLine[linestyle=dotted]{%
{<->}{aa}{dd}\Aput{\sigma}\Bput{\sim}
\ncLine[linestyle=dotted]{%
{<->}{b}{c}\Aput{\tau}\Bput{\sim}
$$

```

Si osservi come i *boxes* che rappresentano i nodi possano essere inseriti in normali istruzioni \LaTeX !!! Il risultato è un diagramma di commutazione, riportato in figura (3.8).

A differenza di **PST-NODE**, **PST-COIL** non è una *style-option* molto sofisticata.

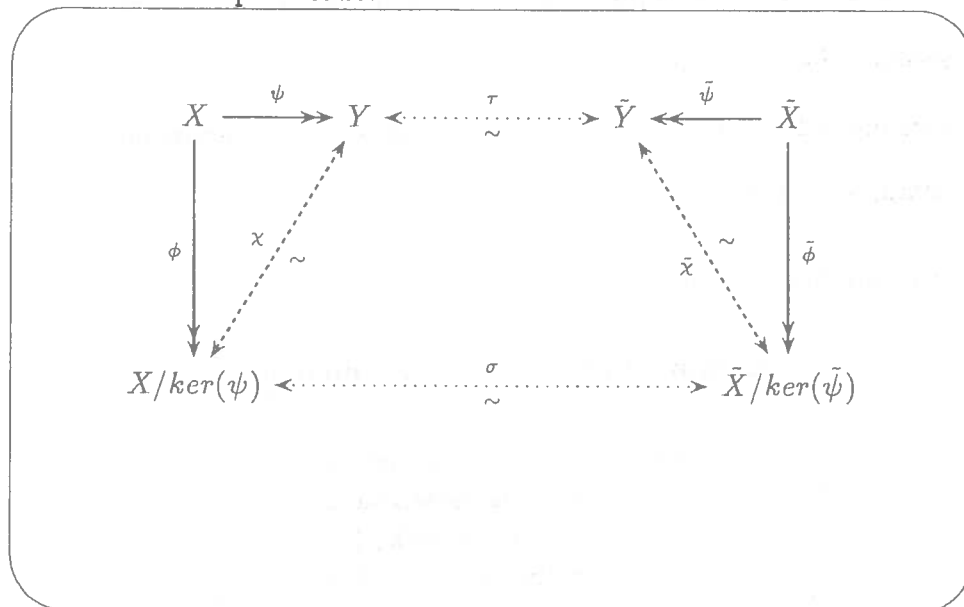
Aggiunge i comandi

- `\pscoil`
- `\psCoil`
- `\pszigzag`

che consentono di tracciare particolari linee a forma di *molla* e *zig-zag*; come al solito un certo numero di parametri permette di dimensionare

nella maniera preferita gli oggetti (per ulteriori informazioni si può consultare la guida di **PSTricks**).

Figura 3.8: Diagramma di commutazione ottenuto sfruttando le possibilità offerte da `pst-node`.



3.6 GRADIENT.STY.

La *style-option* `GRADIENT.STY` definisce `gradient`, un ulteriore qualificatore (cioè valore che può assumere un parametro) da utilizzare con il parametro grafico `fillstyle` per ottenere un nuovo tipo di *filling*, sfumato. Utilizzando questa opzione si può usare, in tutti i comandi grafici che ammettono *filling*, lo stile

```
fillstyle=gradient
```

Ulteriori parametri introdotti sono:

`gradbegin` colore con cui deve iniziare e terminare la sfumatura;

`gradend` colore al *midpoint*;

`gradmidpoint` è la frazione dell'altezza a cui si pone il *midpoint*;

`gradangle` è l'angolo di *filling*.

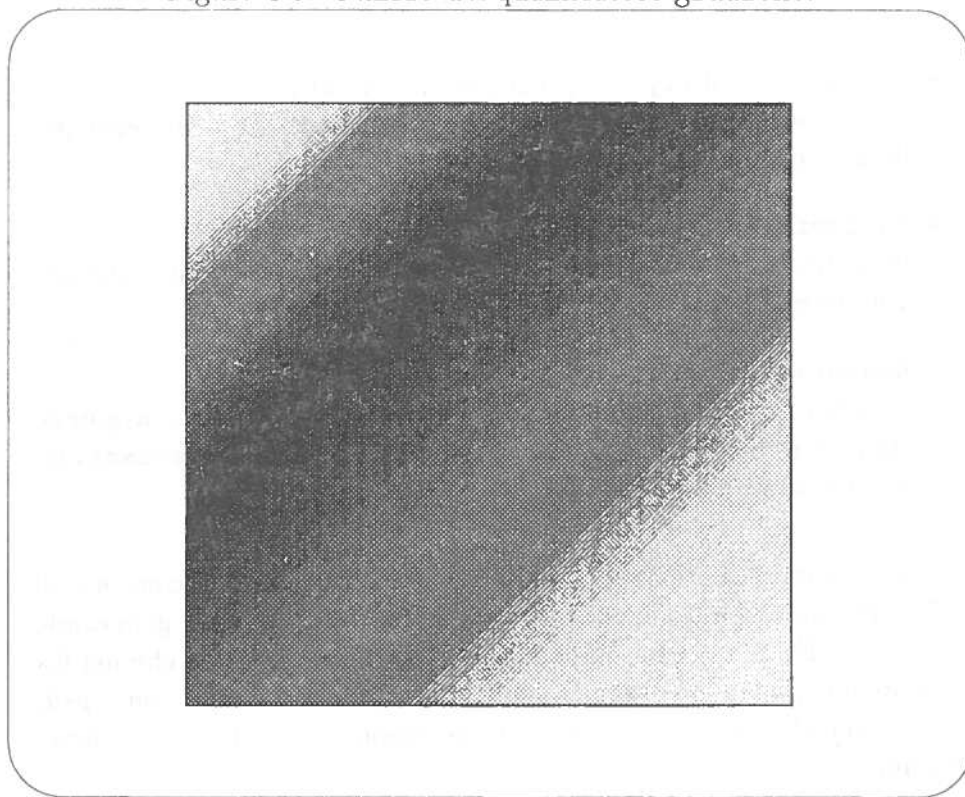
Un semplice esempio è il seguente

Esempio 15 (Gradient: gradient)

```
\psframe[fillstyle=gradient,%  
  gradbegin=white,%  
  gradend=black,%  
  gradmidpoint=.333,%  
  gradangle=45](10,10)
```

con il risultato in figura (3.9).

Figura 3.9: Utilizzo del qualificatore gradient.



3.7 PST2EPS.STY.

Interessante è il comando

$$\backslash\text{PSTtoEPS}[par]\{file\}\{graph\}$$

che consente di creare un file **.EPS** dai comandi presenti nel suo argomento *graph*. Tale file verrà chiamato con il nome *file* e potrà essere inserito nello stesso documento, anche a partire dalla riga successiva. Vediamo più in dettaglio quali sono i parametri consentiti:

- `bbllx=...`, `bbly=...`, `bburx=...`, `bbury=...`
consentono di fissare il **BoundingBox** che altrimenti per default è $(-1, -1), (1, 1)$;
- `headerfile=...`
fissa eventuali files di *header* particolari, che si possono voler aggiungere al codice **PostScript**;
- `headers=none/all/user`
seleziona i files di *header* da inserire: in particolare `all` aggiunge sia i file di header di **PSTricks** che quelli definiti da `headerfile`, mentre `user` solo quelli definiti da `headerfile`.

Questo comando è particolarmente interessante perchè consente di spedire un nostro lavoro anche a chi non possiede **PSTricks**, generando prima i files **EPS** e poi modificando il programma in modo che nel file che si spedirà, vengano eseguiti degli input, grazie a macro come *psfig* o *epsfig*, dei files creati, al posto dell'esecuzione del codice in comandi **PSTricks**.

È importante in tal caso ricordarsi di aggiungere il **BoundingBox** tra i parametri⁵, e di usare il parametro

`headers=all`

⁵Nel caso si inserisca il file **.EPS** in un comando `\rput` invece si può tralasciare di specificarlo; questo può essere utile quando si deve utilizzare più volte uno stesso grafico che però impiega molto tempo per essere tracciato; allora ciò che conviene fare è sfruttare il comando descritto in questa sezione per ottenerne una copia come file **.EPS**; questo poi verrà incluso, ogniqualvolta servirà, senza dover riprocessare il grafico!

in modo che i files **EPS** contengano tutte le intestazioni necessarie per non generare errori in fase di stampa.

3.8 CHARPATH.STY, TEXTPATH.STY, COLORTAB.STY.

Questi stili consentono rispettivamente di:

- fillare un certo testo con il *pattern* scelto tramite i parametri `fillstyle`, `fillcolor`;
- scrivere un testo lungo una curva o qualsiasi altro percorso definito da comandi grafici di **PSTricks**;
- ottenere tabelle a colori.

Non si descrivono ulteriormente, perchè dalla stessa guida di **PSTricks** sono definiti meno **robusti**, cioè non completamente affidabili!

Appendice A

Installare Seminar.

L'installazione di **Seminar** non presenta particolari problemi. È sufficiente mettere i files della distribuzione¹ in una delle directory definite dal *nome logico* `TEX_INPUTS`, meglio se creandone una appositamente. Ad esempio se il path della directory di input per \LaTeX è `TEX_ROOT:[INPUTS]`, si può creare una *subdirectory*

`TEX_ROOT:[INPUTS.SEMINAR]`

in cui mettere i vari files, aggiungendola poi nel nome logico

`TEX_INPUTS`

Si riporta di seguito un elenco dei files presenti nella distribuzione di **Seminar**:

- 2up.doc
- 2up.sty
- 2up.tex
- npsfont.sty
- sem-a4.sty
- sem-page.sty
- sem-read.me
- sem-user.ind
- sem-user.tex
- semcolor.sty
- semhelv.sty
- seminar.sty

¹Questi sono *public-domain* e reperibili tramite *anonymous-ftp* dal nodo `PRINCETON.EDU` directory archives/pub/tvz.

- `semlayer.sty`
- `semlcms.sty`
- `semrot.sty`
- `semsamp1.tex`
- `semsamp2.tex`
- `slidesec.sty`
- `tvz-code.sty`
- `tvz-hax.sty`
- `tvz-user.sty`
- `xcomment.sty`

a cui aggiungiamo quelli della distribuzione di **Fancybox**

- `fancybox.doc`
- `fancybox.sty`
- `fancybox.tex`

Appendice B

Installare PSTricks.

L'installazione di **PSTricks** è leggermente più complicata di quella di **Seminar**. E' consigliabile creare un albero di subdirectory dedicato ai files di **PSTricks**. Un file di documentazione per l'installazione è presente nella distribuzione di **PSTricks**: ecco in sintesi le operazioni da eseguire se si è in possess del *tool* **Rokicki's DVIPS** per convertire i files da *DVI* a *PS*:

1. inserire tutti i files di *input*¹, al solito, in una delle directory di **TEX_INPUTS**;
2. inserire gli *headers* per la stampante **PostScript**² nella directory ove essi vengono cercati da *DVIPS*; alternativamente si possono mettere altrove, a patto di modificare nel file *pstricks.con* la

¹Questi sono:

- | | | |
|----------------|----------------|----------------|
| • pstricks.con | • pst-coil.tex | • textpath.sty |
| • pstricks.tex | • pst-coil.sty | • pst2eps.tex |
| • pstricks.sty | • gradient.tex | • pst2eps.sty |
| • pst-node.tex | • gradient.sty | • multido.tex |
| • pst-node.sty | • charpath.tex | • multido.sty |
| • pst-plot.tex | • charpath.sty | • colortab.tex |
| • pst-plot.sty | • textpath.tex | • colortab.sty |

²Questi sono:

- | | | | |
|----------------|----------------|----------------|----------------|
| • pstricks.pro | • pst-coil.pro | • gradient.pro | • textpath.pro |
| • pst-node.pro | | | |

definizione di `\pstheader` in modo che “punti” alla directory in cui si trovano gli *headers*.

Si rammenta comunque che **PSTricks** può essere usato anche con altri convertitori da *DVI* a *PS*, perchè il file `pstricks.con` è composto da tante parti, ciascuna delle quali è il file di configurazione per un convertitore specifico: tutte queste parti, tranne quella relativa a **Rokicki's DVIPS**, sono “commentate”³ e non vengono quindi processate. Se si possiede un altro compilatore, bisogna allora commentare la parte relativa a **Rokicki's DVIPS** e togliere il “commento” alla sezione relativa al convertitore che interessa.

³carattere % all'inizio della riga!

Indice

— Simboli —

*, 34

— A —

allversions*, 20, 22
appunti, 19
aput, 48-49
article, 19
articlemag, 20, 23

— B —

bblx, 54
bbly, 54
bburx, 54
bbury, 54
bput, 48-49

— C —

centratura
 verticale, 8
charpath, 55
circlenode, 48
closepath, 38
cnode, 48
cnodeput, 48
colori, 16, 32
colorlayers, 16
colortab, 55
comandi

Multido

 multido, 44, 45

PST-coil

 pscoil, 49
 pszigzag, 49

PST-node

 aput, 48-49
 bput, 48-49
 circlenode, 48
 cnode, 48
 cnodeput, 48
 mput, 48
 nc, 48
 ncangle, 48
 ncangles, 48
 ncbar, 48
 nccircle, 48
 nccoil, 48
 nccurve, 48
 ncdiag, 48
 ncdiagg, 48
 ncLine, 48-49
 ncloop, 48
 nczigzag, 48
 ovalnode, 48
 pnode, 48
 put, 48
 rnode, 48-49

PST-plot

 fileplot, 40
 parametricplot, 40, 41
 psaxes, 41, 42
 psplot, 40

PSTricks, 32

 closepath, 38

degrees, 32
 endpsclip, 32
 grid, 32
 newgray, 45
 parabola, 32, 38
 psarc, 32, 38
 psbezier, 32, 34
 pscurve, 32, 34
 pscircle, 32, **33**, 34, 45
 psclip, 32
 pscurve, 32, **33**
 pscustom, 32, **38**
 psdot, **33**
 psdots, 32, 34
 psellipse, 32
 psframe, 32, **33**, 38, 52
 psgrid, **33**, 34
 psline, 32, **33**, 34
 pspolygon, 32, **33**, 34
 psset, 32-34, 41, 45
 pswedge, 32, **33**
 rlineto, 38
 rput, **37**, 38, 45
 Seminar
 articlemag, 20
 colorlayers, 16
 landscapeonly, **27**
 newpagestyle, **11**
 noxcomment, 20-21
 overlay, **16**, 17-18
 pagestyle, **10**
 portraitonly, **28**
 ptsize, 20
 rotateheaderstrue, **11**
 slideframe, **9**
 slideplacement, 20
 slidesmag, **11**, 20

— D —

dash, 34
 degrees, 32

documentazione
 PSTricks, 30-31
 seminar, 5, 12
 documentstyle, 6-7, 9, 17,
 19-21, 28
 dotscale, 34
 dotstyle, 34
 DVIPS, 2, 7, 59

— E —

endpsclip, 32
 environments
 PSTricks
 psclip, **38**
 pspicture, **37**, 38, 41, 45
 Seminar
 allversions*, 20, **22**
 notes, 21, **22**
 overlay, **16**
 slide, **6**
 slide*, 7
 slide, 7
 slide*, 7
 epsfig, 2
 Esempi
 Gradient
 gradient, 52
 Multido
 multido evoluto, 45
 multido semplice, 44
 PST-node
 diagramma di
 commutazione, 49
 PST-plot
 assi cartesiani, 42
 parametricplot, 41
 PSTricks
 clipping, 38
 grafica II°, 34
 grafica I°, 34
 Seminar

appunti II°, 21
 appunti I°, 20
 cornici ed intestazioni, 9
 lucidi elementari, 7
 main file, 24
 overlay, 17

— F —

Fancybox, 3, 6, 9
 fileplot, 40
 fillcolor, 34, 45
 fillstyle, 34, 45, 52
 frecce, 34

— G —

gradangle, 52
 gradbegin, 52
 gradend, 52
 gradient, 52
 gradmidpoint, 52
 grid, 32
 griddots, 34
 gridlabels, 34
 GV, 2

— H —

headers, 54-55

— I —

incorniciatura, 8-9, 14-15
 installazione
 PSTricks, 59
 Seminar, 57
 intestazione, 8-9, 14-15

— L —

labels, 41, 42
 landscape, 7, 27
 landscapeonly, 27
 linearc, 34
 linecolor, 34
 linestyle, 34, 38
 linewidth, 34; 41-42

— M —

mput, 48
 multido, 44, 45

— N —

nc, 48
 ncangle, 48
 ncangles, 48
 ncbar, 48
 nccircle, 48
 nccoil, 48
 nccurve, 48
 ncdiag, 48
 ncdiagg, 48
 ncLine, 48-49
 ncloop, 48
 nczigzag, 48
 newgray, 45
 newpagestyle, 9, 11
 note, 19
 notes, 19, 20, 22
 notesonly, 19
 notesonly*, 19
 noxcomment, 20-21

— O —

ovalnode, 48
 overlay, 16, 17-18

overlays, 16

— P —

pagestyle, 9, 10, 11

parabola, 32, 38

parametri grafici

Gradient

fillstyle=**gradient**, 52

gradangle, 52

gradbegin, 52

gradend, 52

gradmidpoint, 52

PST-plot

labels, 41, 42

linewidth, 41-42

plotpoints, 40, 41

plotstyle, 40, 41

ticks, 41

tickstyle, 41, 42

PST2EPS

bblx, 54

bbly, 54

bburx, 54

bbury, 54

headerfile, 54

headers, 54-55

PSTricks

*****, 34

dash, 34

dotscale, 34

dotstyle, 34

fillcolor, 34, 45

fillstyle, 34, 45, 52

frecce, 34

griddots, 34

gridlabels, 34

linearc, 34

linecolor, 34

linestyle, 34, 38

linewidth, 34

runit, 32

showpoints, 34

subgriddiv, 34

xunit, 32, 41

yunit, 32, 41

parametricplot, 40, 41

plotpoints, 40, 41

plotstyle, 40, 41

pnode, 48

portrait, 7, 19, 28

portraitonly, 28

processare

landscape, 7, 27

portrait, 7, 28

psarc, 32, 38

psaxes, 41, 42

psbezier, 32, 34

psscurve, 32, 34

pscircle, 32, **33**, 34, 45

psclip, 32, **38**

psoil, 49

pscurve, 32, **33**

pscustom, 32, **38**

psdot, **33**

psdots, 32, 34

psellipse, 32

psfig, 2

psframe, 32, **33**, 38, 52

psgrid, **33**, 34

psline, 32, **33**, 34

pspicture, **37**, 38, 41, 45

psplot, 40

pspolygon, 32, **33**, 34

psset, 32-34, 41, 45

pst-coil, 49

pst-node, 47

pst-plot, 40

pst2eps, 54

pstricks, 32

pswedge, 32, **33**

pszigzag, 49

ptsize, 20, 23

put, 48

— R —

rlineto, 38
 rnode, 48–49
 rotateheaderstrue, 9, 11
 rput, 37, 38, 45
 runit, 32

— S —

semcmss, 6
 semcolayer, 16
 semcolor, 5, 10, 16, 17–18
 semhelv, 6
 seminar, 6
 semlayer, 5, 17–18
 showpoints, 34
 slide, 7
 slide*, 7
 slideframe, 9, 10
 slideplacement, 20
 options, 23
 slidesec, 5
 slidesmag, 11, 20, 23
 slidesonly, 19, 22
 slidesplacement, 23
 stampa
 landscape, 7
 stampare
 landscape, 27
 portrait, 28
 style-options
 fancybox, 6, 9
 PSTricks, 30–32
 charpath, 55
 colortab, 55
 gradient, 52
 Multido, 44
 pst-coil, 49
 pst-node, 47
 pst-plot, 40
 pst2eps, 54

textpath, 55
 Seminar, 5, 12
 article, 19
 notes, 19, 20
 notesonly, 19
 notesonly*, 19
 portrait, 7, 19, 28
 semcmss, 6
 semcolor, 5, 10, 16
 semhelv, 6
 semlayer, 5, 16
 semrot, 5
 slidesec, 5
 slidesonly, 19, 22
 xcomment, 5, 19
 subgriddiv, 34

— T —

textpath, 55
 ticks, 41
 tickstyle, 41, 42

— X —

xcomment, 5, 19–20
 xdvi, 2
 xunit, 32, 41

— Y —

yunit, 32, 41

