

# Introduzione a L<sup>A</sup>T<sub>E</sub>X

## Quarta lezione

Pietro Donatis

Liceo Linguistico Statle "Ninni Cassarà"

28 febbraio 2017

# Argomenti della lezione

- 1 Fisica
  - Unità di misura
- 2 Chimica
- 3 Figure
  - Il pacchetto pstricks
  - Estensioni di pstricks
  - Il pacchetto TikZ

# Fisica

## Unità di misura

# Il pacchetto `siunitx`

Consente di utilizzare in modo molto semplice le unità di misura del Sistema internazionale.

## Il pacchetto `siunitx`

Consente di utilizzare in modo molto semplice le unità di misura del Sistema internazionale.

```
\alpha=\ang{30;45;22}  
m=\SI{5.9742e24}{\kilo\gram}  
c=\SI{299792458}{\metre\per\second}  
v=\SI[per=slash]{4.2}{\metre\per\second}
```

## Il pacchetto siunitx

Consente di utilizzare in modo molto semplice le unità di misura del Sistema internazionale.

```
\alpha=\ang{30;45;22}  
m=\SI{5.9742e24}{\kilo\gram}  
c=\SI{299792458}{\metre\per\second}  
v=\SI[per=slash]{4.2}{\metre\per\second}
```

produce

## Il pacchetto siunitx

Consente di utilizzare in modo molto semplice le unità di misura del Sistema internazionale.

```
\alpha=\ang{30;45;22}  
m=\SI{5.9742e24}{\kilo\gram}  
c=\SI{299792458}{\metre\per\second}  
v=\SI[per=slash]{4.2}{\metre\per\second}
```

produce

$$\alpha = 30^\circ 45' 22''$$
$$m = 5.9742 \times 10^{24} \text{ kg}$$
$$c = 299\,792\,458 \text{ m s}^{-1}$$
$$v = 4.2 \text{ m/s}$$

# Chimica



# Formule e reazioni

Si può usare il pacchetto `mhchem`

# Formule e reazioni

Si può usare il pacchetto `mhchem`

```
\ce{1/2H2O}\\  
\ce{^{227}_{90}Th+}  
\ce{C6H5\bond{1}CHO}  
\ce{CH2\bond{2}CH2}  
\ce{CH\bond{3}CH}  
\ce{CO2 + C <-> 2CO}  
\ce{SO4^2 + Ba^2+ -> BaSO4 v}
```

# Formule e reazioni

Si può usare il pacchetto `mhchem`

```
\ce{1/2H2O}\
```

```
\ce{^{227}_{90}Th+}
```

```
\ce{C6H5\bond{1}CHO}
```

```
\ce{CH2\bond{2}CH2}
```

```
\ce{CH\bond{3}CH}
```

```
\ce{CO2 + C <-> 2CO}
```

```
\ce{SO4^2 + Ba^2+ -> BaSO4 v}
```

$$\frac{1}{2} \text{H}_2\text{O}$$

$${}^{227}_{90}\text{Th}^+$$

$$\text{C}_6\text{H}_5\text{-CHO}$$

$$\text{CH}_2=\text{CH}_2$$

$$\text{CH}\equiv\text{CH}$$

$$\text{CO}_2 + \text{C} \longleftrightarrow 2 \text{CO}$$

$$\text{SO}_4^{2-} + \text{Ba}^{2+} \longrightarrow \text{BaSO}_4 \downarrow$$

# Strutture molecolari

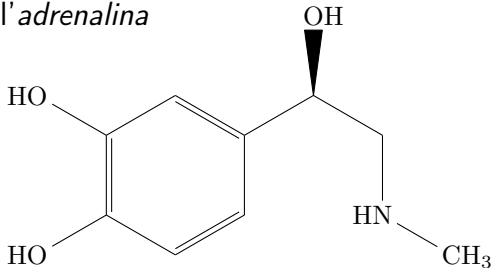
Si usa il pacchetto `chemfig`

# Strutture molecolari

Si usa il pacchetto `chemfig`  
La molecola dell'*adrenalina*

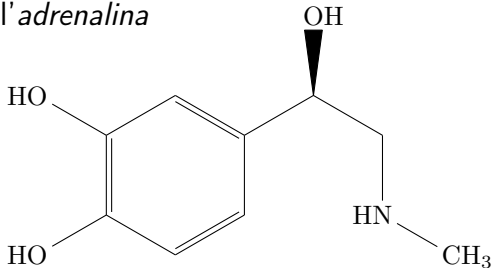
# Strutture molecolari

Si usa il pacchetto `chemfig`  
La molecola dell'*adrenalina*



# Strutture molecolari

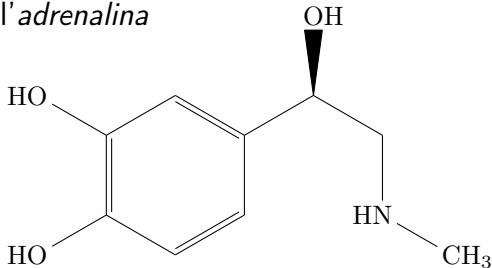
Si usa il pacchetto `chemfig`  
La molecola dell'*adrenalina*



si ottiene con il comando

# Strutture molecolari

Si usa il pacchetto `chemfig`  
La molecola dell'*adrenalina*



si ottiene con il comando

```
\chemfig{*6((-HO)--(-(<[:60]OH)-[:-60]-[:-60,,2]
HN-[:+60]CH_3)--(-HO)=)}
```



# Figure

## Il pacchetto pstricks

# Le basi

È un pacchetto che permette di costruire figure in formato `.ps` da importare nel documento principale.

# Le basi

È un pacchetto che permette di costruire figure in formato `.ps` da importare nel documento principale.

Le istruzioni per la figura da produrre stanno nell'ambiente `pspicture`, il cui utilizzo è

```
\begin{pspicture}(x_0,y_0)(x_1,y_1)
  :
\end{pspicture}
```

## Le basi

È un pacchetto che permette di costruire figure in formato `.ps` da importare nel documento principale.

Le istruzioni per la figura da produrre stanno nell'ambiente `pspicture`, il cui utilizzo è

```
\begin{pspicture}(x_0, y_0)(x_1, y_1)
  :
\end{pspicture}
```

Le coordinate stabiliscono le coordinate dell'angolo inferiore sinistro e superiore destro.

## Le basi

È un pacchetto che permette di costruire figure in formato `.ps` da importare nel documento principale.

Le istruzioni per la figura da produrre stanno nell'ambiente `pspicture`, il cui utilizzo è

```
\begin{pspicture}(x_0,y_0)(x_1,y_1)
  :
\end{pspicture}
```

Le coordinate stabiliscono le coordinate dell'angolo inferiore sinistro e superiore destro.

L'ambiente `pspicture*` taglia la parte di figura che esce dal riquadro definito.

# Inserire oggetti

Un possibile comando è

```
\uput{labelsep}[refangle]{rotation}(x_0, y_0){oggetto}
```

# Inserire oggetti

Un possibile comando è

```
\uput{labelsep}[refangle]{rotation}(x_0,y_0){oggetto}
```

che pone l'oggetto a distanza `labelsep` dal punto di coordinate  $(x_0, y_0)$ , in direzione individuata dall'angolo `refangle`, ruotato dell'angolo `rotation`.

# Tracciare linee

Si usa il comando

```
\psline[par]{arrows}(x_0, y_0)(x_1, y_1)
```



# Tracciare linee

Si usa il comando

```
\psline[par]{arrows}(x_0, y_0)(x_1, y_1)
```

che traccia una linea, eventualmente dotata di frecce, dal punto  $(x_0, y_0)$  al punto  $(x_1, y_1)$ .

# Tracciare linee

Si usa il comando

```
\psline[par]{arrows}(x_0, y_0)(x_1, y_1)
```

che traccia una linea, eventualmente dotata di frecce, dal punto  $(x_0, y_0)$  al punto  $(x_1, y_1)$ .

Le frecce possono essere da un solo dei due estremi o da entrambi con le opzioni

# Tracciare linee

Si usa il comando

```
\psline[par]{arrows}(x_0,y_0)(x_1,y_1)
```

che traccia una linea, eventualmente dotata di frecce, dal punto  $(x_0, y_0)$  al punto  $(x_1, y_1)$ .

Le frecce possono essere da un solo dei due estremi o da entrambi con le opzioni

->	<-	<->	>-<	«->	»-<
-	<->	[-]	(-)	o-o	*_*

# Tracciare linee

I parametri `par` definiscono il tipo di linea, il suo spessore, il suo colore, ... con le opzioni

`linestyle`: `none`, `solid`, `dashed`, `dotted`;

# Tracciare linee

I parametri `par` definiscono il tipo di linea, il suo spessore, il suo colore, ... con le opzioni

`linestyle`: `none`, `solid`, `dashed`, `dotted`;

`linewidth`: si indica la dimensione dello spessore, predefinito `8pt`;

# Tracciare linee

I parametri `par` definiscono il tipo di linea, il suo spessore, il suo colore, ... con le opzioni

`linestyle`: `none`, `solid`, `dashed`, `dotted`;

`linewidth`: si indica la dimensione dello spessore, predefinito `8pt`;

`linecolor`: il valore predefinito è `black`.

# Tracciare linee

I parametri `par` definiscono il tipo di linea, il suo spessore, il suo colore, ... con le opzioni

`linestyle`: `none`, `solid`, `dashed`, `dotted`;

`linewidth`: si indica la dimensione dello spessore, predefinito `8pt`;

`linecolor`: il valore predefinito è `black`.

Per esempio, il comando

```
\psline[linestyle=dashed,linewidth=0.4pt,  
linecolor=red]{->}(0,0)(2,1)(3,3)
```

# Tracciare linee

I parametri `par` definiscono il tipo di linea, il suo spessore, il suo colore, ... con le opzioni

`linestyle`: `none`, `solid`, `dashed`, `dotted`;

`linewidth`: si indica la dimensione dello spessore, predefinito `8pt`;

`linecolor`: il valore predefinito è `black`.

Per esempio, il comando

```
\psline[linestyle=dashed,linewidth=0.4pt,  
        linecolor=red]{->}(0,0)(2,1)(3,3)
```

genera una linea tratteggiata, spezzata, dello spessore di `0.4pt`, di colore rosso e che termina con una freccia.



# Poligoni

Il comando

```
\pspolygon[par](x_0,y_0)\cdots(x_n,y_n)
```

disegna un poligono di vertici  $(x_0, y_0), \dots, (x_n, y_n)$ .

# Poligoni

Il comando

```
\pspolygon[par](x_0, y_0) \cdots (x_n, y_n)
```

disegna un poligono di vertici  $(x_0, y_0), \dots, (x_n, y_n)$ .

Il comando

```
\psframe[par](x_0, y_0)(x_1, y_1)
```

disegna un rettangolo di vertici opposti  $(x_0, y_0)$  e  $(x_1, y_1)$ .

# Poligoni

Il comando

```
\pspolygon[par](x_0, y_0) \cdots (x_n, y_n)
```

disegna un poligono di vertici  $(x_0, y_0), \dots, (x_n, y_n)$ .

Il comando

```
\psframe[par](x_0, y_0)(x_1, y_1)
```

disegna un rettangolo di vertici opposti  $(x_0, y_0)$  e  $(x_1, y_1)$ .

Il comando

```
\diamond[par](x_0, y_0)(x_1, y_1)
```

disegna un rombo centrato in  $(x_0, y_0)$  e di semidiagonali di ampiezza  $x_1$  e  $y_1$ .

# Poligoni

Una regione delimitata può essere riempita con i parametri

`fillstyle=stile`

`fillcolor=colore`

Gli oggetti grafici hanno una versione asteriscata che dà un solido pieno con colore `linecolor`

# Poligoni

Una regione delimitata può essere riempita con i parametri

`fillstyle=stile`

`fillcolor=colore`

Gli oggetti grafici hanno una versione asteriscata che dà un solido pieno con colore `linecolor`

Il colore predefinito è nero.

# Archi, circonferenze

Il comando

```
\psarc[par]{arrows}(x,y){radius}{angleA}{angleB}
```

disegna un arco di centro  $(x, y)$ , di raggio `radius`, fra gli angoli `angleA` e `angleB`.

# Archi, circonferenze

Il comando

```
\psarc[par]{arrows}(x,y){radius}{angleA}{angleB}
```

disegna un arco di centro  $(x, y)$ , di raggio `radius`, fra gli angoli `angleA` e `angleB`.

Il comando

```
\pscircle[par](x,y){radius}
```

disegna una circonferenza di centro  $(x, y)$  e raggio `radius`,

# Ellissi e parabola

Il comando

```
\psellipse[par]( $x_0, y_0$ )( $x_1, y_1$ )
```

disegna un'ellisse di centro  $(x_0, y_0)$  e semiassi  $x_1$  e  $y_1$ .



# Ellissi e parabola

Il comando

```
\psellipse[par]( $x_0, y_0$ )( $x_1, y_1$ )
```

disegna un'ellisse di centro  $(x_0, y_0)$  e semiassi  $x_1$  e  $y_1$ .

Il comando

```
\psellipticarc[par]{arrows}( $x_0, y_0$ )( $x_1, y_1$ ){angleA}{angleB}
```

disegna un arco ellittico dall'angolo  $angleA$  e  $angleB$ , con centro in  $(x_0, y_0)$  e semiassi  $x_1$  e  $y_1$ .

# Ellissi e parabola

Il comando

```
\psellipse[par](x_0,y_0)(x_1,y_1)
```

disegna un'ellisse di centro  $(x_0, y_0)$  e semiassi  $x_1$  e  $y_1$ .

Il comando

```
\psellipticarc[par]{arrows}(x_0,y_0)(x_1,y_1){angleA}{angleB}
```

disegna un arco ellittico dall'angolo  $angleA$  e  $angleB$ , con centro in  $(x_0, y_0)$  e semiassi  $x_1$  e  $y_1$ .

Il comando

```
\parabola[par]{arrows}(x_0,y_0)(x_1,y_1)
```

Partendo da  $(x_0, y_0)$  disegna una parabola di vertice  $(x_1, y_1)$ .

# Interpolazioni

Il comando

```
\pscurve[par]{arrows}(x_1, y_1) \cdots (x_n, y_n)
```

disegna una curva che interpola i punti di coordinate  $(x_1, y_1) \cdots (x_n, y_n)$ .

# Interpolazioni

Il comando

```
\pscurve[par]{arrows}(x_1, y_1) \cdots (x_n, y_n)
```

disegna una curva che interpola i punti di coordinate  $(x_1, y_1) \cdots (x_n, y_n)$ .

Il comando

```
\psccurve[par]{arrows}(x_1, y_1) \cdots (x_n, y_n)
```

disegna una curva chiusa che interpola i punti di coordinate  $(x_1, y_1) \cdots (x_n, y_n)$ .

# Punti

Il comando

```
\psdot[par](x_1, y_1)
```

# Punti

Il comando

```
\psdot[par](x_1, y_1)
```

Che cosa sia il *punto*, dipende dal valore del parametro `dotstyle`.

# Punti

Il comando

```
\psdot[par](x_1, y_1)
```

Che cosa sia il *punto*, dipende dal valore del parametro `dotstyle`.  
Per esempio:

# Punti

Il comando

```
\psdot [par] (x1, y1)
```

Che cosa sia il *punto*, dipende dal valore del parametro `dotstyle`.

Per esempio:

`dotstyle=*` disegna un punto tondo nero;

`dotstyle=o` disegna un punto tondo vuoto;

`dotstyle=square` disegna un punto quadrato vuoto;

`dotstyle=square*` disegna un punto quadrato nero;

`dotstyle=asterisk` disegna un asterisco;

⋮



# Punti

Il comando

```
\psdot [par] (x1, y1)
```

Che cosa sia il *punto*, dipende dal valore del parametro `dotstyle`.

Per esempio:

`dotstyle=*` disegna un punto tondo nero;

`dotstyle=o` disegna un punto tondo vuoto;

`dotstyle=square` disegna un punto quadrato vuoto;

`dotstyle=square*` disegna un punto quadrato nero;

`dotstyle=asterisk` disegna un asterisco;

⋮

La dimensione del punto è determinata dal parametro `dotsize`.

# Figure

## Estensioni di pstricks

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

Il comando

```
\psplot[par]{ $x_{min}$ }{ $x_{max}$ }{funzione}
```

disegna il grafico della `funzione` fra le ascisse  $x_{min}$  e  $x_{max}$ .

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

Il comando

```
\psplot[par]{ $x_{min}$ }{ $x_{max}$ }{funzione}
```

disegna il grafico della `funzione` fra le ascisse  $x_{min}$  e  $x_{max}$ .

Per scrivere l'equazione della funzione in normale forma algebrica è necessario utilizzare il comando `\psset{algebraic=true}`.

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

Il comando

```
\psplot[par]{ $x_{min}$ }{ $x_{max}$ }{funzione}
```

disegna il grafico della `funzione` fra le ascisse  $x_{min}$  e  $x_{max}$ .

Per scrivere l'equazione della funzione in normale forma algebrica è necessario utilizzare il comando `\psset{algebraic=true}`.

Diversamente si deve scrivere con la sintassi `postscript`.

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

Il comando

```
\psplot[par]{ $x_{min}$ }{ $x_{max}$ }{funzione}
```

disegna il grafico della `funzione` fra le ascisse  $x_{min}$  e  $x_{max}$ .

Per scrivere l'equazione della funzione in normale forma algebrica è necessario utilizzare il comando `\psset{algebraic=true}`.

Diversamente si deve scrivere con la sintassi `postscript`.

Si confronti

```
3 x 2 exp mul 2 x mul sub 15 x mul 2 sub div
```

# Grafici di funzioni

Serve il pacchetto `pst-plot`.

Il comando

```
\psplot[par]{ $x_{min}$ }{ $x_{max}$ }{funzione}
```

disegna il grafico della `funzione` fra le ascisse  $x_{min}$  e  $x_{max}$ .

Per scrivere l'equazione della funzione in normale forma algebrica è necessario utilizzare il comando `\psset{algebraic=true}`.

Diversamente si deve scrivere con la sintassi `postscript`.

Si confronti

```
3 x 2 exp mul 2 x mul sub 15 x mul 2 sub div
```

con

```
(3*x^2-2*x)/(15*x-2)
```



# Assi cartesiani

Per gli assi si usa il comando

```
\psaxes[par]{arrows}(x_0,y_0)(x_1,y_1)(x_2,y_2)[Xlabel,Xangle]  
[Ylabel,Yangle]
```

che disegna un asse delle ascisse fra  $x_1$  e  $x_2$  e uno delle ordinate fra  $y_1$  e  $y_2$ , che si incontrano in  $x_0, y_0$ .

# Assi cartesiani

Per esempio, il comando

```
\psaxes[linewidth=0.5pt,dx=1,dy=1,ticksiz=2pt,  
labelFontSize=\scriptstyle]{->}(0,0)(-3,-2)(3,2.5)  
[\(x\),-90][\(\f(x)\),180]
```

disegna due assi cartesiani di spessore  $0.5\text{pt}$ , con segni spazati di una unità e di lunghezza  $2\text{pt}$ , con numerazione con carattere piccolo; l'asse delle ascisse va da  $-3$  a  $3$  e si chiama  $x$ ; l'asse delle ordinate va da  $-2$  a  $2.5$  e si chiama  $f(x)$ ; il simbolo  $x$  è posto sotto la freccia, il simbolo  $y$  è posto a sinistra della freccia.

# Nodi

Nel pacchetto `pst-plot` è compresa la possibilità di definire dei `nodi` con la sintassi

```
\pnode( $x_0, y_0$ ){name}
```

# Nodi

Nel pacchetto `pst-plot` è compresa la possibilità di definire dei `nodi` con la sintassi

```
\pnode( $x_0, y_0$ ){name}
```

In questo modo si dà il nome `name` al punto di coordinate  $(x_0, y_0)$ .

# Nodi

Nel pacchetto `pst-plot` è compresa la possibilità di definire dei `nodi` con la sintassi

```
\pnode( $x_0$ ,  $y_0$ ){name}
```

In questo modo si dà il nome `name` al punto di coordinate  $(x_0, y_0)$ .

Se non si utilizza `pst-plot` è possibile utilizzare i nodi con il pacchetto `pst-node`.

# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

L'armamentario dell'ottica geometrica (specchi, lenti, prismi, ...) sono costruiti con il pacchetto `pst-optics`.

# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

L'armamentario dell'ottica geometrica (specchi, lenti, prismi, ...) sono costruiti con il pacchetto `pst-optics`.

Per disegnare molle, eliche e spire si usa il pacchetto `pst-coil`.



# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

L'armamentario dell'ottica geometrica (specchi, lenti, prismi, ...) sono costruiti con il pacchetto `pst-optics`.

Per disegnare molle, eliche e spire si usa il pacchetto `pst-coil`.

Per per costruire un codice a barre si usa il pacchetto `pst-barcode`.

# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

L'armamentario dell'ottica geometrica (specchi, lenti, prismi, ...) sono costruiti con il pacchetto `pst-optics`.

Per disegnare molle, eliche e spire si usa il pacchetto `pst-coil`.

Per per costruire un codice a barre si usa il pacchetto `pst-barcode`.

Per ogni altra informazione si rimanda alla guida ufficiale  
<http://ctan.mirror.garr.it/mirrors/CTAN/graphics/pstricks/base/doc/pstricks-doc.pdf>.

# Altro

I circuiti elettrici sono costruiti con il pacchetto `pst-circ`.

L'armamentario dell'ottica geometrica (specchi, lenti, prismi, ...) sono costruiti con il pacchetto `pst-optics`.

Per disegnare molle, eliche e spire si usa il pacchetto `pst-coil`.

Per per costruire un codice a barre si usa il pacchetto `pst-barcode`.

Per ogni altra informazione si rimanda alla guida ufficiale  
<http://ctan.mirror.garr.it/mirrors/CTAN/graphics/pstricks/base/doc/pstricks-doc.pdf>.

Si segnala anche la guida in italiano: <http://www.guitex.org/home/images/ArsTeXnica/AT004/IntroduzioneeaPSTricks.pdf>.

# Come utilizzare le figure

Non funziona con PDF $\text{\LaTeX}$ , ma con il semplice  $\text{\LaTeX}$ .

# Come utilizzare le figure

Non funziona con PDF $\text{\LaTeX}$ , ma con il semplice  $\text{\LaTeX}$ .

Con il comando `latex figura.tex` si genera il file `figura.dvi`.

# Come utilizzare le figure

Non funziona con PDF $\LaTeX$ , ma con il semplice  $\LaTeX$ .

Con il comando `latex figura.tex` si genera il file `figura.dvi`.

Con il comando `dvips figura.dvi` si ottiene il file `figura.ps`.

# Come utilizzare le figure

Non funziona con PDF $\LaTeX$ , ma con il semplice  $\LaTeX$ .

Con il comando `latex figura.tex` si genera il file `figura.dvi`.

Con il comando `dvips figura.dvi` si ottiene il file `figura.ps`.

Questa figura ha l'estensione dell'intera pagina,

# Come utilizzare le figure

Non funziona con PDF $\LaTeX$ , ma con il semplice  $\LaTeX$ .

Con il comando `latex figura.tex` si genera il file `figura.dvi`.

Con il comando `dvips figura.dvi` si ottiene il file `figura.ps`.

Questa figura ha l'estensione dell'intera pagina, quindi quando viene importata nel file principale si porta dietro l'intera pagina; mentre si vuole importare *la sola figura*.



# Come utilizzare le figure

Non funziona con PDF $\text{\LaTeX}$ , ma con il semplice  $\text{\LaTeX}$ .

Con il comando `latex figura.tex` si genera il file `figura.dvi`.

Con il comando `dvips figura.dvi` si ottiene il file `figura.ps`.

Questa figura ha l'estensione dell'intera pagina, quindi quando viene importata nel file principale si porta dietro l'intera pagina; mentre si vuole importare *la sola figura*.

## Attenzione

È necessario evitare i numeri di pagina con il comando

```
\pagestyle{empty}
```

# Il formato .eps

È quindi necessario *ritagliare* la figura nelle sue dimensioni reali, producendo un file **Encapsulated PostScript (.eps)**.

## Il formato .eps

È quindi necessario *ritagliare* la figura nelle sue dimensioni reali, producendo un file **Encapsulated PostScript** (.eps).

Questo viene fatto, da terminale, con il comando

```
dvips -E -o figura.eps figura.dvi
```

## Il formato .eps

È quindi necessario *ritagliare* la figura nelle sue dimensioni reali, producendo un file **Encapsulated PostScript (.eps)**.

Questo viene fatto, da terminale, con il comando

```
dvips -E -o figura.eps figura.dvi
```

A questo punto la figura in formato **.eps** può essere importata con il comando

```
\begin{figure}  
\includegraphics[scale=1]{figura}  
\caption{Ecco la figura.}  
\end{figure}
```

# Figure

## Il pacchetto TikZ

# Vantaggi e svantaggi

Il pacchetto **TikZ** consente di costruire la figura scrivendone il codice direttamente nel documento principale.

# Vantaggi e svantaggi

Il pacchetto **TikZ** consente di costruire la figura scrivendone il codice direttamente nel documento principale.

Si compila in PDF $\text{\LaTeX}$ .

Il suo uso è piú complesso.

## Per esempio

Per tracciare una linea spezzata si usa un comando del tipo

```
\begin{tikzpicture}[scale=1]  
\draw (0,0) -- (1,2) -- (2,3) -- (4,0);  
\end{tikzpicture}
```

si ottiene

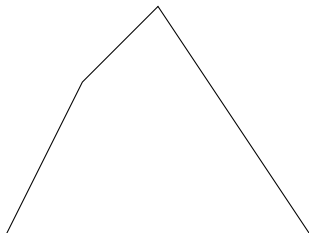


## Per esempio

Per tracciare una linea spezzata si usa un comando del tipo

```
\begin{tikzpicture}[scale=1]  
\draw (0,0) -- (1,2) -- (2,3) -- (4,0);  
\end{tikzpicture}
```

si ottiene

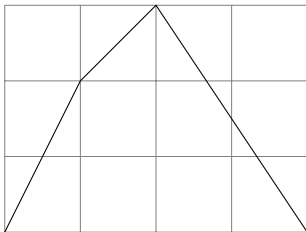


## Per esempio

Per tracciare una linea spezzata si usa un comando del tipo

```
\begin{tikzpicture}[scale=1]
\draw[help lines] (0,0) grid (4,3);
\draw (0,0) -- (1,2) -- (2,3) -- (4,0);
\end{tikzpicture}
```

si ottiene



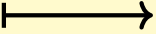



# Linee

Possono avere frecce, vari spessori, tipologie, colori:



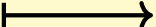

# Linee

Possono avere frecce, vari spessori, tipologie, colori:

```
\draw[->,red] (0,0) -- (2,0); 
\draw[<->,very thin,green] (0,0) -- (2,0); 
\draw[|->,ultra thick] (0,0) -- (2,0); 
\draw[dashed,line width=0.2cm] (0,0) -- (1,0); 
\draw[dotted,thick] (0,0) -- (2,0); .....
```

# Linee

Possono avere frecce, vari spessori, tipologie, colori:

```
\draw[->,red] (0,0) -- (2,0); 
\draw[<->,very thin,green] (0,0) -- (2,0); 
\draw[|->,ultra thick] (0,0) -- (2,0); 
\draw[dashed,line width=0.2cm] (0,0) -- (1,0); 
\draw[dotted,thick] (0,0) -- (2,0); .....
```

La figura può essere inserita  anche in mezzo  $\square$  al testo.

# Varie figure geometriche

Con i comandi

```
\begin{tikzpicture}  
\draw[red,rounded corners] (0,0) rectangle (2,1.5);  
\draw[green,dashed] (3,0.75) circle (0.5);  
\draw[thick] (5,1.5) arc (90:-30:1.5);  
\end{tikzpicture}
```

# Varie figure geometriche

Con i comandi

```
\begin{tikzpicture}  
\draw[red,rounded corners] (0,0) rectangle (2,1.5);  
\draw[green,dashed] (3,0.75) circle (0.5);  
\draw[thick] (5,1.5) arc (90:-30:1.5);  
\end{tikzpicture}
```

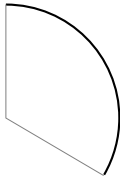
si ottiene

# Varie figure geometriche

Con i comandi

```
\begin{tikzpicture}  
\draw[red,rounded corners] (0,0) rectangle (2,1.5);  
\draw[green,dashed] (3,0.75) circle (0.5);  
\draw[thick] (5,1.5) arc (90:-30:1.5);  
\end{tikzpicture}
```

si ottiene



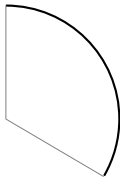


# Varie figure geometriche

Con i comandi

```
\begin{tikzpicture}  
\draw[red,rounded corners] (0,0) rectangle (2,1.5);  
\draw[green,dashed] (3,0.75) circle (0.5);  
\draw[thick] (5,1.5) arc (90:-30:1.5);  
\end{tikzpicture}
```

si ottiene



Per l'arco la notazione è assai scomoda.

# Nodi

Servono ad individuare i punti e ad assegnare un simbolo.

# Nodi

Servono ad individuare i punti e ad assegnare un simbolo.

Il comando è

```
\node[options](name) at ( $x_0, y_0$ ) {symbol}
```

# Nodi

Servono ad individuare i punti e ad assegnare un simbolo.

Il comando è

```
\node[options](name) at (x_0,y_0) {symbol}
```

Per esempio

```
\node[shape=circle,draw](A) at (0,0) {A}
```

Produce



# Nodi

Servono ad individuare i punti e ad assegnare un simbolo.

Il comando è

```
\node[options](name) at (x_0,y_0) {symbol}
```

Per esempio

```
\node[shape=circle,draw](A) at (0,0) {A}
```

Produce



Non è necessario indicare il nome; può essere utilizzato per collocare del testo in un disegno:

# Nodi

Servono ad individuare i punti e ad assegnare un simbolo.

Il comando è

```
\node[options](name) at (x_0,y_0) {symbol}
```

Per esempio

```
\node[shape=circle,draw](A) at (0,0) {A}
```

Produce



Non è necessario indicare il nome; può essere utilizzato per collocare del testo in un disegno:

```
\draw (0,0) -- (1,0);  
\node[right] at (1,0) {P};
```



# Riempimenti

Si usano i comandi `\fill` e `\filldraw`;

# Riempimenti

Si usano i comandi `\fill` e `\filldraw`;

```
\fill[fill=green] (0,0) circle (0.5);  
\filldraw[fill=green,draw=red] (2,0) circle (0.5);
```

Produce



# Riempimenti

Si usano i comandi `\fill` e `\filldraw`;

```
\fill[fill=green] (0,0) circle (0.5);  
\filldraw[fill=green,draw=red] (2,0) circle (0.5);
```

Produce



# Ombreggiatura

Si usa il comando `\shade`:

# Ombreggiatura

Si usa il comando `\shade`:

```
\shade[top color=yellow,bottom color=blue] (0,0)
rectangle (2,1);
\shade[left color=red,right color=black] (2.5,0)
rectangle (4.5,1);
\shade[inner color=green,outer color=cyan] (5,0)
rectangle (7,1);
\shade[ball color=brown] (8,0.5) circle (0.5);
```

# Ombreggiatura

Si usa il comando `\shade`:

```
\shade[top color=yellow,bottom color=blue] (0,0)
rectangle (2,1);
\shade[left color=red,right color=black] (2.5,0)
rectangle (4.5,1);
\shade[inner color=green,outer color=cyan] (5,0)
rectangle (7,1);
\shade[ball color=brown] (8,0.5) circle (0.5);
```

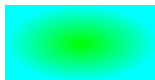
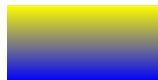
produce

# Ombreggiatura

Si usa il comando `\shade`:

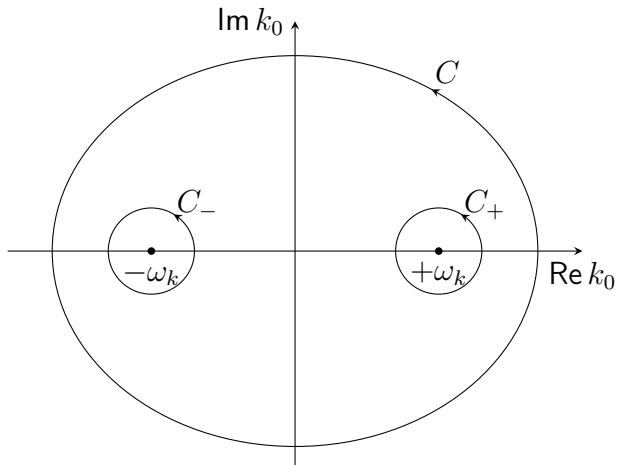
```
\shade[top color=yellow,bottom color=blue] (0,0)
rectangle (2,1);
\shade[left color=red,right color=black] (2.5,0)
rectangle (4.5,1);
\shade[inner color=green,outer color=cyan] (5,0)
rectangle (7,1);
\shade[ball color=brown] (8,0.5) circle (0.5);
```

produce



# Un esempio

Cammini di integrazione nel piano complesso  $k_0$

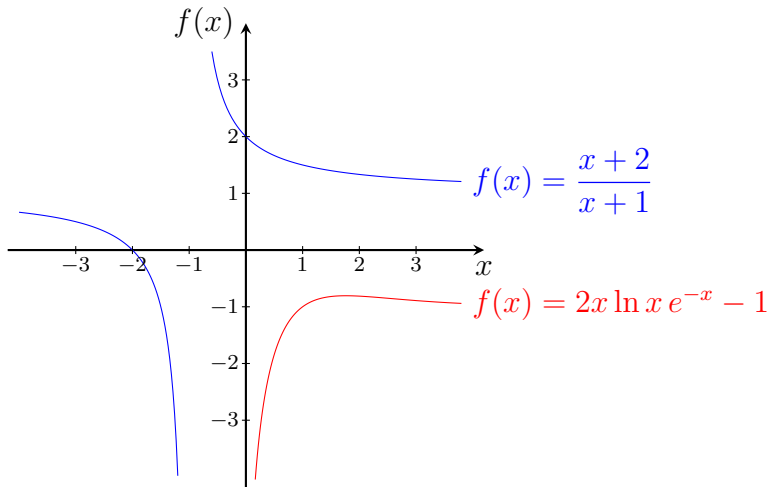


# Grafici di funzioni

Ecco una funzione razionale e una trascendente:

# Grafici di funzioni

Ecco una funzione razionale e una trascendente:





# Il pacchetto PGF

Sta per Portable Graphics Format.

# Il pacchetto PGF

Sta per **P**ortable **G**raphics **F**ormat.

Si deve richiamare il pacchetto **pgfplots**.

# Il pacchetto PGF

Sta per **Portable Graphics Format**.

Si deve richiamare il pacchetto **pgfplots**.

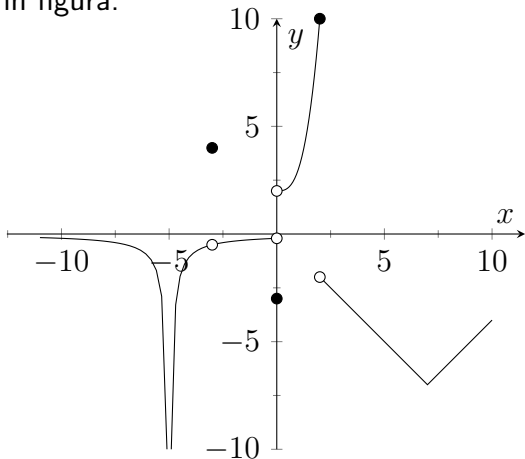
Consente di gestire al meglio assi e grafici.

# Esempio

**Problema:** Discutere i punti di discontinuità della funzione il cui grafico è dato in figura:

# Esempio

**Problema:** Discutere i punti di discontinuità della funzione il cui grafico è dato in figura:



# Geogebra

I grafici prodotti da **GEOGEBRA** possono essere esportati sia in formato **pstricks** che in formato **TikZ**.