Semplici figure con l'ambiente picture

Massimo Caschili

Sommario

Una breve guida: come usare picture, l'ambiente standard di LaTeX per creare semplici ma efficaci figure.

1 Introduzione

L'ambiente picture delimita uno spazio all'interno del quale è possibile inserire degli oggetti. Gli elementi sono inseriti con precisione usando un sistema di riferimento e le coordinate rispetto a tale sistema. La posizione relativa degli oggetti formerà la figura complessiva. Gli oggetti, inseriti con un comando specifico, possono essere linee, vettori (o frecce), rettangoli, cerchi, ovali, curve o testo semplice. Il disegno, nel suo complesso, è quindi composto assemblando questi oggetti, le cui dimensioni sono espresse in unità. Le modalità d'utilizzo delle unità di misura meriterebbero un articolo dedicato, ma al momento occorre solo sapere che il disegno costruito nell'ambiente picture è proporzionale all'unità di misura indicata dal parametro `\unitlength`. Nel caso non venga specificato, il valore standard adottato è pari a un punto tipografico (1pt, vedi tabella 1) che è possibile (e spesso conveniente) modificare nel seguente modo:

```
\setlength{\unitlength}{1mm}
```

A partire dal punto in cui è stato inserito il comando, l'unità avrà il valore indicato, che, in questo caso, è stata posta pari a un millimetro. Se si decidesse di modificarla, ponendola ad esempio pari a dieci millimetri, tutti gli oggetti all'interno dell'ambiente risulterebbero ingranditi di dieci volte *fuorché il testo*. La modifica del valore di `\unitlength` non ha effetti sul testo.

2 Dichiarazione dell'ambiente

L'ambiente ha la seguente sintassi:

```
\begin{picture}(Δx, Δy)(x_a, y_a)
⋮
Oggetti
⋮
\end{picture}
```

Gli argomenti Δx e Δy indicano rispettivamente la larghezza e l'altezza dell'area che conterrà il disegno, espressi nelle unità correnti, mentre le coordinate x_a e y_a individuano l'angolo inferiore sinistro dell'area delimitata dall'ambiente rispetto all'origine delle coordinate. Se si pone `(0,0)`, l'origine coincide con l'angolo inferiore sinistro dell'area: omettere `(`x_a, y_a`)` equivale a porre `(0,0)`.
Esempio:

```
\begin{picture}(50,40)(0,0)
⋮
\end{picture}
```

Questo codice delimita un'area con le caratteristiche evidenziate in figura 1. Le linee tratteggiate[1] delimitano il perimetro dell'area con larghezza 50 unità ed altezza 40 unità. Si noti la posizione dell'origine delle coordinate (0,0): l'angolo superiore destro avrà di conseguenza coordinate (50,40).

in	$1\ in = 2,54\ cm$
	$1\ in = 72,27\ pt$
pt	$1\ pt = 0,3515\ mm$
bp	$1\ in = 72\ bp$
pc	$1\ pc = 12\ pt$
dd	$1157\ dd = 1238\ pt$
	$1\ dd = 0,3759\ mm$
cc	$1\ cc = 12\ dd$
em	Misura relativa pari all'altezza della lettera M del carattere corrente
ex	Misura relativa pari alla larghezza della lettera x del carattere corrente

Tabella 1: Unità di misura

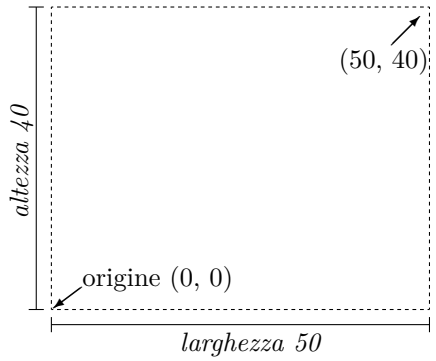

Figura 1: Area delimitata dall'ambiente picture

Più esattamente il codice è:

1. Il tratteggio è usato solo per evidenziare l'area definita dall'ambiente, che in realtà, ha bordi invisibili.

```
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(50,40)(0,0)
   oggetti
\end{picture}\end{center}
```

L'ambiente center centra la figura e, nella seconda riga, si fissa ad un millimetro l'unità di misura per l'ambiente `\picture`.

Proviamo ora a cambiare le coordinate dell'angolo inferiore sinistro, portandole a (5,5) come in figura 2.

L'area delimitata ha le stesse dimensioni di prima, come mostra il codice seguente usato per generarla:

```
\begin{picture}(50,40)(5,5)
⋮
\end{picture}
```

Il punto di coordinate (0,0) risulterà fuori dall'area 'protetta' dall'ambiente. Ciò significa che è possibile posizionare gli oggetti in qualunque punto della pagina rispetto all'area individuata dall'ambiente, come ad esempio il punto di coordinate (-19,0); ovviamente ciò comporterà possibili sovrapposizioni con altri elementi della pagina.

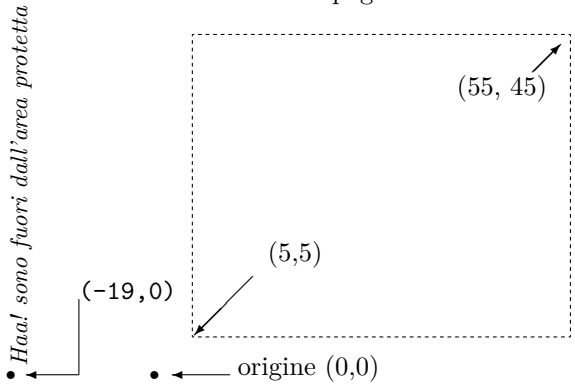

FIGURA 2: Origine diversa da (0,0) e oggetti fuori area

Per inserire gli oggetti nell'ambiente si usa il comando `\put`, la cui sintassi è:

`\put(x,y){`*oggetto*`}`

(x,y) Indicano le coordinate in cui inserire l'oggetto.

oggetto Indica uno qualunque degli oggetti dell'ambiente, l'oggetto può anche essere del semplice testo.

Per inserire più volte il medesimo oggetto:

`\multiput(x,y)(Δx,Δy){`*n*`}{`*oggetto*`}`

(x,y) Indicano le coordinate dalle quali parte l'inserimento multiplo dell'oggetto.

(Δx,Δy) Indicano la distanza (orizzontale Δx, e verticale Δy) che deve separare un oggetto dall'altro.

n Indica quante volte deve essere ripetuto l'inserimento, cioè quanti oggetti inserire.

oggetto Indica uno qualunque degli oggetti dell'ambiente, o un testo.

Inserire un oggetto significa mettere *il punto di riferimento* dell'oggetto in coincidenza delle coordinate indicate. È bene chiarire cosa sia questo *punto di riferimento*: ogni oggetto ha il suo e sarà indicato nella sezione dedicata (es. il punto di riferimento di un rettangolo è l'angolo inferiore sinistro). Per quanto concerne il punto di riferimento dei caratteri è bene far riferimento alla figura 3; si noti la posizione del punto di riferimento per i caratteri `G` e `g`. I punti di riferimento dei caratteri della stessa parola cadono tutti quanti sulla *linea di base*.

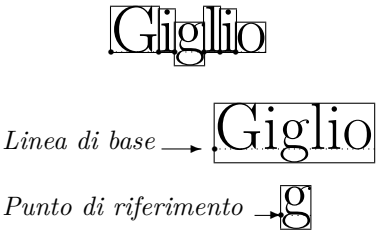

FIGURA 3: Linea di base e punto di riferimento

Semplificando, possiamo dire che TEX e LATEX trattano i caratteri come oggetti inseriti in scatole: per ogni parola sarà costruita un'unica scatola contenente la sequenza delle scatole relative a ciascuna lettera che forma la parola.

`\put(35,45){Testo}`

`(35,45)` → Testo

`\multiput(20,10)(10,10){3}{X}`

X

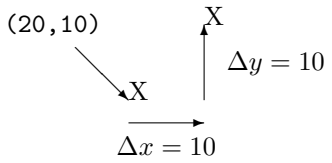

FIGURA 4: Esempio dell'uso di `\put` e `\multiput`

3 Rettangoli

Un rettangolo[2] può essere prodotto con uno dei seguenti comandi:

2. Nei manuali in inglese si parla genericamente di *box*, cioè scatola.

Opzioni	Significato
t	Alto, sul limite superiore del rettangolo, centrato rispetto alla larghezza.
b	Basso, sul limite superiore del rettangolo,centrato rispetto alla larghezza.
l	Sinistra, sul limite sinistro del rettangolo,centrato rispetto all'altezza.
r	Destra, sul limite destro del rettangolo,centrato rispetto all'altezza.
s	Estende il testo orizzontalmente per occupare tutta la lunghezza della scatola, centrato verticalmente.
tl	Alto e a sinistra, inserisce il testo presso l'angolo in alto a sinistra.
tr	Alto e a destra, inserisce il testo presso l'angolo in alto a destra.
bl	Basso e a sinistra, inserisce il testo presso l'angolo in basso a sinistra.
br	Basso e a destra, inserisce il testo presso l'angolo in basso a destra.

TABELLA 2: Le opzioni per i comandi `\makebox`, `\framebox`, `\dashbox` indicano in quale posizione, relativa al rettangolo sarà inserito il testo.

`\makebox(`$\Delta x, \Delta y$`)[opzioni]{ogg.}`
`\framebox(`$\Delta x, \Delta y$`)[opzioni]{ogg.}`
`\dashbox{ltratt}(`$\Delta x, \Delta y$`)[opzioni]{ogg.}`
Le opzioni sono riportate nella tabella 2.

Il comando `\makebox` crea un rettangolo con perimetro invisibile.

Il comando `\framebox` crea un rettangolo con perimetro continuo.

Il comando `\dashbox` crea un rettangolo con perimetro tratteggiato e lunghezza del tratteggio pari a `ltratt`; Δx e Δy definiscono rispettivamente la larghezza e l'altezza del rettangolo.

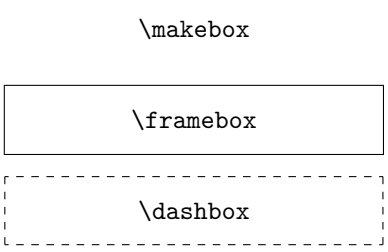

FIGURA 5: Tipi di rettangoli con relativo comando, hanno tutti le stesse dimensioni, il primo ha i bordi invisibili

Il comando `\shortstack` è un po' particolare: crea un rettangolo con bordi invisibili e opera come una tabella a colonna singola. Il testo, accettato come argomento, può essere disposto su più righe separate dal comando `\\`. L'indicazione riguardo

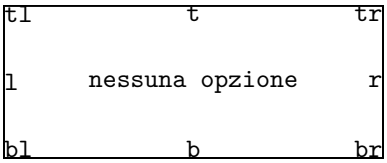

FIGURA 6: La disposizione delle lettere corrisponde alla disposizione data dalle relative opzioni.

l'allineamento del testo è opzionale e lo stile predefinito è quello centrato. Per ottenere un allineamento a destra o a sinistra si usa rispettivamente `r` e `l`. La sintassi completa del comando è:

`\shortstack[op]{ogg \\ ... \\...}`

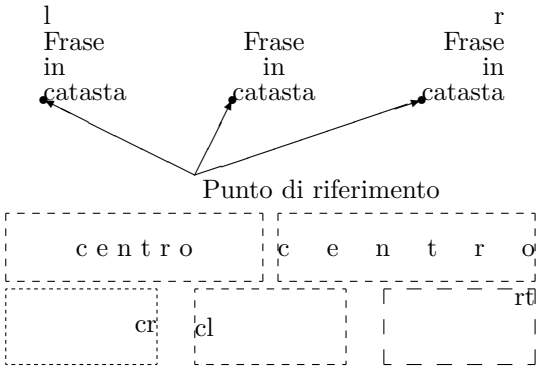

FIGURA 7: In alto esempi per il comando `\shortstack` (si notino i punti di riferimento). In basso risultato del comando `\dashbox`, senza opzione, con opzione `s`, con tratteggio di 0,5 1 e 2 millimetri, il testo indica l'opzione usata per posizionarlo.

4 Linee e frecce

L'oggetto *linea* (che in realtà è un segmento) si ottiene con il comando `\line`, mentre l'oggetto *freccia* (o *vettore*) si ottiene con il comando `\vector`. La sintassi per questi due comandi è:

`\line(`$\Delta x, \Delta y$`){lung}`

`\vector(`$\Delta x, \Delta y$`){lung}`

Gli incrementi Δx e Δy sono rappresentati da valori interi che vanno da -6 a $+6$ per `\line` e da -4 a $+4$ per `\vector`; `lung` è l'incremento della variabile x che individua l'ascissa del secondo estremo del segmento.

Vediamo un esempio in figura 8.

Il segmento AB è stato tracciato con il seguente codice:

`\put(10,10){\line(2,1){35}}`

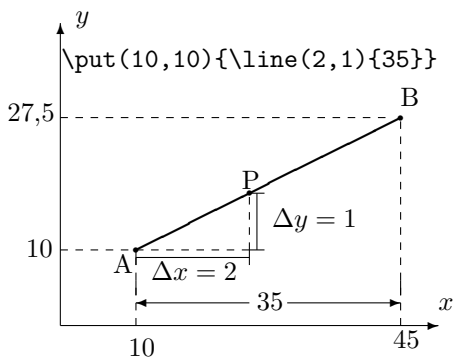

FIGURA 8: Oggetto linea e il codice relativo

Il comando `\put(10,10)` fissa il punto d'inserimento dell'oggetto `\line`, di coordinate `(10,10)`, che indichiamo con A. L'istruzione `{\line(2,1){35}}` traccia il segmento e il suo argomento, `(2,1)`, indica di quanto incrementare le variabili per dare al segmento una specifica inclinazione. In pratica, aumentando di 2 la variabile x (mi sposto verso destra di due unità) e di 1 la variabile y (mi sposto verso l'alto di una unità), si individua il generico punto 'successivo' del segmento. Dato che per due punti passa una e una sola retta, è così determinata l'inclinazione del segmento. Aumentando la variabile x di 35 unità, si determina la lunghezza del segmento e il punto B avrà ascissa $x = 10 + 35$ (l'ordinata risulterà $y = \frac{35}{2} = 17,5$).

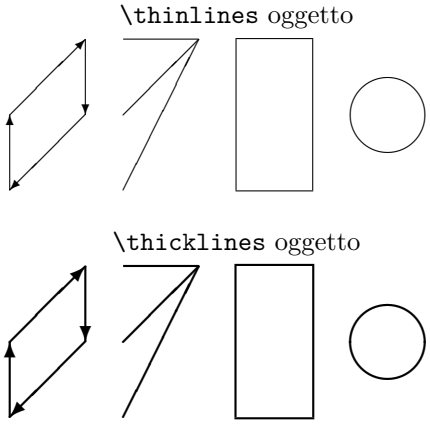

FIGURA 9: Linee e vettori con spessore `\thinlines`, in alto, e `\thicklines`.

Gli elementi `\line`, `\vector`, `\circle`, `\oval` possono essere disegnati anche con linee un più spesse, anteponendo il comando `\thicklines` al comando che produce l'elemento. Infatti, da quel punto in poi fino alla chiusura dell'ambiente, tutti gli oggetti saranno disegnati con una linea più spessa, a meno d'inserire il comando `\thinlines` per riportare lo spessore delle linee allo stato predefinito (più sottile).

Il comando `\linethickness{spess}` stabilisce lo spessore delle linee con `spess` qualunque, che deve essere indicato con l'unità di misura; `\linethickness` ha effetto solo su `\vector`, `\line` (a patto che siano in posizione orizzontale o verticale), sugli oggetti `\framebox`, `\dashbox` e sulle curve ottenute con `\qbezier`. Bisogna fare attenzione all'uso combinato di `\linethickness{spess}` e `\vector`, infatti, all'aumentare di `spess`, s'inspessisce il segmento ma non la punta della freccia (che risulta *inglobata* già con spessori modesti della linea). In tal caso, vista la resa scadente, se ne sconsiglia l'uso. Per la precisione, `\linethickness` non ha effetto su `\circle` e `\oval`.

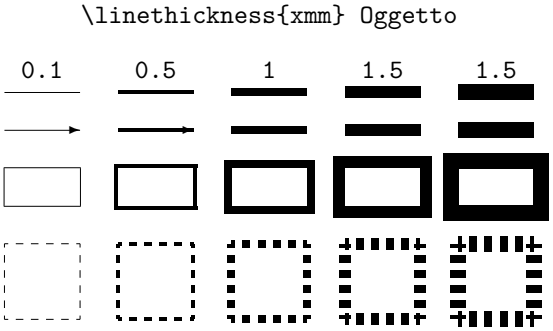

FIGURA 10: Linee e rettangoli con diverso spessore, si noti la pessima resa sul comando `\vector` e, per spessori notevoli, con `\dashbox`.

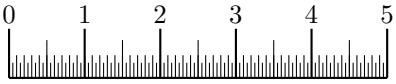

FIGURA 11: Una semplice applicazione del comando `\multiput` e `\line`, vedi il codice nella sezione 9.

5 Circonferenze e cerchi

Gli oggetti circonferenza e cerchio possono essere tracciati, rispettivamente, con i seguenti comandi:

```
\circle{diam}

\circle*{diam}
```

per i quali la misura del diametro è espressa nell'unità corrente. Il diametro massimo consentito per `\circle` è di 40pt, mentre il disco massimo riproducibile con `\circle*` ha diametro pari a 15pt. Se si utilizzano unità diverse dal punto tipografico, è bene operare la conversione nell'unità di misura che si intende usare, vedi tabella 1. Il centro del cerchio rappresenta il punto di riferimento dell'oggetto, si veda la figura 12.

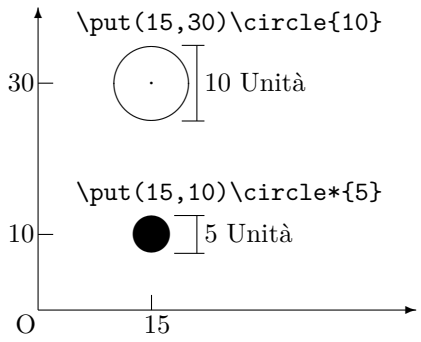

FIGURA 12: Si noti che il punto di riferimento è il centro. Non tutte le misure del diametro sono possibili, il diametro massimo per `\circle` è di 40pt, per `\circle*` il massimo è 15pt

6 Angoli arrotondati

Il comando `\oval`, a dispetto del nome, non costruisce veri e propri ovali o ellissi, ma piuttosto dei rettangoli con angoli smussati. L'effetto è ottenuto usando quarti di cerchio raccordati con segmenti, al limite di lunghezza nulla; la massima curvatura dei quarti di cerchio è pari a `\circle` con diametro massimo. La sintassi del comando è la seguente:

`\oval(`Δx`,`Δy`)[op]`

Le opzioni `op` permettono di disegnare parti dell'oggetto e se tali opzioni si omettono, come mostrato in figura 14, la figura sarà disegnata per intero. Si noti che in figura 13 il punto di riferimento dell'ovale coincide con il *centro* della figura intera.

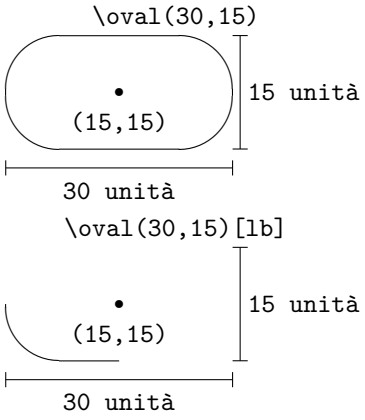

FIGURA 13: In alto, `\oval` è inserito con `\put(15,15)` e manca l'opzione, per cui la figura è intera; sotto, l'opzione è indicata e traccia solo la porzione inferiore sinistra dell'oggetto. Si noti che le dimensioni dei due oggetti, sono le stesse così come è la stessa la posizione relativa del punto di riferimento.

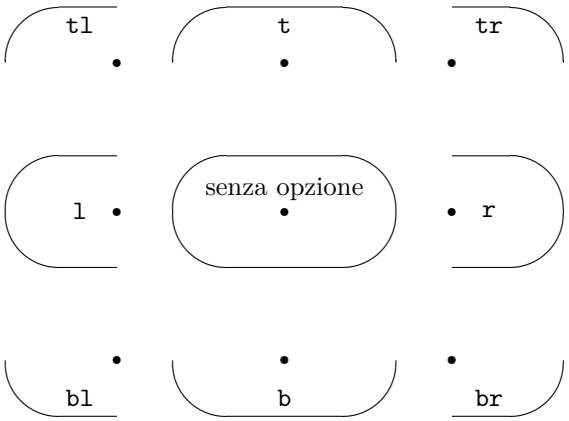

FIGURA 14: Panoramica delle opzioni per `\oval`

7 Curve

Disegnare curve generiche non è cosa da poco. Il comando `\qbezier` consente di disegnare curve sotto forma di archi individuati da tre punti (vedi figura 15), dove i segmenti sono tangenti alla curva. La sintassi del comando è:

`\qbezier(`x_1, y_1`)(`x_2, y_2`)(`x_3, y_3`)`
`\qbezier[n](`x_1, y_1`)(`x_2, y_2`)(`x_3, y_3`)`

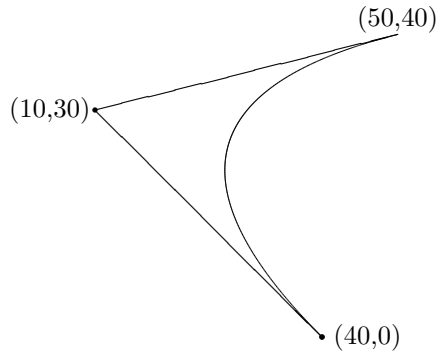

FIGURA 15: Curva `\qbezier` e relative tangenti

Se l'opzione è omessa, il tratto della curva risulterà continuo, altrimenti viene descritta una curva con un numero di punti pari a `n`. Il primo e il terzo punto fissano gli estremi della curva mentre il secondo punto è quello d'intersezione delle tangenti alla curva nei punti uno e tre. Se i tre punti `(`x_1, y_1`)(`x_2, y_2`)(`x_3, y_3`)` sono allineati, il risultato è rappresentato da una retta.

8 Ambienti annidati

Come si è visto, l'ambiente picture genera un'area le cui dimensioni sono specificate al momento della dichiarazione d'apertura dell'ambiente. Quest'area è a tutti gli effetti una *scatola* come le altre trattate da LaTeX, ma occorre fare attenzione ad individuare il suo punto di riferimento che chiamiamo `(`x_a, y_a`)`. Esso, infatti, coincide con l'angolo

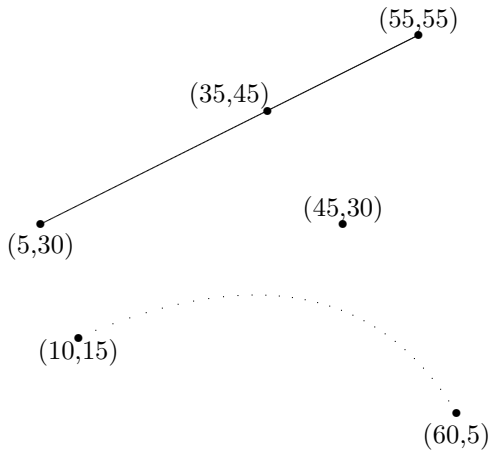

FIGURA 16: Curva `\qbezier` con opzione n = 30, e in alto il risutato con tre punti allineati, in evidenza i punti di riferimento.

inferiore sinistro solo nel caso di (x_a, y_a) = $(0, 0)$. Detto ciò, è possibile annidare un ambiente picture all'interno di un altro e la posizione indicata da `\put`(x_i, y_i) inserirà l'ambiente in modo che (x_a, y_a) = (x_i, y_i). L'esempio di figura 17 è costruito inserendo due ambienti all'interno di un terzo: in ciascuno degli ambienti è stato inserito un rettangolo di dimensioni uguali all'area dichiarata dall'ambiente stesso.

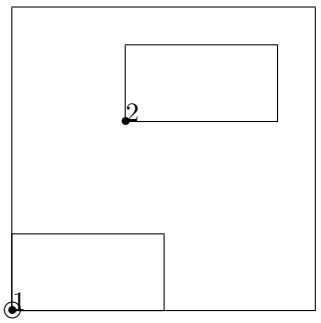

FIGURA 17: Ambienti annidati, tutti gli oggetti sono stati inseriti in `(0,0)`, si veda il codice relativo.

Codice figura 17

```
\begin{center}
\setlength{\unitlength}{1mm}
\begin{picture}(40,40)(0,0)
 \put(0,0){\framebox(40,40){}}
 \put(0,0){\circle{2}}
 \put(0,0){\begin{picture}(20,10)(0,0)
            \put(0,0){\framebox(20,10){}}
            \put(0,0){\circle*{1}}
            \put(0,0){1}
           \end{picture}}
 \put(0,0){\begin{picture}(20,20)(-15,-25)
            \put(0,0){\framebox(20,10){}}
            \put(0,0){\circle*{1}}
            \put(0,0){2}
           \end{picture}}
\end{picture}
```

Come si legge alla riga 6 (x_a, y_a) = $(0, 0)$, questo ambiente è inserito esattamente all'origine delle coordinate con `\put(0,0)`. Il secondo ambiente (riga 11) risulterà traslato perché (x_a, y_a) = $(-15, -25)$. L'annidamento degli ambienti semplifica la costruzione di disegni complessi, costruendo i singoli pezzi in ambienti distinti per poi assemblarli alla fine. Questa tecnica è stata usata per realizzare la figura 13, come indicato dal codice seguente:

Codice figura 13

```
\begin{picture}(45,55)(0,0)
 \put(22.5,53){\makebox(0,0){
       \texttt{\char`\\oval(30,15)}}}
 \put(0,0){\circle*{1}}
 \put(0,28){
   \begin{picture}(45,25)(0,0)
    \put(15,15){\oval(30,15)}
    \put(15,15){\circle*{1}}
    \put(0,5){\line(1,0){30}}
    \put(0,4){\line(0,1){2}}
    \put(30,4){\line(0,1){2}}
    \put(15,11){\makebox(0,0){%
       \texttt{(15,15)}}}
    \put(15,2){\makebox(0,0){%
       \texttt{30 unità}}}
    \put(32,14){\texttt{15 unità}}
    \put(31,7.5){\line(0,1){15}}
    \put(30,7.5){\line(1,0){2}}
    \put(30,22.5){\line(1,0){2}}
   \end{picture}}
  \put(22.5,25){\makebox(0,0){
        \texttt{\char`\\oval(30,15)[lb]}}}
  \put(0,0){
   \begin{picture}(45,25)(0,0)
    \put(15,15){\oval(30,15)[bl]}
    \put(15,15){\circle*{1}}
    \put(0,5){\line(1,0){30}}
    \put(0,4){\line(0,1){2}}
    \put(30,4){\line(0,1){2}}
    \put(15,11){\makebox(0,0){%
       \texttt{(15,15)}}}
    \put(15,2){\makebox(0,0){%
       \texttt{30 unità}}}
    \put(32,14){\texttt{15 unità}}
    \put(31,7.5){\line(0,1){15}}
    \put(30,7.5){\line(1,0){2}}
    \put(30,22.5){\line(1,0){2}}
   \end{picture}}
\end{picture}
```

9 Esempi e codice

In questa sezione sono presentati diversi esempi, con relativo codice commentato, e il codice di alcune figure dell'articolo.

Leggiamo il codice che riproduce la figura 11; la riga 1 apre l'ambiente center, che centrerà orizzontalmente la figura, e che si chiude con la riga 15. La riga 2 dichiara un contatore di nome num, il cui valore predefinito è zero e sarà usato nella riga 9. Nella riga 3 si fissa la dimensione dell'unità di misura. Con la riga 4, l'ambiente costruirà un rettangolo privo di bordi di 50 per 15 millimetri. L'angolo inferiore sinistro dell'area coincide con l'origine delle coordinate e nel caso in questione l'indicazione facoltativa (0,0) è stata esplicitata. Con le righe 5-6-7 s'inseriscono i segmenti per le tacche di uno, mezzo e cinque millimetri. La riga 8 indica d'inserire sei volte quanto eseguito dal suo argomento, che si chiude nella riga 10; \arabic{num} specifica che sarà mostrato il valore del contatore num in cifre arabe. Tale contatore ha valore iniziale pari a zero e, di volta in volta, viene incrementato di una unità tramite \addcounter{num}{1} (riga 10). Infine \multiput ripete l'inserimento del valore del contatore. Si noti che, per centrare al meglio i numeri, essi vengono racchiusi uno ad uno (riga 9) all'interno di scatole di dimensione zero in altezza e larghezza. Si realizza ciò per far sì che il loro punto di riferimento coincida con il *centro* del numero, il quale sarà posto nel punto indicato. La riga 11 dichiara \thicklines, col risultato d'inspessire la linea di tracciatura degli oggetti che seguono (vedi anche il codice della figura 9).

Codice del righello di figura 11

```
1 \begin{center}
2   \newcounter{num}
3   \setlength{\unitlength}{0.1cm}
4 \begin{picture}(50,15)(0,0)
5     \multiput(5,0)(10,0){5}{\line(0,1){5}}
6     \multiput(1,0)(1,0){49}{\line(0,1){3}}
7     \multiput(0.5,0)(1,0){50}{\line(0,1){2}}
8     \multiput(0,8.5)(10,0){6}{%
9        \makebox(0,0){\arabic{num}}
10       \addtocounter{num}{1}}
11  \thicklines
12    \put(0,0){\line(1,0){50}}
13    \multiput(0,0)(10,0){6}{\line(0,1){6.5}}
14 \end{picture}
15 \end{center}
```

Il codice che riproduce la figura 9 mostra l'uso dei comandi \thinlines (riga 4) e \thicklines (riga 15). Il significato di tali comandi è immediato: il primo, una volta dichiarato, influenza il codice che lo segue fino alla dichiarazione del secondo. I due comandi sono usati all'interno di un ambiente, pertanto essi hanno dominio solo al suo interno.

Codice figura 9

```
1   \begin{picture}(45,58)(0,0)
2     \put(22.5,53){\makebox(0,0){
3       \texttt{\char`\\thinlines} oggetto}}
4    \thinlines
5     \put(0,30){\vector(0,1){10}}
6     \put(0,40){\vector(1,1){10}}
7     \put(10,50){\vector(0,-1){10}}
8     \put(10,40){\vector(-1,-1){10}}
9     \put(15,30){\line(1,2){10}}
10    \put(15,40){\line(1,1){10}}
11    \put(15,50){\line(1,0){10}}
12    \put(30,30){\framebox(10,20){}}
13    \put(22.5,23){\makebox(0,0){
14      \texttt{\char`\\thicklines} oggetto}}
15  \thicklines
16    \put(0,0){\vector(0,1){10}}
17    \put(0,10){\vector(1,1){10}}
18    \put(10,20){\vector(0,-1){10}}
19    \put(10,10){\vector(-1,-1){10}}
20    \put(15,0){\line(1,2){10}}
21    \put(15,10){\line(1,1){10}}
22    \put(15,20){\line(1,0){10}}
23    \put(30,0){\framebox(10,20){}}
24 \end{picture}%
```

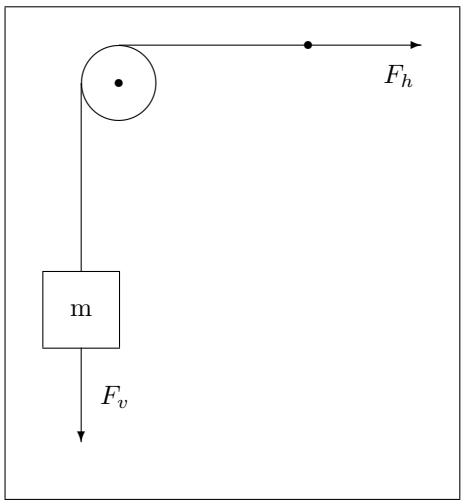

FIGURA 18: Questa figura è composta da cinque oggetti: \circle*, \circle, \framebox, \vector, \line

Il codice della figura 18 contiene un comando nuovo alla riga 2, \frame, che come argomento racchiude l'intero ambiente picture. Il suo unico scopo è quello d'inserire la cornice alla figura. La prima riga contiene l'indicazione dell'unità di misura e a questo proposito si ricorda che, modificando tale unità, la figura risulterebbe alterata in rapporto all'entità della modifica apportata e solo il testo alle righe 4, 10, 11 non ne verrebbe coinvolto. Se si desidera ingrandire o rimpicciolire il disegno è quindi consigliabile usare un'altra strada, che non sia quella di scalare l'unità di misura. Una buona *lente* è data dal comando \scalebox{sca}{oggetto} del pacchetto graphics, dove sca è un numero che indica il fattore d'ingrandimento. Tale estensione si può richiamare inserendo \usepackage{graphics} nel preambolo del documento.

Codice figura 18

```
\begin{center}\setlength{\unitlength}{0.5cm}
 \frame{\begin{picture}(12,13)(0,0)
 \put(2,4){\vector(0,-1){2.5}}
 \put(1,4){\framebox(2,2){m}}
 \put(2,6){\line(0,1){5}}
 \put(3,11){\circle{2}}
 \put(3,12){\vector(1,0){8}}
 \put(3,11){\circle*{0.2}}
 \put(8,12){\circle*{0.2}}
 \put(2.5,2.5){$F_v$}
 \put(10,11){$F_h$}
\end{picture}}%
```

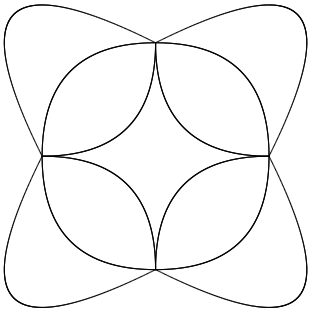

FIGURA 19: Un disegno con 12 curve `\qbezier`

La figura 19 è ottenuta con il seguente codice:

Codice per la figura 19

```
\begin{center}
\setlength{\unitlength}{1mm}
 \begin{picture}(60,60)(0,0)
  \qbezier(30,15)(15,15)(15,30)
  \qbezier(15,30)(15,45)(30,45)
  \qbezier(30,45)(45,45)(45,30)
  \qbezier(45,30)(45,15)(30,15)
  \qbezier(30,15)(0,0)(15,30)
  \qbezier(15,30)(0,60)(30,45)
  \qbezier(30,45)(60,60)(45,30)
  \qbezier(45,30)(60,0)(30,15)
  \qbezier(30,15)(30,30)(15,30)
  \qbezier(15,30)(30,30)(30,45)
  \qbezier(30,45)(30,30)(45,30)
  \qbezier(45,30)(30,30)(30,15)
  \end{picture}%
 \end{center}
```

La costruzione della figura 19 è semplice: le curve si saldano in corrispondenza del primo e terzo punto di ciascuna dichiarazione; se si prova a modificare il secondo punto di ciascuna curva, il risultato sarà una maggiore o minore concavità della curva. Non è semplice immaginare il risultato finale, occorre fare qualche prova compilando più volte il codice. Onde evitare di perdere tempo in tentativi inutili, prima di ottenere quanto si desidera, è bene evitare di seguire la tentazione che ci porta a scrivere immediatamente il codice del disegno. È preferibile disegnare con cura tutti gli elementi della figura su un foglio quadrettato o di carta millimetrata e aiutarsi con l'analoga griglia elettronica messa a disposizione dal pacchetto `graphpap`. Tale estensione deve essere richiamata nel preambolo attaverso `\usepackage{graphpap}`. Nella figura 20 è riportato un esempio di griglia. La riga 24, che riporta il comando `\graphpaper`, costruisce la griglia con le stesse dimensioni dichiarate per l'ambiente e con punto di riferimento l'origine del sistema di coordinate. Le righe della griglia sono spaziate di due unità.

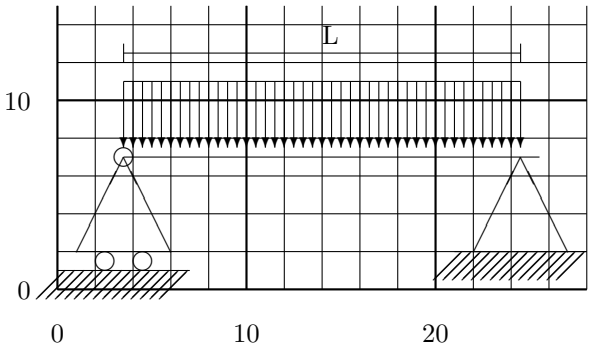

FIGURA 20: Disegno con griglia di riferimento.

Codice per la figura 20

```
\begin{center}
\setlength{\unitlength}{0.25cm}
\begin{picture}(28,15)(0,0)
  \put(0,1){\line(1,0){7}}
  \multiput(0.4,1)(0.5,0){14}{%
    \line(-1,-1){1.5}}
  \put(2.5,1.5){\circle{1}}
  \put(4.5,1.5){\circle{1}}
  \put(1,2){\line(1,0){5}}
  \put(1,2){\line(1,2){2.5}}
  \put(6,2){\line(-1,2){2.5}}
  \put(3.5,7){\circle{1}}
  \put(3.5,7){\line(1,0){22}}
  \put(22,2){\line(1,0){5}}
  \put(22,2){\line(1,2){2.5}}
  \put(27,2){\line(-1,2){2.5}}
  \put(21,2){\line(1,0){7}}
  \multiput(21.4,2)(0.5,0){14}{%
    \line(-1,-1){1.5}}
  \multiput(3.5,11)(0.5,0){43}{%
    \vector(0,-1){3.5}}
  \put(3.5,11){\line(1,0){21}}
  \put(3.5,12.5){\line(1,0){21}}
  \put(3.5,12){\line(0,1){1}}
  \put(24.5,12){\line(0,1){1}}
  \put(14,13){L}
  \graphpaper[2](0,0)(28,15)
\end{picture}
```

10 Conclusione

Come si è visto, l'ambiente offre un metodo veloce per realizzare disegni semplici e man mano che i disegni diventano più complessi il codice si

'appesantisce' di pari passo. Per quanto l'ambiente picture possa dare le sue soddisfazioni, non è lo strumento più adatto per realizzare disegni complessi; pacchetti quali epic, eepic, xy-pic, pstricks offrono qualche strumento in più rispetto a picture. Esistono poi numerosi altri pacchetti grafici per quasi ogni necessità e la cui documentazione è facilmente reperibile in rete.

Riferimenti bibliografici

Helmut Kopka, P. W. D. (2004). *Guide to LaTeX*. Addison-Wesley, Reading, MA, USA, 4ª edizione.

Lamport, L. (1994). *LaTeX: A Document Preparation System: User's Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 2ª edizione. Aggiornato per LaTeX 2ε.

▷ Massimo Caschili
Milano
massimo.caschili@tiscali.it