

Due parole su METAPOST

Alessandro Zaccagnini

25 giugno 2005

1 Struttura dei file METAPOST

La struttura di un file METAPOST è la seguente:

```
beginfig(1);  
  COMANDI metapost  
endfig;
```

```
beginfig(2);  
  COMANDI metapost  
endfig;
```

...

```
end
```

Una volta preparato il file `Figure.mp` che contiene, per esempio, la definizione di 5 figure, il comando

```
$ mpost Figure.mp  
This is MetaPost, Version 0.641 (Web2C 7.3.1)  
(Figure.mp [1] [2] [3] [4] [5] )  
5 output files written: Figure.1 .. Figure.5  
Transcript written on Figure.log.
```

produce 5 files in formato postscript con le figure. Per poter includere le figure così generate nei documenti scritti in $\LaTeX 2\epsilon$, è necessario dare il comando

```
\usepackage{graphicx}
```

nel preambolo del documento. La figura 1 può essere richiamata mediante il comando

```
\includegraphics{Figure.1}
```

Con alcune distribuzioni di Linux più vecchie (come Red Hat) è necessario cambiare nome ai files in modo che abbiano estensione `mps` (`Figure.1` diventa per esempio `Figure1.mps` e così via), oppure mettere link simbolici alle figure così generate. L'argomento di `includegraphics` dovrà essere modificato di conseguenza.

Nelle prossime pagine daremo qualche esempio di figure commentate, con i comandi di METAPOST più frequenti. Una breve introduzione a METAPOST è `/usr/share/texmf/doc/metapost/base/mpintro.ps`. Il manuale completo è `/usr/share/texmf/doc/metapost/base/mpman.ps`. Un'altra utile sorgente di informazioni è il §5.4.1 del file `/usr/share/texmf/doc/pdftex/base/pdfTeX-FAQ.pdf` (conviene procurarsi almeno la versione 0.12). Per disegnare grafici a partire da dati numerici utilizzando METAPOST si può consultare `/usr/share/texmf/doc/metapost/base/mpgraph.pdf`.

2 Esempi

```

% E' possibile inserire commenti come in
% TeX, facendoli precedere dal %

% Figura 1: Struttura di  $(\mathbb{Z}/11\mathbb{Z})^*$ 
% Gli elementi sono generati dalle
% potenze successive di 2
beginfig(1);

% Dichiarazione di variabili numeriche
% u rappresenta la 'scala' del disegno
numeric N, u, t, v;
N = 10;
t = 360/N;
v = 1.3;
u = 1.5cm;

% w e' un punto (coppia di coordinate)
pair w;
z0 = u*(1,0);
w = (0,0);

% Definiamo i punti z[i]
for i = 1 upto N-1:
    z[i] = z0 rotated (i*t);
endfor;

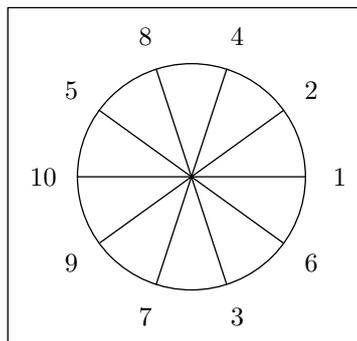
% Disegniamo un cerchio centrato
% nell'origine di diametro 2u
draw fullcircle scaled 2u;

% Ora disegniamo i 'raggi' e le
% etichette corrispondenti
% NB: j := 1 e' una 'assegnazione'
% e non un'uguaglianza e quindi
% possiamo assegnare un nuovo
% valore a j piu' tardi
j := 1;
for i = 0 upto N-1:
    draw w--z[i];
    label(decimal(j),v*z[i]);
    j := (2j) mod 11;
endfor;

% Mettiamo una scatola attorno alla
% figura lasciando dello spazio ai margini
draw bbox currentpicture scaled 1.1;

% Chiudiamo la figura corrente.
% Tutte le variabili sono cancellate
endfig;

```



Si noti che le *uguaglianze* denotate per mezzo di $=$ sono *equazioni* soddisfatte dalle coordinate dei punti, e non *assegnazioni*. Sono invece *assegnazioni* quelle denotate dal simbolo $:=$, ed, implicitamente, quelle dei cicli `for`.

Si notino i differenti tipi di dato: `numeric` e `pair`. Nelle prossime figure incontreremo anche dati di tipo `path` e `pen`. Si noti anche la sintassi del ciclo `for`: nella figura 2 ne vedremo una forma piú generale.

```

% Figura 2: Struttura di  $(\mathbb{Z}/24\mathbb{Z})^*$ 
% Questa volta il gruppo non e' ciclico
beginfig(2);

% Tutte le variabili s[...] sono numeriche
numeric N, u, t, v, s[];
N = 7;
t = 360/N;
v = 1.3;
u = 1.5cm;

pair w;
z0 = u*(1,0);
w = (0,0);

% Definizione dei punti necessari
% NB: z0 = z[0]
for i = 1 upto N-1:
    z[i] = z0 rotated (i*t);
endfor;

% La variabile 'p' conterra' una parte
% del nostro disegno, che useremo piu'
% volte.
path p;

% Costruiamo un 'petalo' partendo
% dall'origine andando verso Sud-Est
% e tornandovi puntando verso Sud-Ovest
% NB: 'p' NON viene disegnato,
% ma solo costruito con questo comando
p = w{ z0 rotated -0.5t}..z0{up}..
    {(-z0) rotated 0.5t}w;

% Disegniamo N copie del petalo
% ruotando 'p'
for i = 0 upto N-1:
    draw p rotated (i*t);
endfor;

% Variabile 'penna': salviamo il valore
% corrente della penna, e ne usiamo
% temporaneamente un'altra piu' grande
pen pp;
pp = currentpen;

% Scegliamo come nuova penna una penna
% circolare 3 volte piu' grande
pickup pencircle scaled 3;

% Assegniamo i valori delle etichette
% che competono ai singoli petali
i := 0;
% Ciclo 'for' generalizzato:
% 'j' assume solo i valori indicati
for j = 5, 7, 11, 13, 17, 19, 23:
    s[i] = j;
    i := i+1;
endfor;

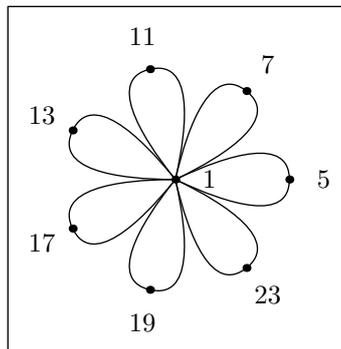
% Disegniamo i punti con le relative
% etichette
for i = 0 upto N-1:
    draw z[i];
    label (decimal(s[i]),v*z[i]);
endfor;

% Mettiamo l'etichetta '1' a 3/10 della
% strada fra w0 e z0
draw w;
label("1",.3[w, z0]);

% Torniamo alla penna iniziale
% per il bordo della scatola
pickup pp;
draw bbox currentpicture scaled 1.1;

endfig;

```



```

% Figura 3: Circonferenza trigonometrica
beginfig(3);
numeric a, b, u, v;
u = 2cm;
v = 1.4;
a = 0.6;
b = 0.8;

% Definizione dei punti in termini relativi
z0 = (0,0);
z1 = -z3 = (u,0);
z5 = -z7 = v*z1;
z6 = -z8 = z5 rotated 90;
z9 = u*(a,b);
x10 = 0;
z10 - z3 = whatever*(z9 - z3);

% Disegna la circonferenza
draw fullcircle scaled 2u;

% Disegna gli assi cartesiani
drawarrow z7--z5;
drawarrow z8--z6;

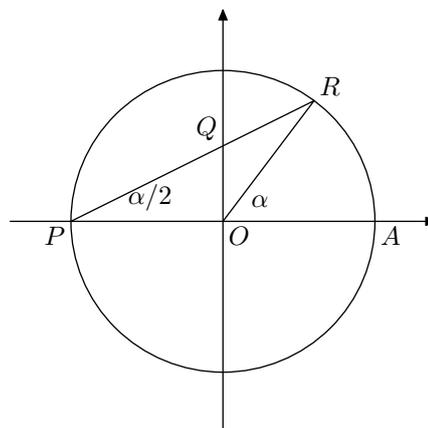
% Disegna i segmenti OR e PR
draw z0--z9--z3;

% Assegna nomi ai punti.
% Il suffisso di label indica la posizione
% dell'etichetta rispetto al punto
% indicato. Per esempio lrt indica
% lower right, ulft indica upper left,
% e cosi' via
label.lrt (btex $O$ etex,z0);
label.lrt (btex $A$ etex,z1);
label.ulft (btex $Q$ etex,z10);
label.ur (btex $R$ etex,z9);
label.llft (btex $P$ etex,z3);

% Etichette che usano comandi del TeX:
% tutto cio' che compare fra btex ed etex
% sara' composto usando il (plain) TeX.
% NB ricordarsi i $. $ quando si compongono
% formule
label.ur (btex $\strut\alpha\phantom{/2}$
etex, .15[z0,z1]);
label.top (btex $\strut\alpha/2$
etex, .55[z3,z0]);

endfig;

```



Si noti come il punto Q sia determinato come intersezione di due segmenti opportuni:

```

x10 = 0;
z10 - z3 = whatever*(z9 - z3);

```

La prima relazione implica che il punto z_{10} è sull'asse delle ordinate, mentre la seconda implica che il vettore $z_{10}-z_3$ è un multiplo del vettore z_9-z_3 , e quindi che z_{10} si trova sulla retta che contiene P ed R . Il fattore di proporzionalità è indicato con la parola riservata `whatever` perché il suo valore è irrilevante. Se `whatever` compare più volte in una medesima figura, il suo valore non è necessariamente lo stesso: in effetti, METAPOST genera una nuova variabile anonima ogni volta che incontra `whatever`.

```

% Figura 4: Crivello di Eratostene
beginfig(4);
numeric u;
path p[];

u = 0.6cm;

% Mettiamo gli interi da 1 a 144=12^2
% in un quadrato
for i = 1 upto 144:
  x[i] = u * ( (i-1) mod 12);
  y[i] = u * (12-((i-1) div 12));
  label(decimal(i),z[i]);
endfor;

% Identifichiamo i primi < 12
for i = 2, 3, 5, 7, 11:
  p[i] = fullcircle
    scaled .9u shifted z[i];
endfor;

% Le linee tratteggiate cancellano i
% multipli dei primi piccoli.
% Tagliamo via la parte di linea
% tratteggiata dentro i cerchi
p[1] = z[ 2]--z[134] cutbefore p[ 2];
p[4] = z[ 3]--z[135] cutbefore p[ 3];
p[6] = z[ 5]--z[130] cutbefore p[ 5];
p[8] = z[ 7]--z[112] cutbefore p[ 7];
p[9] = z[11]--z[121] cutbefore p[11];

draw p[1];
draw p[4];
draw p[6] dashed evenly scaled 4;
draw p[8] dashed evenly scaled 2;
draw p[9] dashed evenly;

% Crivello con il primo p = 2:
% 6 segmenti verticali
for i=4 step 2 until 12:
  draw z[i]--z[i+132];
endfor;

% Crivello con il primo p=3:
% 4 segmenti verticali
for i=6 step 3 until 12:
  draw z[i]--z[i+132];
endfor;

```

```

% Crivello con il primo p=5
draw z[10]--z[60] dashed evenly scaled 4;
draw z[15]--z[140] dashed evenly scaled 4;
draw z[20]--z[120] dashed evenly scaled 4;
draw z[25]--z[125] dashed evenly scaled 4;
draw z[85]--z[135] dashed evenly scaled 4;

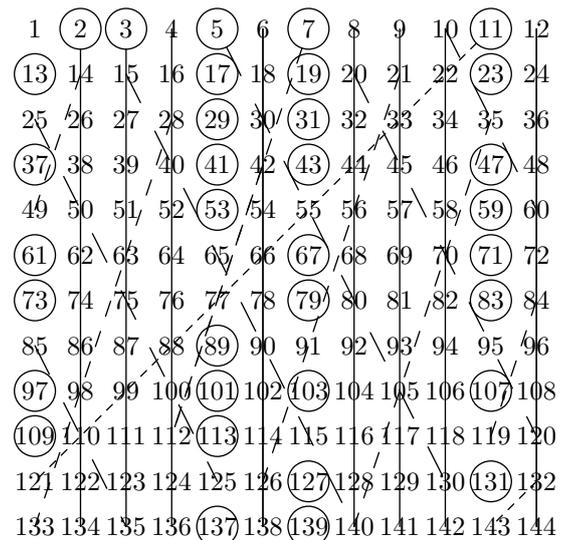
% Crivello con il primo p=7
draw z[14]--z[49] dashed evenly scaled 2;
draw z[42]--z[112] dashed evenly scaled 2;
draw z[21]--z[126] dashed evenly scaled 2;
draw z[28]--z[133] dashed evenly scaled 2;
draw z[35]--z[140] dashed evenly scaled 2;
draw z[84]--z[119] dashed evenly scaled 2;

% Crivello con il primo p=11
draw z[132]--z[143] dashed evenly;

% Metti un cerchio attorno ai numeri primi
for i=2, 3, 5, 7, 11, 13, 17, 19, 23, 29,
  31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
  73, 79, 83, 89, 97, 101, 103, 107, 109,
  113, 127, 131, 137, 139:
  draw fullcircle scaled .9u shifted z[i];
endfor;

endfig;

```



```

% Figura 5: Metodo dell'iperbole
beginfig(5);

% Definizione delle variabili
numeric u, d, f, N;
u = 0.4cm;
d = 1.5u;
N = 10;
f = 2.2;
path p[];

% I punti necessari per il disegno
% NB: z7 = (x7,y7) di default
z0 = (0,0);
z1 = z0+(-d,0);
z2 = z0+(0,-d);
z3 = u*(N,0);
z4 = z3+( d,0);
z5 = z3 rotated 90;
z6 = z4 rotated 90;
z9 = u*(f,N/f);
x7 = x9;
y7 = 0;
x8 = 0;
y8 = y9;

% Gli assi cartesiani
drawarrow z1--z4;
drawarrow z2--z6;

% Costruiamo due cammini chiusi
% Si noti il ciclo for dentro la
% definizione del cammino
p1 = z5--
  for i = 2 upto 2*N-1:
    u*(.5i, 2N/i){i*i,-4N}..
  endfor
  {N,-1}u*(N,1)--z3--z0--cycle;

p2 = z7--z9--z8--z0--cycle;

```

```

% Riempiamo l'interno di un cammino chiuso
% con un colore
fill p1 withcolor .8 white;
fill p2 withcolor .6 white;

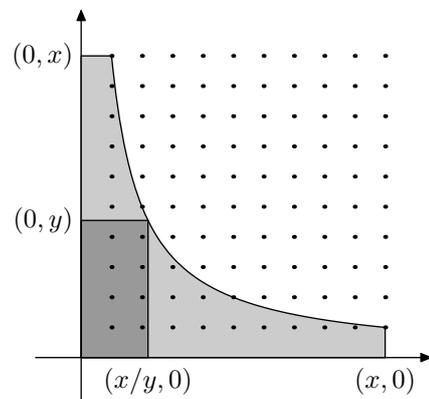
% Disegniamo i cammini
draw p1;
draw p2;

% Usiamo una penna piu' grande
% per i puntini
pickup pencircle scaled 1.5;
for i = 1 upto N:
  for j = 1 upto N:
    draw u*(i,j);
  endfor;
endfor;

% Etichette varie
label.bot(btex $\strut (x/y,0)$ etex, z7);
label.bot(btex $\strut (x,0) $ etex, z3);
label.lft(btex $(0,y)$ etex, z8);
label.lft(btex $(0,x)$ etex, z5);

endfig;

```



Si noti che l'*ordine* con cui si danno i comandi draw e fill è rilevante: se avessimo messo i due comandi fill p1 ...e fill p2 ...alla fine, i puntini nelle regioni in grigio non sarebbero stati visibili. Se invece avessimo scambiato fra loro i comandi fill, tutta l'area sarebbe stata colorata in grigio chiaro.

```

% Figura 6: Se  $(a|p) = -1$ , la
% biiezione  $f(x) = ax \pmod p$  scambia
% l'insieme dei residui quadratici con
% quello dei non residui quadratici

beginfig(6)
  numeric u;
  pair v, w, dx;
  dx = (0.2, 0) * u;
  u = 1.5 cm;
  v = (1,0) rotated 30;
  w = (1,0) rotated -30;
  z0 = (0,0);

  label(decimal(i), z[i] - dx);
  dotlabel("", z[i]);
  x[20 + i] = -x[i];
  y[20 + i] = y[i];
  drawarrow z[i]{v} .. {w}z[i + 20];
  label(decimal((5*i) mod 13),
    z[i + 20] + dx);
  dotlabel("", z[i + 20]);
endfor;

% Coordinate dei punti del dominio
for i = 1, 3, 4:
  y[i] = 1.5 u;
  y[13 - i] = .7 u;
endfor;
for i = 2, 5, 6:
  y[i] = -.7 u;
  y[13 - i] = - 1.5u;
endfor;
for i = 1, 2, 7, 9:
  x[i] = -1.8u;
endfor;
for i = 3, 5, 8, 10:
  x[i] = -1.3u;
endfor;
for i = 4, 6, 11, 12:
  x[i] = -.8u;
endfor;

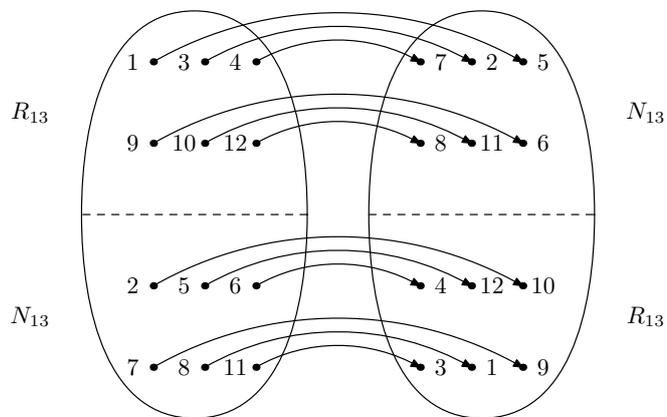
% Il diagramma di Eulero-Venn
z40 = -z42;
z41 = -z43;
y40 = y41 = 0;
x40 = -.3u;
x41 = -2.5u;
z44 = -z47;
z45 = -z46;
x44 = x45 = .5 (x40 + x41);
y44 = -y45 = 2u;

draw z40{up} .. z44{left} .. z41{down}
  .. z45{right} .. {up}z40;
draw z40 -- z41 dashed evenly;
draw z43{up} .. z46{left} .. z42{down}
  .. z47{right} .. {up}z43;
draw z42 -- z43 dashed evenly;

endfig;

% Le frecce rappresentano la funzione  $f$ 
for i = 1 upto 12:

```



```

% t e' il punto di partenza della freccia,
% n il valore corrispondente
vardef immagine(expr t, n) =
  begingroup
    pair v, w, s;
    v = (1,0) rotated 30;
    w = (1,0) rotated -30;
    label(decimal(n), t - dx);
    dotlabel("", t);
    s = (- xpart(t), ypart(t));
    drawarrow t{v} .. {w}s;
    label(decimal((5*n) mod 13), s + dx);
    dotlabel("", s);
  endgroup
enddef;

beginfig(7)
  numeric u;
  pair dx;
  dx = (0.2, 0) * u;
  u = 1.5 cm;
  z0 = (0,0);

  for i = 1, 3, 4:
    y[i] = 1.5 u;
    y[13 - i] = .7 u;
  endfor;
  for i = 2, 5, 6:
    y[i] = -.7 u;
    y[13 - i] = - 1.5u;
  endfor;
  for i = 1, 2, 7, 9:
    x[i] = -1.8u;
  endfor;
  for i = 3, 5, 8, 10:
    x[i] = -1.3u;
  endfor;
  for i = 4, 6, 11, 12:
    x[i] = -.8u;
  endfor;
  for i = 1 upto 12:
    immagine(z[i], i);
  endfor;

  label(btex $R_{13}$ etex, (-3, 1) * u);
  label(btex $N_{13}$ etex, (-3,-1) * u);
  label(btex $N_{13}$ etex, ( 3, 1) * u);
  label(btex $R_{13}$ etex, ( 3,-1) * u);

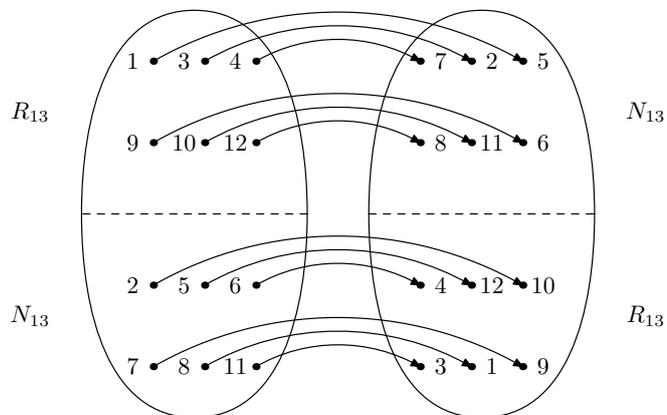
  z40 = -z42;
  z41 = -z43;
  y40 = y41 = 0;
  x40 = -.3u;
  x41 = -2.5u;
  z44 = -z47;
  z45 = -z46;
  x44 = x45 = .5 (x40 + x41);
  y44 = -y45 = 2u;

  draw z40{up} .. z44{left} .. z41{down}
    .. z45{right} .. {up}z40;
  draw z40 -- z41 dashed evenly;
  draw z43{up} .. z46{left} .. z42{down}
    .. z47{right} .. {up}z43;
  draw z42 -- z43 dashed evenly;

endfig;

% Fine del file
end

```



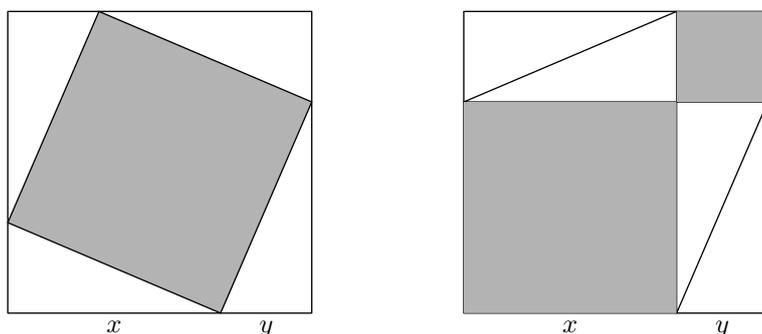
Come abbiamo visto, è possibile definire macro (con o senza parametri), includere macro del \TeX da usare all'interno di costrutti $\text{\bTeX} \dots \text{\eTeX}$, usare condizionali, linee tratteggiate, trasformazioni affini, ecc. Per i dettagli si rimanda ai manuali.

Altri comandi utili sono

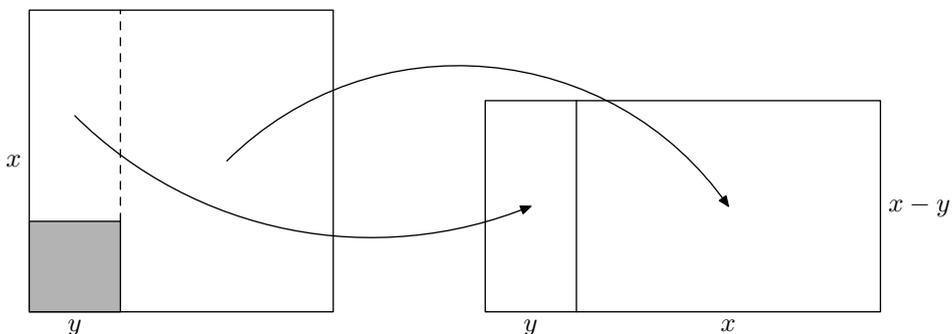
`if ... else ... fi`, `dotlabel`, `dotlabels`, `unfill`, `picture`, `halfcircle`, `transform`

3 Esercizi

1. Si riproduca la figura seguente, e la si utilizzi per dimostrare il Teorema di Pitagora.



2. Si riproduca la figura seguente, e la si utilizzi per dimostrare l'identità $(x-y)(x+y) = x^2 - y^2$.



3. Si riproduca la figura seguente, e la si utilizzi per dimostrare l'identità

$$\sum_{n=0}^{+\infty} \alpha^n = \frac{1}{1-\alpha} \quad \text{per } \alpha \in (0, 1).$$

Suggerimento: se $\overline{AF} = \overline{AC} = 1$, $\overline{EF} = \alpha$, allora $\overline{AB} : \overline{AC} = \overline{CD} : \overline{DE}$.

