GUIT *meeting* 2011

Creare grafici con **pgfplots**

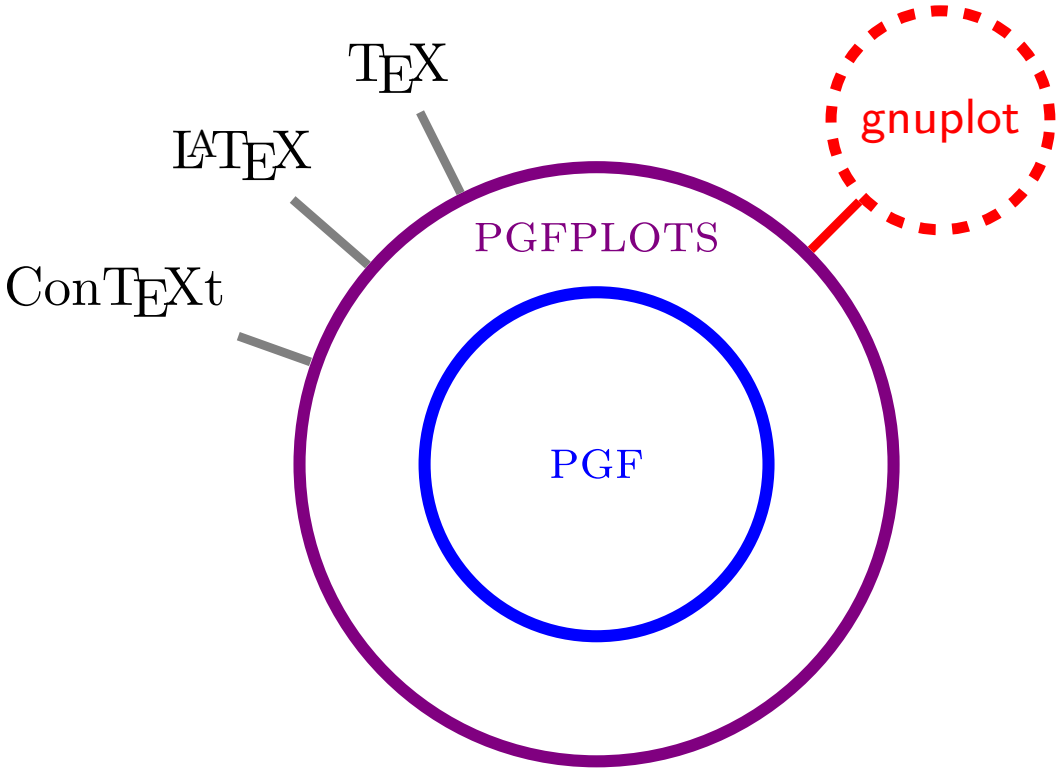
Agostino De Marco e Roberto Giacomelli

15 ottobre 2011

Progetto di un grafico

- **come costruire i dati numerici?**
  - espressione matematica valutata internamente
  - coordinate esplicite nel sorgente
  - tabelle in file su disco
  - programma esterno
- **come rappresentare i dati?**
  - piano cartesiano
  - piano polare
  - piano logaritmico
  - spazio cartesiano
- **come disegnare i dettagli?**
  - coerenza tipografica con il documento
  - dimensioni, griglie, marcature e formato assi
  - stili testuali, titoli ed etichette assi
  - elementi grafici aggiuntivi

Struttura del pacchetto **pgfplots**

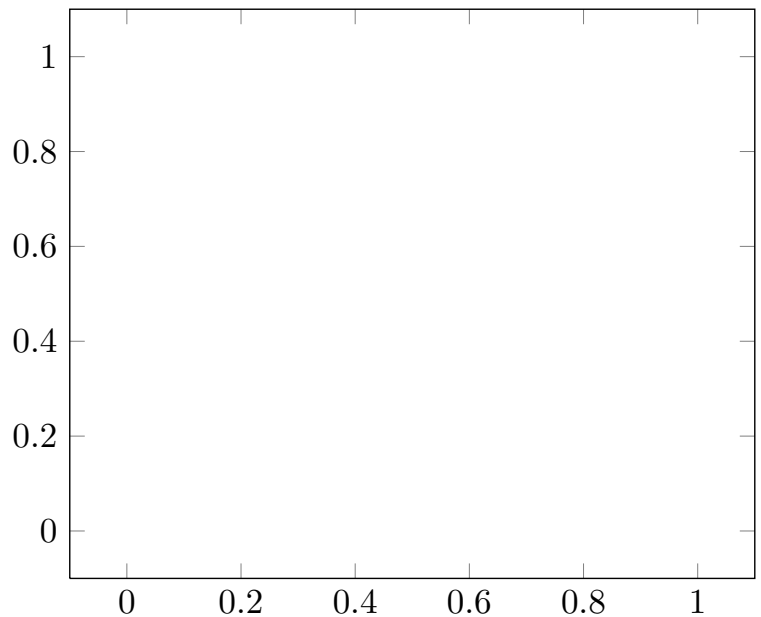


Sintassi dell'ambiente axis

Per LaTeX, il piano cartesiano è rappresentato in **pgfplots** dall'ambiente **axis** a sua volta racchiuso nell'ambiente **tikzpicture**.

↯ ricordiamoci che in **pgf** i comandi terminano con un **;**

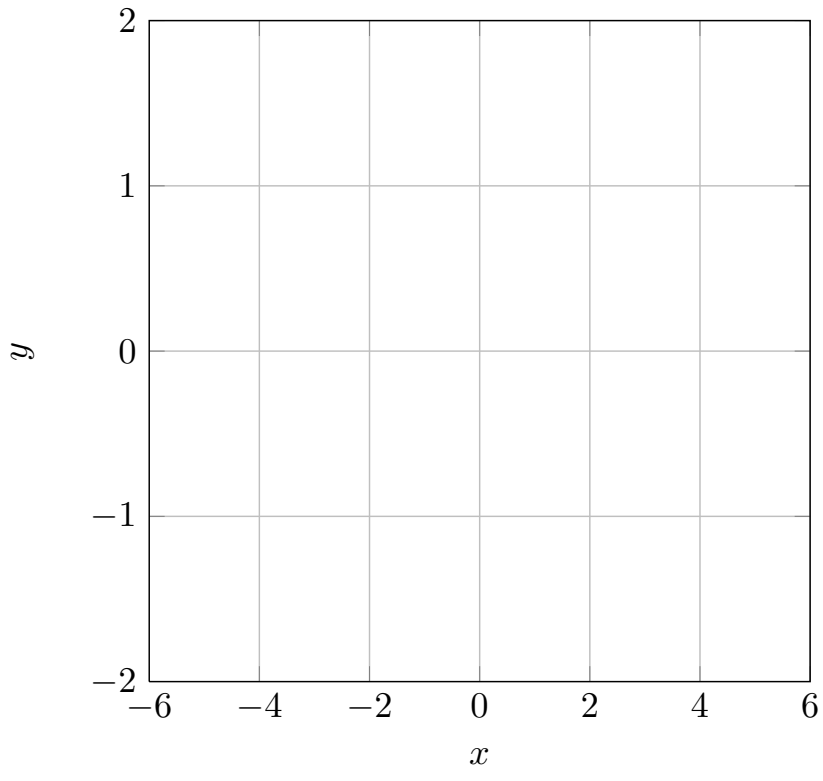
```
% nel preambolo
\usepackage{pgfplots}
...
% nel documento
% il grafico minimo
\begin{tikzpicture}
    \begin{axis}
    \end{axis}
\end{tikzpicture}
```



Personalizzare il grafico

Il linguaggio di **pgfplots** prevede l'efficiente sintassi chiave=valore per la personalizzazione del grafico:

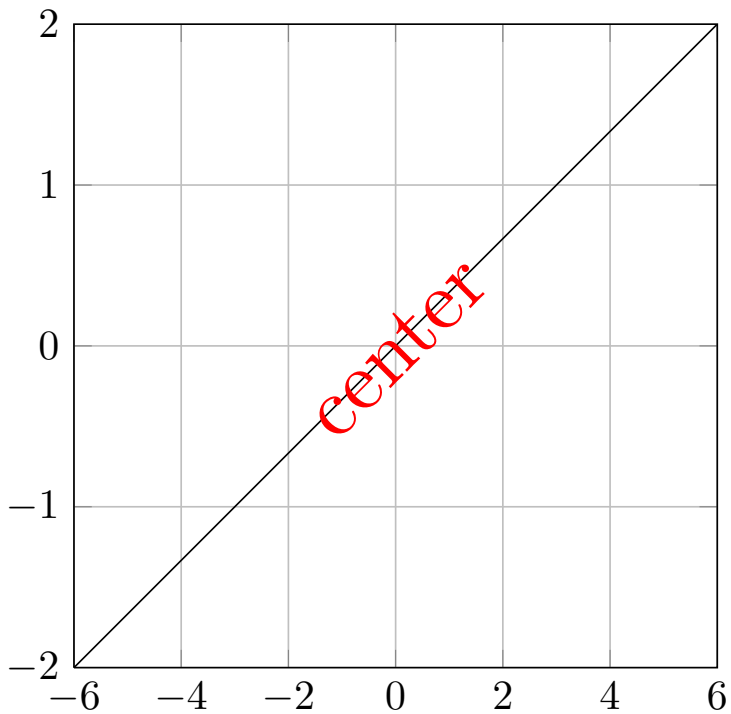
```
\begin{tikzpicture}
\begin{axis}[
    % griglia
    grid=major,
    % limiti assi
    xmin=-6, xmax=6,
    ymin=-2, ymax=2,
    % dimensione tela
    width=8cm, height=8cm,
    % etichette assi
    xlabel=$x$, ylabel=$y$]
\end{axis}
\end{tikzpicture}
```



Disegnare direttamente sulla tela

Disegnare sulla *tela* è una particolarità unica e molto utile di **pgfplots**, per esempio oggetti linee e testi ma ciò si estende a tutte le potenzialità del pacchetto padre **pgf**:

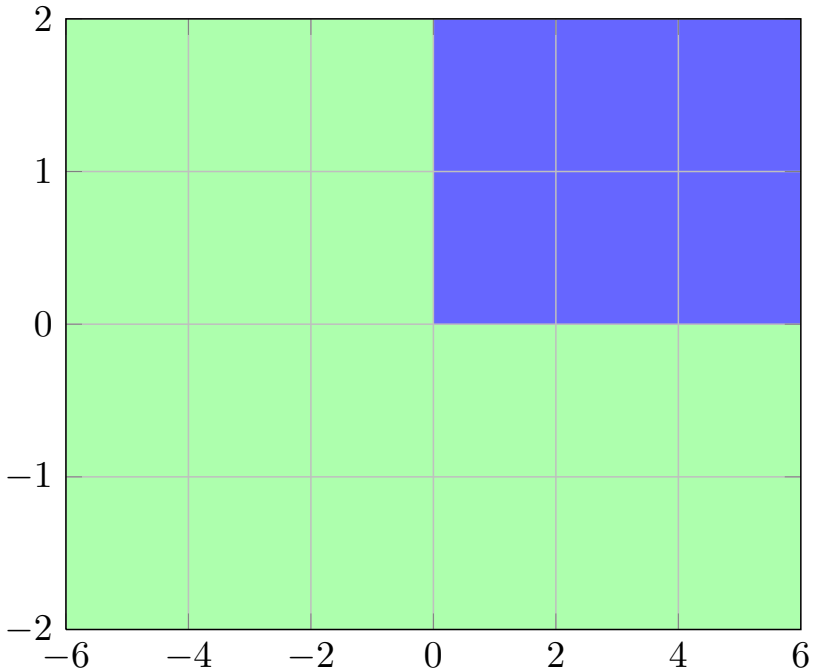
```
\begin{tikzpicture}
\begin{axis}[grid=major,
    xmin=-6, xmax=6, ymin=-2, ymax=2,
    width=7cm, height=7cm,
]
% disegno linee
\draw (rel axis cs:0,0) --
      (rel axis cs:1,1);
% disegno testi
\node[color=red,rotate=45,scale=2] at
     (rel axis cs: 0.5,0.5) {center};
\end{axis}
\end{tikzpicture}
```



Disegnare direttamente sulla tela

Il disegno sulla tela del grafico con comandi **pgf** è facilitato con molti sistemi di riferimento appositamente creati in **pgfplots**:

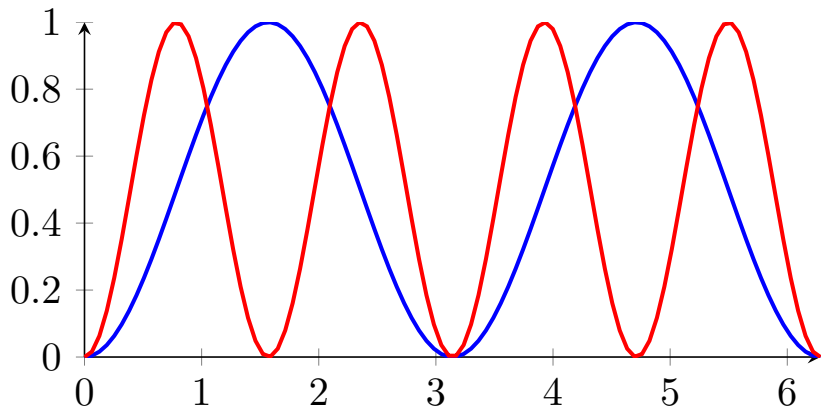
```
\begin{tikzpicture}
\begin{axis}[grid=major,
    xmin=-6, xmax=6, ymin=-2, ymax=2,
    % ordine disegno assi
    axis on top]
% disegno aree di colore
\fill[color=green!32]
  (axis description cs:0,0)
    rectangle
    (axis description cs:1,1);
\fill[color=blue!60]
  (axis cs:0,0) rectangle
    (axis cs:6,2);
\end{axis}
\end{tikzpicture}
```



Tracciamento curve: il comando \addplot

Per disegnare una curva in **pgfplots** si utilizza il comando \addplot all'interno dell'ambiente che definisce il grafico. \addplot accetta espressioni matematiche che valuta numericamente con la libreria interna **pgf**math:

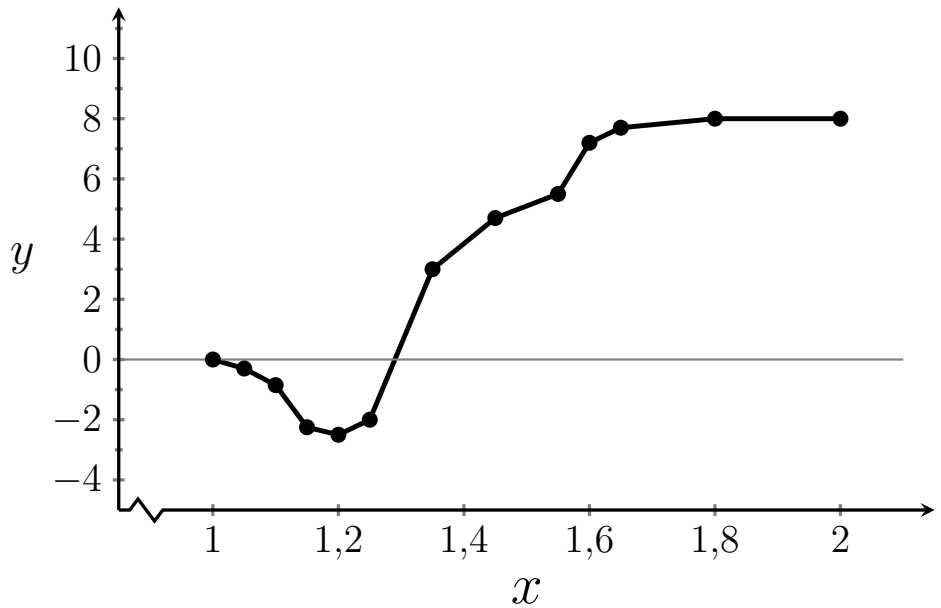
```
\begin{tikzpicture}
\begin{axis}[
tela →  domain=0:2*pi, samples=100,
assi →  axis x line=bottom,
        axis y line=left,
dim →   width=8cm, height=4.5cm,
]
\addplot[color=blue, line width=1pt] {
    sin(deg(x))^2};
\addplot[color=red, line width=1pt] {
    sin(2*deg(x))^2};
\end{axis}
\end{tikzpicture}
```



Tracciamento curve: definizione per coordinate

Oppure si possono definire i punti della curva come lista di coordinate con la parola chiave **coordinates**:

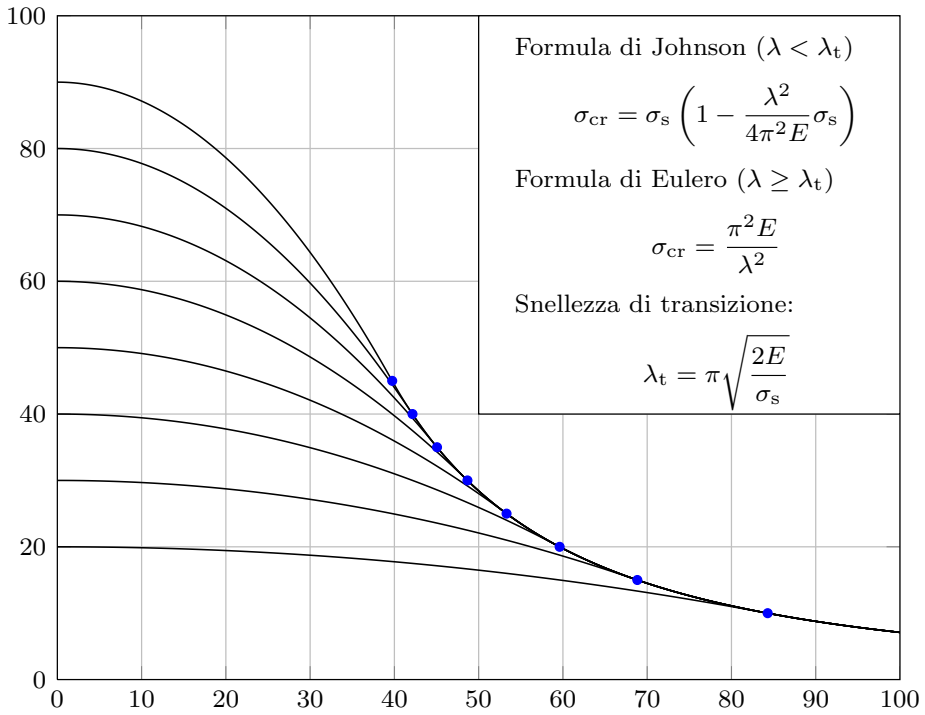
```
\addplot [
    line width=1.8pt, mark=*
    ] coordinates {
    (1.00,   0.00)
    (1.05,  -0.30)
    (1.10,  -0.85)
    (1.15,  -2.25)
    (1.20,  -2.50)
    (1.25,  -2.00)
    (1.35,   3.00)
    (1.45,   4.70)
    (1.55,   5.50)
...
};
```



Tracciamento curve: file numerici

Oppure le coordinate si possono memorizzare in un file testuale in un formato standard e tracciare la relativa curva fornendo ad `\addplot` il nome del file con la parola chiave **file**:

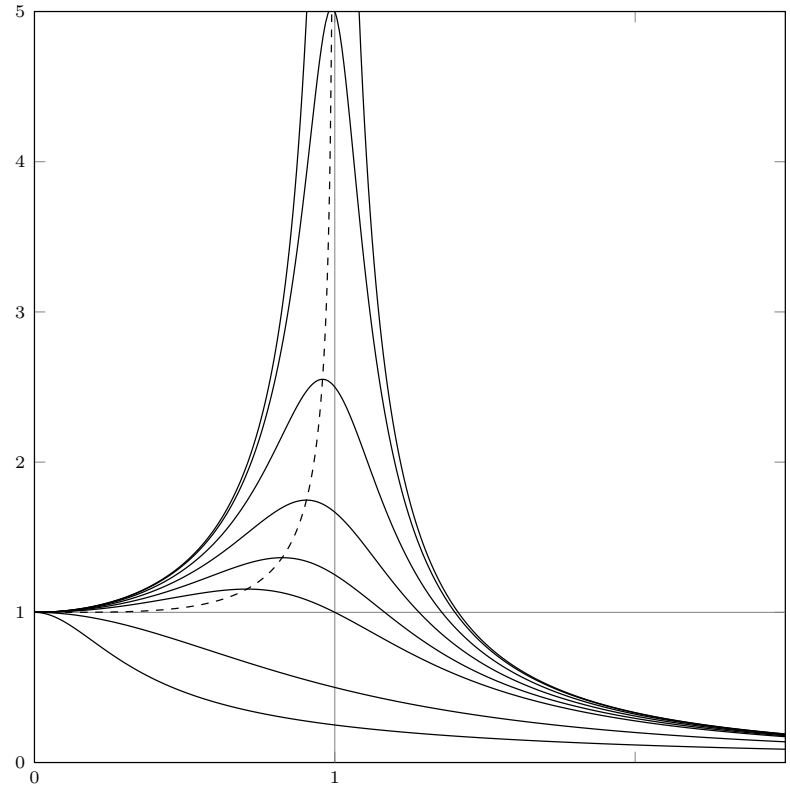
```
\begin{tikzpicture}
\begin{axis}[...]
\foreach \id in {20,30,...,90}
   \addplot[line width=0.5pt]
           file {dati\id.txt};
\addplot[mark=*, mark size=1.4pt,
        only marks,
        draw=blue, fill=blue]
        file {points.txt};
...
\end{axis}
\end{tikzpicture}
```



Famiglia di curve: tracciamento multiplo

L'esempio applicativo comprende il tracciamento di una famiglia di curve. Si definiscono più comandi `\addplot` interni all'ambiente **axis**. I calcoli vengono effettuati con **gnuplot** per maggiore efficienza (occorre l'opzione `-shell-escape` in compilazione).

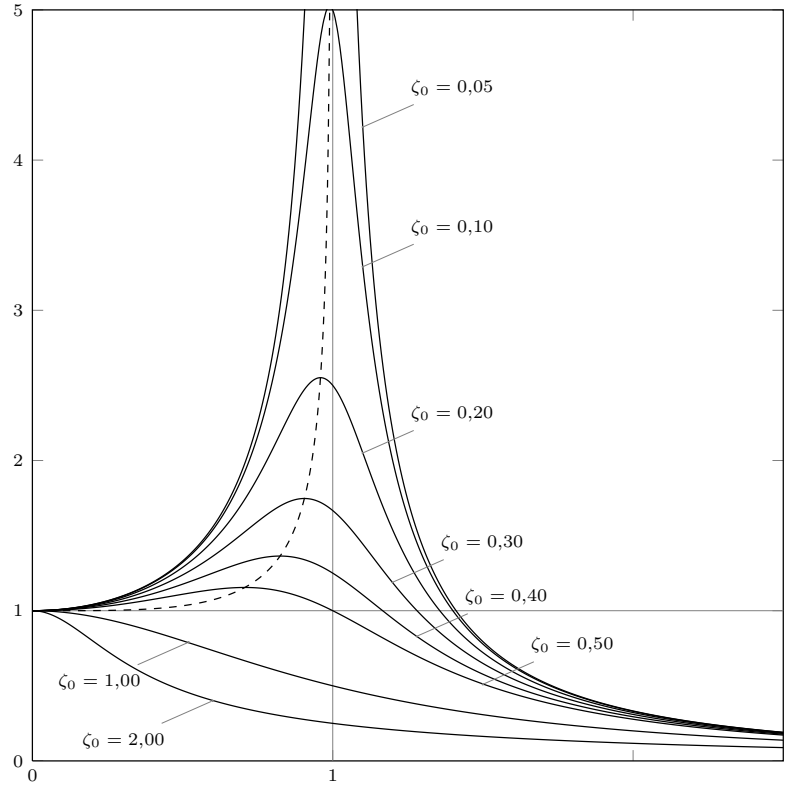
```
% tracciamento curve
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.05^2*x^2)};
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.10^2*x^2)};
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.20^2*x^2)};
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.30^2*x^2)};
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.40^2*x^2)};
\addplot gnuplot
  {1/sqrt((1-x^2)^2+4*0.50^2*x^2)};
```



Famiglia di curve: Etichettare le curve

Per etichettare le curve con il parametro corrispondente possiamo far uso dell'oggetto **pin** del pacchetto padre **pgf** disegnandolo nel punto opportuno con un comando grafico `\node`:

```
% etichette curve
\node[pin=30:{$\zeta_0=0{,}05$}]
   at (axis cs:1.10,4.22) {};
\node[pin=30:{$\zeta_0=0{,}10$}]
   at (axis cs:1.10,3.29) {};
\node[pin=30:{$\zeta_0=0{,}20$}]
   at (axis cs:1.10,2.05) {};
\node[pin=30:{$\zeta_0=0{,}30$}]
   at (axis cs:1.20,1.19) {};
\node[pin=30:{$\zeta_0=0{,}40$}]
   at (axis cs:1.28,0.83) {};
```



Etichettare le curve in automatico: codice

Un elegante modo di etichettare una curva si basa sulla libreria **intersections** di **pgf**. Un nuovo stile esprime l'etichetta a partire dal valore dell'ascissa e dal formato richiesto.

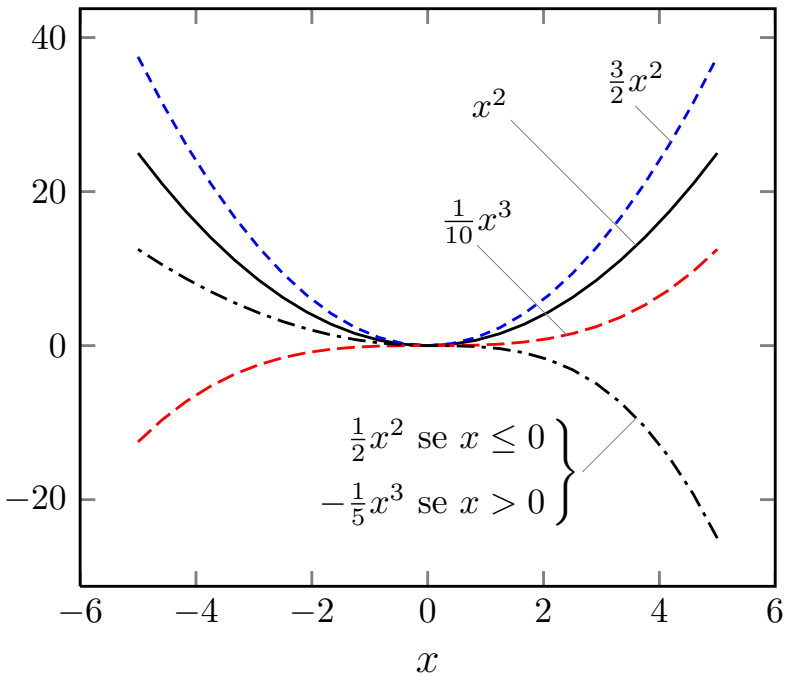
```
\usetikzlibrary{intersections}% nel preambolo
\pgfkeys{
   /pgfplots/linelabel/.style args={#1:#2:#3}{%
     name path global=labelpath, % la curva da etichettare
     execute at end plot={% la linea verticale
        \path [name path global=labelpositionline]
            (rel axis cs:#1,0) -- (rel axis cs:#1,1);
        \draw [help lines,text=black,
            inner sep=0pt,
            name intersections={% intersezione
                of=labelpath and labelpositionline
            }] (intersection-1) -- +(#2)
                node [label={#3}] {}; % etichetta
      }}}
```

http://tex.stackexchange.com/questions/12207/

Etichettare le curve in automatico: esempio

Ecco un esempio di applicazione dello stile di etichettatura:

```
\addplot [thick,
  linelabel=
  0.80:                    % ⟶ ξ
  {-135:0.75cm}:           % ⟶ Δθ, Δρ
  {left:                   % etichetta
  $\left.
  \begin{array}{rl}
  \frac{1}{2}x^2 &\text{se } x \leq 0\\
 -\frac{1}{5}x^3 &\text{se } x > 0
  \end{array}
  \right\}$
  }
]{(x<0)*0.5*x^2+(!(x<0))*(-0.20*x^3)};
```



Piano semilogaritmico: stili di **pgfplots**

Costruzione di un grafico semilogaritmico *passo passo*.

Come in **pgf** anche in **pgfplots** è possibile definire stili da assegnare agli elementi e modificare quelli già esistenti.
Per esempio, modificare lo spessore predefinito delle curve è semplice:

```
\pgfplotsset{%
    every axis plot/.append style={line width=1pt}}
```

Piano semilogaritmico: creare il grafico

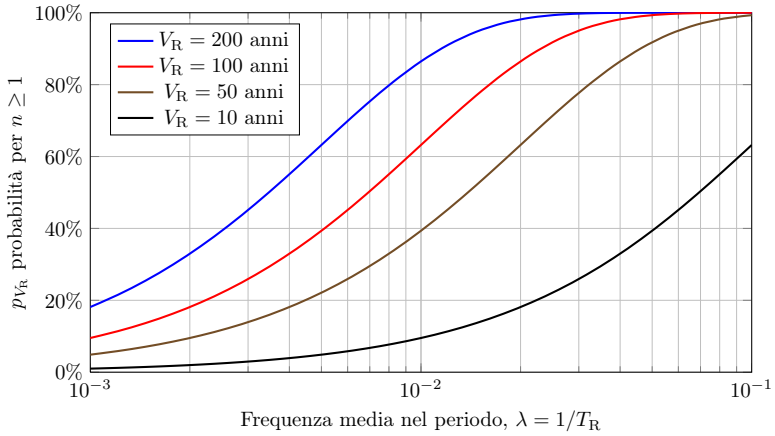
In **pgfplots** sono a disposizione alcuni ambienti oltre **axis** per le altre tipologie di grafico. Al piano logaritmico in ascissa corrisponde l'ambiente **semilogxaxis**.

```
\begin{tikzpicture}
\begin{semilogxaxis}[
    xlabel={Frequenza media nel periodo,
            $\lambda=1/T_\mathrm{R}$},
    ylabel={$p_{V_\mathrm{R}}$ probabilità per $n\geq1$},
    ytick={0,0.2,0.4,0.6,0.8,1},
    yticklabels={0%,20%,40%,60%,80%,100%},
    grid=both, no markers, smooth,
    xmin=0.001, xmax=0.1, ymin=0, ymax=1,
    domain=0.001:0.1,
    width=12.75cm, height=7.65cm,
    legend pos=north west]
...
\end{semilogxaxis}
\end{tikzpicture}
```

Piano semilogaritmico: il risultato finale

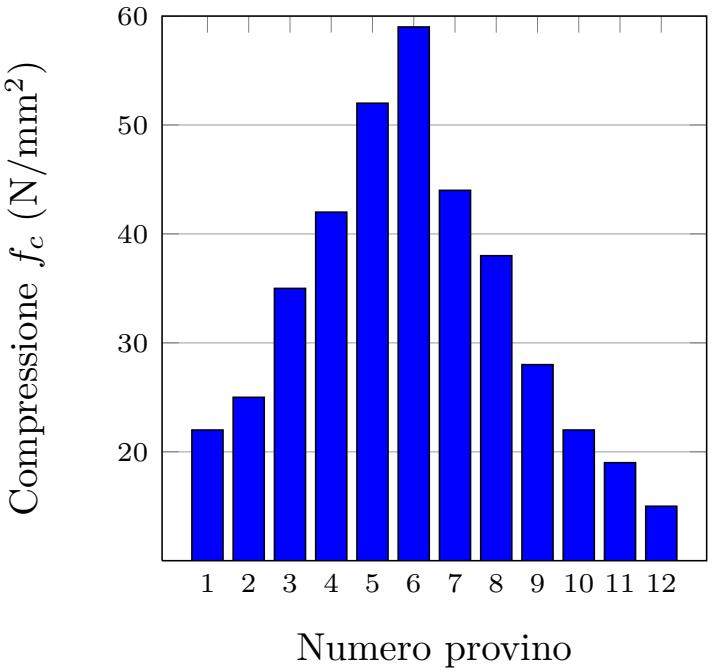
Impostate le proprietà grafiche della tela si tracciano le curve con il comando `\addplot` e si costruisce la legenda con `\addlegendentry`:

```
\addplot gnuplot[id=iv] {1-exp(-x*200)};
\addlegendentry{$V_\mathrm{R}=200$ anni};

\addplot gnuplot[id=iii] {1-exp(-x*100)};
\addlegendentry{$V_\mathrm{R}=100$ anni};

\addplot gnuplot[id=ii] {1-exp(-x*50)};
\addlegendentry{$V_\mathrm{R}=50$ anni};

\addplot gnuplot[id=i] {1-exp(-x*10)};
\addlegendentry{$V_\mathrm{R}=10$ anni};
```

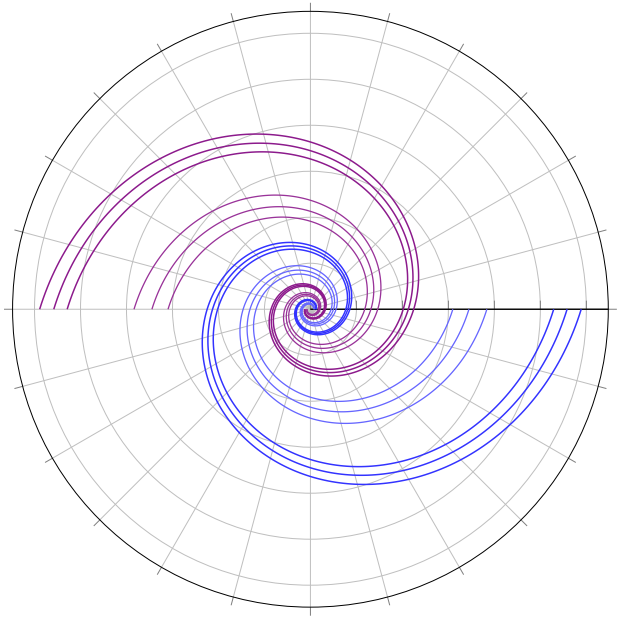


Altri due esempi

Un grafico ad istogramma:

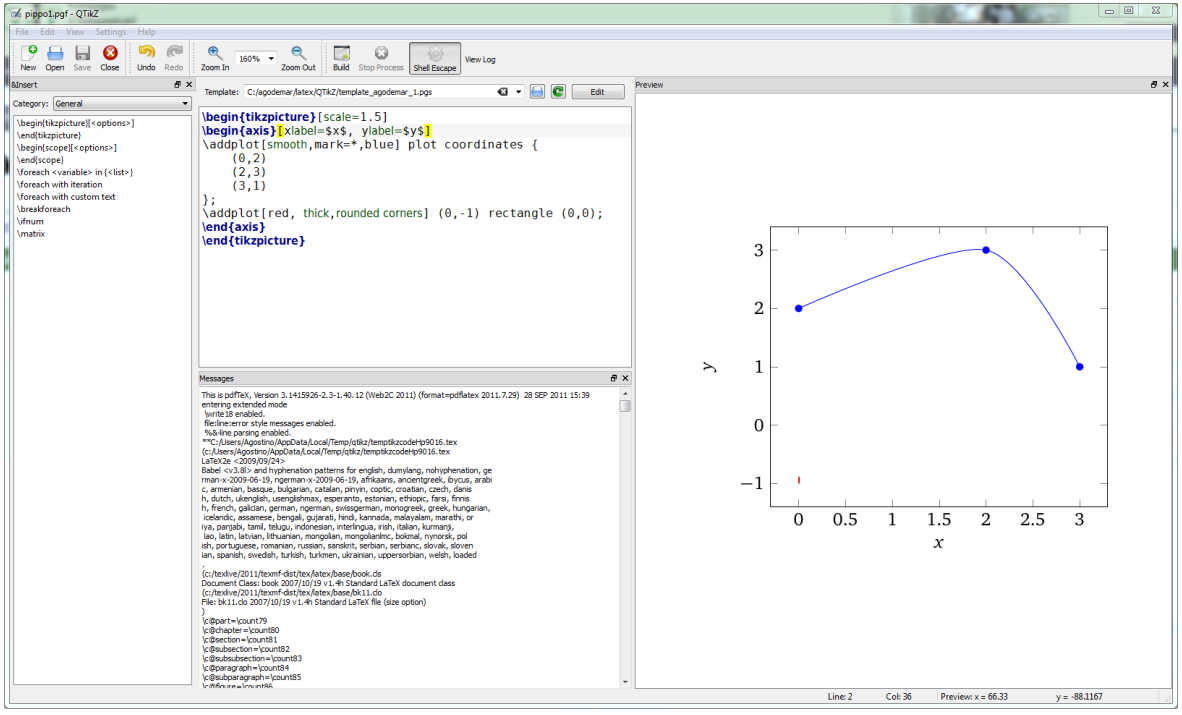


Un grafico polare:



Un utile strumento: QTikz

http://www.hackenberger.at/blog/ktikz-editor-for-the-tikz-language/



Esempio di template:

```
\documentclass{article}
\usepackage{pgfplots}
% eventuali altri pacchetti
% ...
\pagestyle{empty}
\begin{document}
<>
\end{document}
```

Per utenti Windows e Linux.

I comandi e le opzioni di **pgf** vengono suggerite dall'editor.

Output personalizzabile tramite template.

L'opzione di compilazione `-shell-escape` è prevista.

Il pdf risultato della compilazione può essere salvato.

Conclusioni

- **pgfplots** è risultato davvero potente ed in grado di costruire grafici professionali coerenti con la tipografia del documento.
- il pacchetto di Christian Feuersänger dimostra grande precisione grafica e notevole flessibilità nella definizione dei dati da plottare.
- ricordiamo che i sorgenti degli esempi proposti nell'articolo sono disponibili in rete su `github.com`.

Grazie per l'attenzione.