Graphics for LaTeX users

Agostino De Marco
Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Industriale
Gruppo Utilizzatori Italiani di TeX

GUIT meeting 2019
Politecnico di Torino
26 October 2019
Outline

General guidelines on illustration design

Drawing with \LaTeX-aware software
   Using Inkscape + TexText extension

Drawing with natively available \LaTeX environments/packages
   The standard environment picture
   The package pstricks (PostScript)
   The package tikz

Data plots with package pgfplots
Outline

General guidelines on illustration design

Drawing with \LaTeX-aware software
  Using Inkscape + TexText extension

Drawing with natively available \LaTeX environments/packages
  The standard environment picture
  The package \texttt{pstricks} (PostScript)
  The package \texttt{tikz}

Data plots with package \texttt{pgfplots}
The term *illustration* refers to all kind of pictorial graphics — photographs, drawings, diagrams, and schematics.
The term *illustration* refers to all kinds of pictorial graphics — photographs, drawings, diagrams, and schematics.
(a) An example of technical illustration showing the Reflex principle.

(b) A newspaper illustration. This example shows a particular kind of artwork known as ‘infographics.’

Figure 1: Examples of on-the-job technical illustrations.
It is important in typography to maintain a consistency between text and graphics.
**Figure 2:** A technical book in the hands of a reader. The right-hand page contains a full-height annotated illustration.
Illustrations – Benefits

Benefits coming from a careful use of *visual material in technical documents*:

▶ Readers look for and want graphics.
▶ Graphics enhance a communication’s visual appeal, thereby increasing the readers’ concentration on its message.
▶ Well-crafted graphics really can say more than many lines of text, much more efficiently than prose.
▶ Graphics enable writers to convey information to readers who do not share a common language with the writers — or with each other.
▶ Graphics communicate information so effectively that they sometimes convey the entire message (see Figure /one.osfa, Reflex camera).
Benefits coming from a careful use of *visual material in technical documents*:

- **Readers look for and want graphics.**
Illustrations – Benefits

Benefits coming from a careful use of visual material in technical documents:

- **Readers look for and want graphics.**
- Graphics enhance a communication’s visual appeal, thereby increasing the readers’ concentration on its message.

Well-crafted graphics really can say more than many lines of text, much more efficiently than prose. Graphics enable writers to convey information to readers who do not share a common language with the writers — or with each other. Graphics communicate information so effectively that they sometimes convey the entire message (see Figure /one.osfa, Reflex camera).
Illustrations – Benefits

Benefits coming from a careful use of *visual material in technical documents*:

▶ *Readers look for and want graphics.*

▶ Graphics enhance a communication’s *visual appeal*, thereby increasing the readers’ concentration on its message.

▶ *Well-crafted graphics really can say more than many lines of text,* much more efficiently than prose.
Benefits coming from a careful use of visual material in technical documents:

- Readers look for and want graphics.
- Graphics enhance a communication’s visual appeal, thereby increasing the readers’ concentration on its message.
- Well-crafted graphics really can say more than many lines of text, much more efficiently than prose.
- Graphics enable writers to convey information to readers who do not share a common language with the writers—or with each other.
Benefits coming from a careful use of visual material in technical documents:

- Readers look for and want graphics.
- Graphics enhance a communication’s visual appeal, thereby increasing the readers’ concentration on its message.
- Well-crafted graphics really can say more than many lines of text, much more efficiently than prose.
- Graphics enable writers to convey information to readers who do not share a common language with the writers—or with each other.
- Graphics communicate information so effectively that they sometimes convey the entire message (see Figure 1a, Reflex camera).
Keep in mind that, at some point, readers’ attention will be going back and forth between text and figures, necessarily.
Keep in mind that, at some point, readers’ attention will be going back and forth between text and figures, necessarily.

Make the effort of having the readers feel at ease during the process.
Keep in mind that, at some point, readers’ attention will be going back and forth between text and figures, necessarily.

Make the effort of having the readers feel at ease during the process.

Design graphics with a special focus on usability.
Keep in mind that, at some point, readers’ attention will be going back and forth between text and figures, necessarily.

Make the effort of having the readers feel at ease during the process.

Design graphics with a special focus on usability.

Graphics should have the same good qualities of author’s prose, easy for readers to understand and use.
Le equazioni (17.13)-(17.14)-(17.15) formano il sistema

\[
\begin{align*}
m \ddot{V} &= - [D - T \cos (\alpha_0 + \mu_T)] \cos \beta + Y_\alpha \sin \beta \\
m \ddot{\dot{V}} \cos \nu &= mg \sin \nu + [D - T \cos (\alpha_0 + \mu_T)] \sin \beta + Y_\alpha \cos \beta \\
m \ddot{\dot{V}} \sin \nu &= - mg \cos \nu + L + T \sin (\alpha_0 + \mu_T)
\end{align*}
\]

di equazioni del moto di virata a quota costante. La seconda e terza delle equazioni (17.16) possono essere ulteriormente manipolate per arrivare ad una conveniente forma finale. Ad esempio, moltiplicando la seconda per \(\sin \nu\), la terza per \(\cos \nu\) e sommando membro a membro si ottiene una prima equazione alternativa in cui non compaiono termini contenenti il peso \(W^* = mg\). Inoltre, moltiplicando la seconda per \(\sin \nu\), la terza per \(\cos \nu\) e dividendo membro a membro si ottiene un primo membro unitario; moltiplicando per il denominatore del secondo membro si ottiene una seconda equazione alternativa. Dopo alcune fattorizzazioni atte a far comparire nelle espressioni i tipici rapporti \(T/W\), \(W/S\) e i coefficienti aerodinamici, si arriva pertanto alla forma definitiva

\[
\begin{align*}
\frac{\ddot{V}}{T} &= \cos (\alpha_0 + \mu_T) \cos \beta - \frac{\ddot{\dot{V}}}{W/S} (C_D \cos \beta - C_F \sin \beta) \\
\frac{\ddot{\dot{V}}}{T} &= \frac{\cos (\alpha_0 + \mu_T)}{W} \sin \nu - \sin (\alpha_0 + \mu_T) \sin \beta + \frac{\ddot{\dot{V}}}{W/S} (C_D \sin \beta + C_F \cos \beta) \sin \nu + C_L \sin \nu \\
\frac{1}{T} &= \frac{\cos (\alpha_0 + \mu_T)}{W} \sin \nu + \sin (\alpha_0 + \mu_T) \cos \nu - \frac{\ddot{\dot{V}}}{W/S} (C_D \sin \beta + C_F \cos \beta) \sin \nu - C_L \cos \nu
\end{align*}
\]

(17.17)

I coefficienti aerodinamici che figurano nelle equazioni della virata (17.17) risultano espressi dalle

\[
C_D = \frac{D}{\frac{1}{2} \rho V^2 S} = C_{D_0} + k C_L^n
\]

(7.53)

\[
C_L = \frac{L}{\frac{1}{2} \rho V^2 S} = C_{L_0} + C_{L_\alpha} \alpha + \left( C_{L_\beta} \beta + C_{L_\delta} \delta \right) \frac{\theta}{\alpha} + C_{L_\rho} \rho
\]

(7.56)

\[
C_F = \frac{C}{\frac{1}{2} \rho V^2 S} = C_{F_0} + C_{F_\beta} \beta + C_{F_\delta} \delta, + \left( C_{F_\beta} \beta + C_{F_\delta} \delta \right) \frac{\theta}{\alpha} + C_{F_\rho} \rho, \frac{h}{\alpha}
\]

(17.18)

Nelle (7.53)-(7.56)-(17.18) le grandezze \((p, q, r)\) vanno interpretate in generale come le componenti di \(\Omega_3\) nel sistema di assi in cui sono espresse le equazioni del moto. Per il moto in esame è opportuno operare le sostituzioni

\[
p = p^* = \Omega_{3x}, \quad q = q^* = \Omega_{3y}, \quad r = r^* = \Omega_{3z}
\]

(17.19)

che in base alle (17.12) forniscono

**Figure 3**: Aerospace engineering textbook.

http://wpng.unina.it/agodemar/DSV-DQV/DSV-DQV_Quaderno_17.pdf
Illustrations – Usability rules

Design to **support any possible readers’ tasks**. Imagine your readers in the act of using your graphic material.
Design to **support any possible readers’ tasks**. Imagine your readers in the act of using your graphic material.

**Consider carefully your readers’ knowledge and expectations.** Specialized graphics as opposed to simplified visuals (‘infographics’).
Illustrations – Usability rules

Design to **support any possible readers’ tasks**. Imagine your readers in the act of using your graphic material.

*Consider carefully your readers’ knowledge and expectations.* Specialized graphics as opposed to simplified visuals (‘infographics’).

*Seek for simplicity.* Especially for graphics that will be read on a computer screen or from a projected image. To keep it simple:
  - Include only a manageable amount of material.
  - Eliminate unnecessary details.
Design to **support any possible readers’ tasks**. Imagine your readers in the act of using your graphic material.

**Consider carefully your readers’ knowledge and expectations.** Specialized graphics as opposed to simplified visuals (‘infographics’).

**Seek for simplicity.** Especially for graphics that will be read on a computer screen or from a projected image. To keep it simple:

- Include only a manageable amount of material.
- Eliminate unnecessary details.

**Seek for the effectiveness of textual labels.** Important content should always be labelled clearly.
Design to **support any possible readers’ tasks**. Imagine your readers in the act of using your graphic material.

**Consider carefully your readers’ knowledge and expectations.** Specialized graphics as opposed to simplified visuals (‘infographics’).

**Seek for simplicity.** Especially for graphics that will be read on a computer screen or from a projected image. To keep it simple:
- Include only a manageable amount of material.
- Eliminate unnecessary details.

**Seek for the effectiveness of textual labels.** Important content should always be labelled clearly.

**Choose effective informative titles** (figure and table captions). Possibly, make them brief and informative at the same time.
Outline

General guidelines on illustration design

Drawing with \LaTeX-aware software

Using Inkscape + TexText extension

Drawing with natively available \LaTeX environments/packages

The standard environment picture
The package pstricks (PostScript)
The package tikz

Data plots with package pgfplots
1.7 Assi vento

(Wind Axes, W)

La terza assi vento $\mathbf{T}_W$ è una terza triangolare levogira con origine nel baricentro del velivolo (punto $G = C$) e avente l’asse vento longitudinale $\mathbf{T}_W$ diretto secondo la direzione della velocità $\mathbf{V} = V_W$ del velivolo, con verso positivo nello senso del moto (figura 1.13). L’asse vento $\mathbf{T}_W$ è definito dall’interazione del piano verticale $\mathbf{x}_V$, contenente $\mathbf{V}$ e $\mathbf{G}$ con il piano $\mathbf{x}_W$, normale alla traiettoria in $\mathbf{G}$, con verso positivo verso il basso. L’asse trasversale $\mathbf{T}_W$ è tale da completare la terza $\{\mathbf{G}, \mathbf{x}_V, \mathbf{y}_W, \mathbf{z}_W\}$.

Figura 1.13: Terna di assi vento $\mathbf{T}_W = \{\mathbf{G}, \mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W\}$ (o assi traiettoria). In questa particolare circostanza la traiettoria del baricentro è orizzontale e l’orientamento del velivolo non è simmetrico rispetto al piano verticale $\mathbf{x}_W$. Si noti che, secondo la definizione data sopra, l’asse vento trasversale $\mathbf{T}_W$ è sempre orizzontale. Esso è la normale al piano $\{\mathbf{G}, \mathbf{x}_W, \mathbf{z}_W\}$ che è, per definizione, costantemente verticale come si osserva dalle figure 1.13, 1.14 e 1.15.

Nella figura 1.13 è rappresentato il caso particolare in cui la traiettoria del baricentro è orizzontale. Si osserva che, anche per un orientamento del velivolo non simmetrico rispetto al piano verticale $\mathbf{x}_W$, in questa circostanza gli assi vento presentano un asse $\mathbf{z}_W$ verticale (allineato con la forza peso $mg$).

Nella figura 1.14, è rappresentata un’evoluzione in cui la traiettoria del baricentro si incassa e allo stesso tempo l’orientamento del velivolo è non simmetrico rispetto al piano verticale $\mathbf{x}_W$. L’asse vento $\mathbf{T}_W$, per definizione tangente alla traiettoria, è non orizzontale e l’asse $\mathbf{z}_W$ è verticale. Naturalmente — come da definizione — anche in questo caso, in cui il piano $\mathbf{x}_W$ non è verticale, l’asse vento $\mathbf{T}_W$ è orizzontale.

Tale particolarità evidenzia un’importante differenza tra le definizioni di asse vento che si ritrovano in diversi libri di testo americani e anglosassoni. Secondo questi ultimi [17, 9, 60] l’asse longitudinale coincide con $x_W$; il

Figure 4: Aerospace engineering textbook.

http://wpage.unina.it/agodemar/DSV-DQV/DSV-DQV_Quaderno_1.pdf
Inkscape is an open source and well-supported vector graphics/SVG editor available for all major operating systems.

Provides effective \LaTeX-related capabilities, e.g. the TexText Python-based plugin extension.

TexText provides the possibility to add and re-edit (multi-line) \LaTeX/Xe\LaTeX/Lua\LaTeX generated SVG elements to a drawing.
Mathematical abstraction using the example of generation of periodic oscillations

\[ \left\{ y : V(y) = \text{const} \right\} \]

\[ \left\{ y : W(y) = \alpha^* \right\} \]

Desired periodic orbit

**Figure 5:** A screenshot of Inkscape with TexText extension in use.
(a) Selecting TexText from Inkscape Extensions menu.

(b) The TexText dialog window.

Figure 6: Using TexText extension plugin in Inkscape.
TexText is a very powerful tool when coupled with the potential of Inkscape itself. Yet the user have to be aware of including the required packages in the preamble file if special commands are used in LaTeX code that rely on such packages. The preamble file can be chosen by the selector mentioned above. The default preamble file shipped with TexText is the following:

```
\usepackage{amsmath,amsthm,amssymb, amsfonts}
\usepackage{color}
```

Basically, user's LaTeX code will be inserted into this template:

```
\documentclass{article}
% ===> preamble file content <===
% default:
% \input{default_packages}
\pagestyle{empty}
\begin{document}
% ==> User's code <===
\end{document}
```

This will be typeset in a separate system thread, the PDF result will be converted to SVG and the vector object will be inserted into the current Inkscape document.
The final SVG object is re-editable via the TexText dialog!

Figure 7: SVG element resulting from user’s input compilation (see Figure 6b).

Inkscape

A. De Marco | Graphics for LaTeX users | GIT meeting 2019
Demo
Outline

General guidelines on illustration design

Drawing with \LaTeX-aware software
  Using Inkscape + TexText extension

Drawing with natively available \LaTeX environments/packages
  The standard environment picture
  The package pstricks (PostScript)
  The package tikz

Data plots with package pgfplots
Making drawings with code

A completely different paradigm.

No pseudo-synchronous visual tools.

Similar to the asynchronous typesetting workflow.
The native environment **picture**

\begin{picture}(120,80)
  \put(30,30){\circle*{3}}
  \put(30,33){\makebox(0,0)[br]{$A$}}
  \put(90,43){\circle*{3}}
  \put(88,47){\makebox(0,0)[bl]{$B$}}
  \linethickness{1.2pt}
  \Line(30,30)(90,43)
  \put(10,10){\vector(1,0){100}}
  \put(110,14){\makebox(0,0)[b]{$x$}}
  \put(10,10){\vector(0,1){60}}
  \put(14,70){\makebox(0,0)[l]{$y$}}
  % dashed box
  \put(0,0){\dashbox{5}(120,80){}}
\end{picture}

**Figure 8:** A drawing made with the standard picture environment enhanced by the pict2e package.
Figure 9: Placing whatever, wherever in a `pspicture` environment.
Drawing with packages `pgf` and `tikz` by Till Tantau ...

https://www.ctan.org/pkg/pgf


https://pgf-tikz.github.io/ (manual on the web)
The name **pgf** means *Portable Graphics Format*.

It is a package for creating **inline graphics**: defines a number of TeX commands that can draw graphics within the typesetting process.

**Graphics objects are put into boxes** and treated as normal items to be taken care of by the LaTeX output routine.

The package `pgf` exposes a **frontend layer**, i.e. a set of commands or a special syntax that makes using the functionalities implemented by basic layer easier.

This frontend is what is called **TIKZ**, the LaTeX package `tikz` that incorporates `pgf`.

The name **TIKZ** is an acronym of *Tikz Ist Kein Zeichenprogramm* (German for ‘tikz is not a drawing program’).
In preamble: \texttt{\usepackage{tikz}}

The package provides the command \texttt{\tikz} as in the following examples.

\begin{verbatim}
\tikz \draw (0pt,0pt) -- (20pt,6pt);
\end{verbatim}

yields the line $\rightarrow$, or

\begin{verbatim}
\tikz \fill[color=orange] (1ex,1ex) circle(1ex);
\end{verbatim}

yields the orange circle $\bullet$.

The argument passed to \texttt{\tikz} is a semicolon-terminated string.
More elaborate drawings are embedded into the environment `tikzpicture`:

\begin{tikzpicture}
\draw (0,0) -- (1,0) -- (1,1) -- cycle;
\end{tikzpicture}

\begin{tikzpicture}
\draw (0,0) rectangle (2,1);
\draw (0,0) -- (2,1);
\draw (0,1) -- (2,0);
\end{tikzpicture}
A \texttt{tikzpicture} can be used \emph{inline} with the running text of a paragraph, like any other box object:

The following draws a $0.4 \times 0.2$ crossed rectangle:

\begin{tikzpicture}
  \draw (0.0,0.0) rectangle (0.4,0.2);
  \draw (0.0,0.0) -- (0.4,0.2);
  \draw (0.0,0.2) -- (0.4,0.0);
\end{tikzpicture}

The following draws a $0.4 \times 0.2$ crossed rectangle: $
\begin{tikzpicture}
  \draw (0.0,0.0) rectangle (0.4,0.2);
  \draw (0.0,0.0) -- (0.4,0.2);
  \draw (0.0,0.2) -- (0.4,0.0);
\end{tikzpicture}
$.
Inside a `tikzpicture` environment *everything is drawn by starting a path and by extending the path*. Paths are constructed using the `\path` command.

```latex
\begin{tikzpicture}
  \path[draw] (0,0) -- (1,1);
  \path[draw] (1,0) -- (2,0);
\end{tikzpicture}
```
Inside a `tikzpicture` environment *everything is drawn by starting a path and by extending the path*. Paths are constructed using the `\path` command.

\begin{tikzpicture}
\path[draw] (0,0) -- (1,1);
\path[draw] (1,0) -- (2,0);
\end{tikzpicture}
Inside a `tikzpicture` environment *everything is drawn by starting a path and by extending the path*. Paths are constructed using the `\path` command.

\begin{tikzpicture}
  \path[draw] (0,0) -- (1,1);
  \path[draw] (1,0) -- (2,0);
\end{tikzpicture}
Inside a `tikzpicture` environment *everything is drawn by starting a path and by extending the path*. Paths are constructed using the `\path` command.

```
\begin{tikzpicture}
  \path[draw] (0,0) -- (1,1);
  \path[draw] (1,0) -- (2,0);
\end{tikzpicture}
```
Inside a `tikzpicture` environment *everything is drawn by starting a path and by extending the path*. Paths are constructed using the \path command.

\begin{tikzpicture}
  \path[draw] (0,0) -- (1,1);
  \path[draw] (1,0) -- (2,0);
\end{tikzpicture}
Line-to and move-to operations

Command \texttt{\textbackslash{}draw} stands for \texttt{\textbackslash{}path[draw]}:

\begin{verbatim}
\texttt{\textbackslash{}path[draw]} (0,0) -- (1,1);  \\
\texttt{\textbackslash{}path[draw]} (1,0) -- (2,0);
\end{verbatim}

\begin{verbatim}
\texttt{\textbackslash{}draw} (0,0) -- (1,1);  \\
\texttt{\textbackslash{}draw} (1,0) -- (2,0);
\end{verbatim}

Multiple paths can be traced with one single command:

\begin{verbatim}
\texttt{\textbackslash{}draw} (0,0) -- (1,1)  \\
\hspace{1cm}\texttt{\textbackslash{}draw} (1,0) -- (2,0); \hspace{1cm}\%	extit{move-to} operation
\end{verbatim}
A **grid** is a path extension operation between two coordinates, much like a line (\(\--\)):

```latex
\begin{verbatim}
% fine, thin grid
\draw[line width=0.1pt,gray!30,step=5mm]
  (0,0) grid (3,2);
% coarse, thicker grid
\draw[help lines]
  (0,0) grid (3,2);
% a thick, closed path
\draw[thick] (1,1) -- (2,2) -- (2,1)
  -- cycle;
\end{verbatim}
```

The **cycle** operation closes a path connecting the current point with the initial point on the path.
More line-to and move-to operations

\begin{verbatim}
\draw[help lines] (0,0) grid (3,2);
\draw[thick] (0,0) -- (1,1) -- (2,1) -- (3,2);
\end{verbatim}

\begin{verbatim}
\draw[help lines] (0,0) grid (3,2);
\draw[thick] (0,0) -- (1,1) -- (2,0) -- (3,2);
\end{verbatim}

\begin{verbatim}
\draw[help lines] (0,0) grid (3,2);
\draw[thick] (0,0) -- (1,1) -- (2,0) -- (3,0) -- (3,1) -- cycle;
\end{verbatim}
More *line-to operations*

\begin{verbatim}
\draw[thick] (0.0,0.0) -| (2.0,0.5)
(1.0,1.0) -| (3.0,0.0);
\end{verbatim}

\begin{verbatim}
\draw[thick] (0.0,0.0) |- (2.0,1.0)
(1.0,0.5) |- (3.0,0.0);
\end{verbatim}
The curve-to operation

% the supporting grid
\draw[help lines] (-2,-4) grid (2,4);
% define labels (nodes)
\path (-2, 0) coordinate (c1)
   (-1, 3) coordinate (c2)
   ( 0,-3) coordinate (c3)
   ( 2,-1) coordinate (c4);
% segments connecting nodes
\draw[dashed] (c1) -- (c2) -- (c3) -- (c4);
% control points
\draw (c1) circle (2pt)
   (c2) circle (2pt)
   (c3) circle (2pt)
   (c4) circle (2pt);
% the Bézier curve
\draw[thick] (c1) .. controls (c2) .. (c3) .. (c4);
% text labels
\path
   (c1) node[anchor=west] {\texttt{c1}}
   (c2) node[anchor=west] {\texttt{c2}}
   (c3) node[anchor=east] {\texttt{c3}}
   (c4) node[anchor=east] {\texttt{c4}};
The Rectangle and Circle operations

\texttt{\textbackslash draw[thick] (0,0) rectangle (1,1)\linebreak rectangle (3,2);}

\texttt{\textbackslash draw[thick] (0,0) rectangle (1,1);\linebreak \textbackslash draw[ultra thick,red] (1,1)\linebreak rectangle (3,2);}

\texttt{\textbackslash draw[thick] (0,0) circle (2pt)\linebreak rectangle (3,1)\linebreak circle (4pt);}
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,4);
\draw[thick] (0,0) circle (2pt)
    -- (1,1) rectangle (2,3)
    -- (3,4)
    -- (2,4) circle (2pt);
\end{tikzpicture}
The *ellipse* operation

\texttt{\textbackslash draw[help lines] (0,0) grid (10,4);}
\texttt{\textbackslash draw (2,2) ellipse (1cm and 1cm)}
\texttt{(3,2) ellipse (3cm and 2cm);}
\texttt{\textbackslash draw[red] (8,2) ellipse (1cm and 1cm);}
The *node* operation

You can add text, math, and other material to paths with the *node path extension operation*.

The node operation

▶ places a given textual content at the current position;
▶ the current position becomes a node in the path;
▶ a label (variable name) can be associated to the node;
▶ named nodes can be used in further drawing operations;

Each node added to a path has an *outer shape*. The outer shape is only drawn if `draw` is part of the options. *The default node shape is a rectangle.*
The *node* operation

\begin{verbatim}
% in preamble: \usetikzlibrary{calc,positioning}
\path[draw] (0,1) node (p1) [draw=none] {$P_1$};\path[draw,fill] (1,0) circle (2pt) node (p2) [anchor=north east] {$P_2$};\path (2,1) coordinate (p3);\path[draw] (p3) circle (10pt) node[draw=none] {$P_3$};\path (3,0) coordinate (p4);\draw[thick,-] (p2) -- (p4);\draw[thick,dashed,red] (p2.north east) -- (p4);\path (4,1) coordinate (p5);\path[draw=none] (p5) circle (8pt) node[draw,red] {box};\node[right=2.0cm of p5, anchor=center, inner sep=0pt, shape=circle, draw=red] (p6) {box};\node[below right=1.0cm and 2.0cm of p6.center, anchor=center, inner sep=2pt, shape=circle, draw=red] (p7) {box};\node[above right=1.0cm and 2.0cm of p7.center, anchor=center, inner sep=2pt, draw=red, fill=red, fill=white] (p7) {\textbf{box}};\end{verbatim}
Placing textual labels

\begin{tikzpicture}
  \draw (0,0) node (hello) {
    [scale=2.0, inner sep=0pt, outer sep=0pt, draw=red]
    \fbox{\textbf{Hello \texttt{GuIT}}};
  }
  \draw (hello.north east) circle (2pt) node[anchor=south west] {north east};
  \draw (hello.north) circle (2pt) node[anchor=south] {north};
  \draw (hello.north west) circle (2pt) node[anchor=south east] {north west};
  \draw (hello.west) circle (2pt) node[anchor=east] {west};
  \draw (hello.south west) circle (2pt) node[anchor=north east] {south west};
  \draw (hello.south) circle (2pt) node[anchor=north] {south};
  \draw (hello.south east) circle (2pt) node[anchor=north west] {south east};
  \draw (hello.east) circle (2pt) node[anchor=west] {east};
\end{tikzpicture}
The **arc operation**

The arc operation adds an arc to the path.

- The arc starts at the current point, $P_1$. The user supplies two angles, $\alpha$ and $\beta$, and a radius $r$.

- The centre of the circle, $C$, is determined by the equation

  $$P_1 = C + (r \cos \alpha, r \sin \alpha)$$

  The end point of the arc is given by $P_2 = C + (r \cos \beta, r \sin \beta)$.

- The arc is drawn in counterclockwise direction from the start point to the end point, which becomes the new current coordinate of the path.
The *arc* operation

\begin{verbatim}
\draw[dashed] (4,0) coordinate (p0) arc (0:180:2cm); % (\alpha, \beta, r)
\draw[fill=black] (2,0) coordinate (c) % ← C
circle (1pt) node[anchor=south east] {$C$};
\path (p0) arc (0:30:2cm) % (\alpha, \beta, r), no arc drawn
coordinate (p30); % ← $P_1$
\draw[fill=black] (p30) circle (1pt) node[anchor=south west] {$P_1$};
\draw[thick] (p30) arc (30:120:2cm) % (\alpha, \beta, r)
coordinate (p120) % ← $P_2$
circle (2pt) node[anchor=north west] {$P_2$};
\draw[->,thick] (c) -- node[anchor=south east] {$r$} (p30); % ← $\vec{r}$
\end{verbatim}
\texttt{\texttt{\textbackslash{draw}[dashed]} (1,1) \texttt{circle} (1cm);  \\
\texttt{\texttt{\textbackslash{draw}} (1,2) \texttt{coordinate} (a) \texttt{circle} (2pt)  \\
\hspace{1em} (2,1) \texttt{coordinate} (b) \texttt{circle} (3pt)  \\
\hspace{2em} (1,0) \texttt{coordinate} (c) \texttt{circle} (4pt);  \\
\texttt{\texttt{\textbackslash{draw}[->,thick]} (a) \texttt{arc} (90:180:1cm);  \\
\texttt{\texttt{\textbackslash{draw}[->,thick]} (b) \texttt{arc} (0:45:1cm);  \\
\texttt{\texttt{\textbackslash{draw}[->,thick]} (c) \texttt{arc} (270:225:1cm);}}

\texttt{\texttt{\textbackslash{draw}[dashed]} (1.5,0) \texttt{circle} (1.5cm and 1cm);  \\
\texttt{\texttt{\textbackslash{draw}[fill=black]} (1.5,0) \texttt{coordinate} (c) \texttt{circle} (1pt);  \\
\texttt{\texttt{\textbackslash{draw}} (3,0) \texttt{coordinate} (a) \texttt{circle} (2pt);  \\
\texttt{\texttt{\textbackslash{draw}} (0,0) \texttt{coordinate} (b) \texttt{circle} (2pt);  \\
\texttt{\texttt{\textbackslash{draw}[->,thick]} (a) \texttt{arc} (0:90:1.5cm and 1cm);  \\
\texttt{\texttt{\textbackslash{draw}[->,thick]} (b) \texttt{arc} (180:340:1.5cm and 1cm);}}
What else?

▶ More actions on paths, e.g. line widths, dash patterns, coloring, filling, shading.
▶ predefined styles of graphic elements and their customizations.
▶ Available coordinate systems and advanced coordinate calculations.

Please have a look at the article on *ArsTEXnica* for more details on *tikz*:

*De Marco, A. “Graphics for \LaTeX users”.*

*ArsTEXnica 28 (October 2019), pp. 64–100.*

All *tikz* examples given in the article are viewable on Overleaf:

[https://www.overleaf.com/read/mgskyfdptttzt](https://www.overleaf.com/read/mgskyfdptttzt)
Outline

General guidelines on illustration design

Drawing with \LaTeX-aware software
   Using Inkscape + TexText extension

Drawing with natively available \LaTeX environments/packages
   The standard environment picture
   The package \pstricks (PostScript)
   The package \tikz

Data plots with package \pgfplots
The package \texttt{pgfplots} is built on top of \texttt{pgf} and is designed to draw graphs in a variety of formats, with a consistent and professional look and feel.

The package also allows to import data stored in files in tabular format via the package \texttt{pgfplotstable}.

As is usual with the \texttt{pgf} family, their manuals are impressive.

https://www.ctan.org/pkg/pgfplots
The workhorse of the `pgfplots` package is an environment called `axis`, which may define one or several plots (2D and 3D).

Each plot is drawn with the command `\addplot`.

The `axis` environment is used inside a `tikzpicture` environment.

Typically, one or more plots are created in LaTeX as follows:

```latex
% in preamble
\usepackage{pgfplots}% loads tikz
...
\begin{tikzpicture}
  \begin{axis}[^⟨graphic options⟩]
  ... 
  ⟨pgfplots or tikz commands⟩
  ...
  \end{axis}
\end{tikzpicture}
```
The axis environment

The simplest possible graph with pgfplots:

\begin{tikzpicture}
\begin{axis}
\end{axis}
\end{tikzpicture}

An empty axis environment, with customized formatting options:

\begin{axis}[
  xmin = -1, xmax = 1,
  ymin = 0, ymax = 2,
  grid = major,
  xlabel = $x$, ylabel = $y$
]
\end{axis}
The \texttt{axis} environment

\begin{axis}[
  xmin = -1, xmax = 1,
  ymin = 0, ymax = 10,
  xtick = {-1,-0.5,...,1},
  ytick = {0,2,...,10},
  minor x tick num = 1,
  minor y tick num = 1,
  grid = major,
  xlabel ={$x$ (\textit{meter})},
  ylabel ={\parbox{2cm}{% 
  \centering
  $\frac{\partial T}{\partial x}$ \\
  ($^\circ\text{C}/\text{m}$)}},
  title = {Temperature gradient},
  axis on top = true]
% a basic tikz drawing command
\fill[blue!40]
  (axis cs: -0.5, 0) -- (axis cs: 0.5, 0) --
  (axis cs: 0.5,10) -- (axis cs: -0.5,10) --
  cycle;
\end{axis}
\begin{axis}[grid = major, xlabel = {$x$}, ylabel = {$y$}, y tick label style = {
    /pgf/number format/.cd, set thousands separator={}
}]
\addplot {-x^5 - 242}; \addlegendentry{model}
\addplot coordinates {
    (-4.77778, 2027.60977)
    (-3.55556, 347.84069)
    (-2.33333, 22.58953)
    (-1.11111, -493.50066)
    ( 0.11111, 46.66082)
    ( 1.33333, -205.56286)
    ( 2.55556, -341.40638)
    ( 3.77778, -1169.2478)
    ( 5.00000, -3269.56775)
}; \addlegendentry{estimate}
\end{axis}
The `\addplot` command

Reading tabular data from file:

\begin{filecontents*}{data1.txt}
Level  |  Cost       |  Error
-------|-------------|---------
1       | 7.0        | 8.4717381e+00
2       | 31.0       | 3.04409349e+00
3       | 111.0      | 1.02214539e+00
4       | 351.0      | 3.30346265e+00
5       | 1023.0     | 1.03886535e+00
6       | 2815.0     | 3.19646457e+00
7       | 7423.0     | 9.65789766e+00
8       | 18943.0    | 2.87339125e+00
9       | 47103.0    | 8.43749881e+00
\end{filecontents*}

\begin{tikzpicture}
\begin{loglogaxis}[xlabel={Cost}, ylabel={Error}]
\addplot [color=red, mark=x] coordinates {
(5, 8.31160034e-02)
(17, 2.54685628e-02)
(49, 7.40715288e-03)
% ...
(9217, 3.26101452e-06)
};
\addplot [color=blue, mark=*] table [x=Cost, y=Error] {data1.txt};
\legend{Case 1, Case 2}
\end{loglogaxis}
\end{tikzpicture}
Plotting data with \pgfplots

What else?

▶ Data column manipulation with \pgfplotstable.
▶ Style customizations of graphic elements.
▶ Available coordinate systems and advanced coordinate calculations. 3D plots.
▶ Exporting \pgfplots sources from other data plotting tools.

Please have a look at the article on ArsTEXnica for more details on \pgfplots:

● De Marco, A., “Graphics for \LaTeX users”. ArsTEXnica 28 (October 2019), pp. 64–100.

Conclusions

We have seen the most common scenarios encountered by \LaTeX{} users when they face the problem of producing quality graphics.

In cases of diagrams, pictures and more or less complicated illustrations the two approaches based on package \texttt{tikz} and on the \texttt{Inkscape} graphics vector software have been presented.

Examples of scientific plots with the package \texttt{pgfplots}.  

A. De Marco  | Graphics for \LaTeX{} users  | \textit{GIT meeting 2019}
Thank you ... 

Questions?

http://texample.net/tikz/examples/dartboard/