Hour 10

Figure Drawing

How a figure can be inserted in a LaTeX document is discussed in Hour 9 on page 81. Besides importing a figure from an external file, LaTeX provides many commands and environments for directly drawing different types of geometric figures.

A geometric figure may be a single or a combination of various smaller elements, like lines and curves. For drawing a figure, a space is first divided into four quadrants by two rectangular coordinate axes (x- and y-axes). The intersection of the axes is called the origin, whose coordinate is (0,0). As shown in Fig. 10.1, x value is positive on the right side of the origin and negative on its left side, while y value is positive above the origin and negative below it. Before drawing a figure, note the following five points:

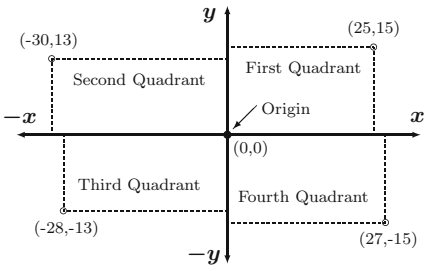


Fig. 10.1 Coordinate axes for drawing figures

1. For drawing figures, a rectangular widow is first reserved through the **picture** environment as **\begin{picture}(**`lx,ly`**)(**`x0,y0`**)**, where (`x0`, `y0`) is the lower left coordinate of the window, and `lx` and `ly` are its lengths along the x- and y-axes, respectively (unlike usual LaTeX commands, which takes an argument in **{}** or **[]**, figure-related commands and environments take coordinates and lengths in **()**, two values separated by a comma). For example, **\begin{picture}(**`50,40`**)(**`0,0`**)** reserves a window for drawing figures with $x \in [0, 50]$ and $y \in [0, 40]$, or **\begin{picture}(**`45,35`**)(**`-20,15`**)** for drawing figures with $x \in [-20, (45 - 20)]$ and $y \in [15, (35 + 15)]$.
2. The scale of drawing can be set by defining the **\unitlength** command outside the **picture** environment, e.g., **\setlength{\unitlength}{**`5mm`**}** for setting 1 unit = 5 mm (millimeter). Other acceptable units are **cm** (centimeter), **in** (inch), **pt** (printer point, 1 in = 72.27 pt, 1 cm = 28.45 pt), **em** (width of `M`), and **pc** (pica, 1 pc = 12 pt).

D. Datta, *LaTeX in 24 Hours: A Practical Guide for Scientific Writing*,
DOI 10.1007/978-3-319-47831-9_10

3. The thickness of a line can be controlled by the **\linethickness{}** command, e.g., **\linethickness{**`0.5`**mm}** for lines of 0.5 mm thickness. Alternatively, the **\thinlines** and **\thicklines** commands may be used directly for thin and thick lines, respectively. These commands can be used in the **picture** environment, and can also be repeated for drawing lines of different thicknesses.
4. Many figure drawing commands do not ask for the starting coordinate of a figure, and by default the figure is started from the current coordinate. Such a command, say `fcmd`, can be forced by the **\put(**`x,y`**){**`fcmd`**}** command to start the figure at (`x,y`). The **\multiput(**`x,y`**)(**Δ`x,`Δ`y`**){**`n`**}{**`fcmd`**}** command can also be used for drawing the same figure `n` times, starting the first one at (`x,y`) and incrementing (`x,y`) each time by (Δ`x`, Δ`y`) for the subsequent figures (**\multiput()(){}{}** is a very convenient command for drawing equidistant parallel lines)[1].
5. The **picture** environment does not support the **\caption{}** command. Hence, the **picture** environment may be put in the **figure** environment with the **\caption{}** and **\label{}** commands, as shown in Hour 9, for the purpose of assigning a serial number to the figure drawn and referring it in the document.

10.1 Circles and Circular Arcs

The most easiest figure in LaTeX is a circle, which is drawn by the **\circle{**`d`**}** or **\circle*{**`d`**}** command (both are defined in the **pict2e** package), where `d` is the diameter of the circle. The **\circle{}** command draws a hollow circle, while the **\circle*{}** command makes it solid. Another available command for drawing a circle is **\bigcircle[**`p`**]{**`d`**}** defined in the **curves** package, where `d` is the diameter and optional `p` is any nonnegative integer representing the pattern (type of line) of the circle (the default value of `p` is 0). The **\bigcircle[]{}** and **\circle{}** commands can also be used for drawing half-filled circles by increasing the line thickness through **\linethickness{}**.

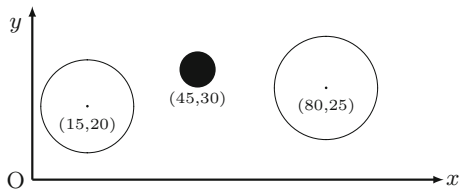
On the other hand, the **\arc[**`p`**](**x', y'**){**`deg`**}** command, defined in the **curves** package, draws a circular arc starting at ($x' + x_c$, $y' + y_c$) and moving anticlockwise through the angle `deg` given in degree (negative value for `deg` can be used for clockwise movement), where (x_c, y_c) is the coordinate of the center of the arc. That is, (x', y') is a relative coordinate that assumes (0, 0) as its center coordinate (x_c, y_c). If not provided (through **\put(){}**), the current coordinate is taken as (x_c, y_c). As in **\bigcircle[**`p`**]{}**, the optional `p` in **\arc[**`p`**]()**`{}` represents the pattern of the arc.

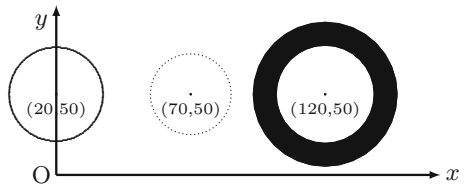
Since none of **\circle{}**, **\circle*{}**, **\bigcircle[]{}** and **\arc[]()**`{}` asks for the coordinate of the center of a circle or an arc, the figure may be drawn through **\put(**x_c, y_c**){**`acomm`**}**, where `acomm` is the figure drawing command and (x_c, y_c) its center coordinate. Some examples of these four commands are shown in Table 10.1 on the following page, where the effects of **\linethickness{}** and **\thinlines** on the line thicknesses are also demonstrated. In Table 10.1, the center points of the

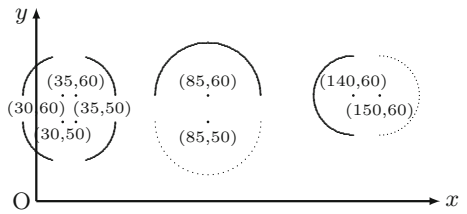
[1] The **\multiput(**`x,y`**)(**Δ`x,`Δ`y`**){**`n`**}{}** command can be used for drawing the same figure `n` times, say equidistant parallel lines, starting the first one at (`x,y`) and incrementing (`x,y`) each time by (Δ`x`, Δ`y`) for the subsequent figures.

Table 10.1 Circle and circular arc drawing

LaTeX input	Output

`\setlength{\unitlength}{0.5mm}`
`\begin{picture}(120,45)(-10,0)`
`\put(15,20){\circle{25}}`
`\put(45,30){\circle*{10}}`
`\linethickness{2mm}`
`\put(80,25){\circle{30}}`
`\end{picture}`
	

`\begin{picture}(150,70)(20,15)`
`\thinlines`
`\put(20,50){\bigcircle{35}}`
`\put(70,50){\bigcircle[4]{30}}`
`\linethickness{3mm}`
`\put(120,50){\bigcircle{45}}`
`\end{picture}`
	

`\begin{picture}(170,70)(20,20)`
`\thinlines`
`\put(35,60){\arc(15,0){75}}`
`\put(30,60){\arc(-15,0){-75}}`
`\put(35,50){\arc(15,0){-75}}`
`\put(30,50){\arc(-15,0){75}}`
`%`
`\put(85,60){\arc(20,0){180}}`
`\put(85,55){\arc[4](20,0){-180}}`
`%`
`\put(140,60){\arc(0,15){180}}`
`\put(150,60){\arc[4](0,-15){180}}`
`\end{picture}`
	

circles and arcs are encircled by small circles as well as their coordinates are shown for illustrative purpose only (the commands for the same are not shown in the LaTeX input file). Note that figures can be drawn in different colors also, e.g., `\put(15,25){\textcolor{red}{\circle{20}}}` will draw a circle in red color.

10.2 Straight Lines and Vectors*

A straight line is drawn by the `\line(`l_x, l_y`){f}` command, where l_x and l_y are the least nondivisible factors of the end coordinate (x_e, y_e) of the line (i.e., slope of the line), and f is their greatest common factor (i.e., horizontal or vertical length of the line, horizontal for $l_x \neq 0$ and vertical for $l_x = 0$). When multiplied by f, (l_x, l_y) gives the end coordinate, i.e., $(l_x f, l_y f) = (x_e, y_e)$. The application of `\line(){}` is slightly complicated. The terms l_x and l_y must obey the following three rules:

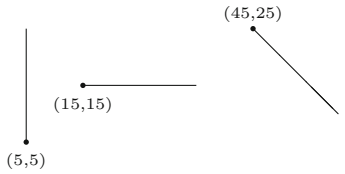
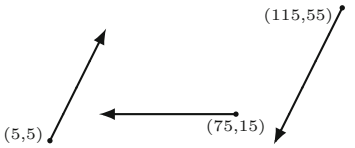
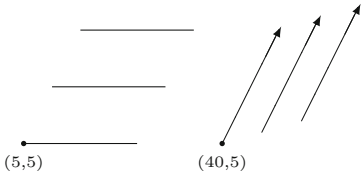
- ▷ Both l_x and l_y are whole numbers, including 0, either positive or negative.
- ▷ Only limited combinations of 0, ±1, …, ±6 in (−6,6) are permitted to l_x and l_y.

▷ l_x and l_y should not have any common factor, like (2,4) or (6,-2) are not permitted. However, (±1,±1) are permitted. In general, the possible combinations of (l_x,l_y) are (1,0), (1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (2,3), (2,5), (3,4), (3,5), (4,5), and (5,6), including their altered combinations and negative counterparts.

Drawing a horizontal or a vertical line is easy. For a horizontal line, the command is simplified to \line(±1,0){hlen}, where hlen is the true length of the line, toward right for \line(1,0){} and toward left for \line(-1,0){}. Similarly, the command for a vertical line is \line(0,±1){hlen}, where the line moves upward for \line(0,1){} and downward for \line(0,−1){}. While seeking horizontal and vertical lines, even if non-unity values are assigned to l_x and l_y (like \line(3,0){} or \line(0,−5){}), they will be treated internally as $l_x = \pm 1$ and $l_y = \pm 1$.

Similar to \line(l_x,l_y){f} for drawing a straight line, the command for drawing a vector (a straight line with an arrow at one end) is \vector(l_x,l_y){f} with l_x,$l_y \in (-4, 4)$. Like the commands of Table 10.1 used for drawing circles or circular arcs, \line(){} and \vector(){} also do not ask for the starting coordinate of a line or a vector. Therefore, their starting coordinates may be specified through \put(){} or \multiput()(){}{}. Some examples of \line(){} and \vector(){} are shown in Table 10.2,

Table 10.2 Straight line and vector (arrow) drawing

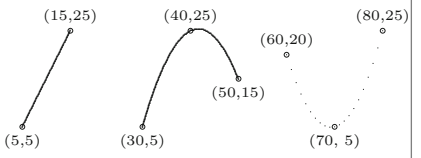
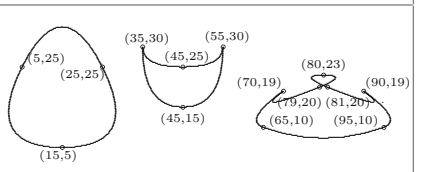
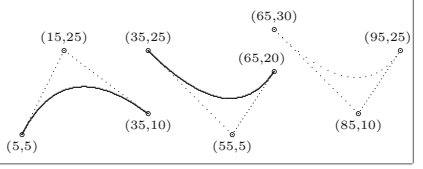
LaTeX input	Output
\setlength{\unitlength}{0.75mm} \begin{picture}(60,30)(0,0) \put(5,5){\line(0,1){20}} \put(15,15){\line(1,0){20}} \put(45,25){\line(1,-1){15}} \end{picture}	
\begin{picture}(120,60)(0,0) \thicklines \put(5,5){\vector(1,2){20}} \put(75,15){\vector(-1,0){50}} \put(115,55){\vector(-1,-2){25}} \end{picture}	
\setlength{\unitlength}{0.75mm} \begin{picture}(60,30)(0,0) \multiput(5,5)(5,10){3}{\line(1,0){20}} \multiput(40,5)(7,2){3}{\vector(1,2){10}} \end{picture}	

where the starting points of the lines and vectors are encircled by small circles as well as their coordinates are shown for illustrative purpose only.

10.3 Curves*

Frequently used commands for drawing curves are **\curve**[p](x1, y1, ..., xn, yn) and **\closecurve**[p](x1, y1, ..., xn, yn), defined in the **curves** package, which draw respectively open and closed curves through the given coordinate points with the optional p as the patterns of the curves. In the **\curve[]()** command, two points draw a straight line and three points draw a parabola. The **\closecurve[]()** command, where at least three points are required, draws a closed curve with continuous tangents at all the points. Besides these two commands, there is **\qbezier**[N](x1, y1)(x2, y2)(x3, y3) for drawing a quadratic Bézier curve through the end points (x1, y1) and (x3, y3), where the middle point (x2, y2) is known as the control point and it is the intersection of the two tangents to the curve at the two end points. The optional argument N instructs to approximate the curve through (N+1) points. If N is not provided, its value is calculated automatically to produce a solid curve. Some examples of **\curve[]()**, **\closecurve[]()** and **\qbezier[]()()()** are shown in Table 10.3, where the curve

Table 10.3 Open and closed curves, and Bézier quadratic curves

LaTeX input	Output
\setlength{\unitlength}{0.7mm} \begin{picture}(80,30)(4,0) \curve(5,5, 15,25) \curve(30,5, 40,25, 50,15) \curve[10](60,20, 70,5, 80,25) \end{picture}	
\setlength{\unitlength}{0.7mm} \begin{picture}(110,40)(4,0) \closecurve(5,25, 15,5, 25,25) \closecurve(35,30, 45,15, 55,30, 45,25) \closecurve(65,10, 70,19, 79,20, 80,23, 81,20, 90,19, 95,10) \end{picture}	
\setlength{\unitlength}{0.7mm} \begin{picture}(100,35)(0,0) \qbezier(5,5)(15,25)(35,10) \qbezier(35,25)(55,5)(65,20) \qbezier[20](65,30)(85,10)(95,25) \end{picture}	

generating points are encircled by small circles as well as their coordinates are shown for illustrative purpose only (these are not shown in the LaTeX input file). Since all the relevant coordinates are provided to **\curve[]()**, **\closecurve[]()** and **\qbezier[]()()()**, the **\put(){}** command, as used in Tables 10.1 and 10.2, is not required here to specify any more point.

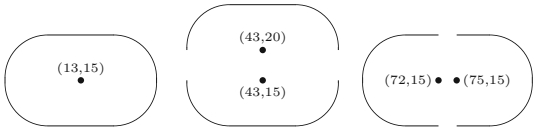
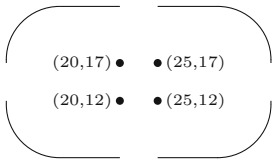
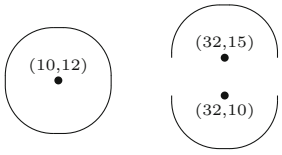
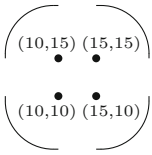
10.4 Oval Boxes*

The \oval(lx,ly)[part] command draws a box of dimensions lx and ly with rounded corners. The part option allows to draw only a part of a box, whose permissible values are given in Table 10.4.

Table 10.4 Options to the \oval()[] command for drawing partial ovals

Option	Meaning
t	Top-half of the box.
b	Bottom-half of the box.
l	Left-half of the box.
r	Right-half of the box.
tl or lt	Top-left quarter of the box.
tr or rt	Top-right quarter of the box.
bl or lb	Bottom-left quarter of the box.
br or rb	Bottom-right quarter of the box.

Table 10.5 Oval boxes (boxes with rounded corners) through the \oval()[] command

LaTeX input	Output
\setlength{\unitlength}{1mm} \begin{picture}(88,30)(0,0) \put(13,15){\oval(25,15)} \put(43,20){\oval(25,15)[t]} \put(43,15){\oval(25,15)[b]} \put(72,15){\oval(25,15)[l]} \put(75,15){\oval(25,15)[r]} \end{picture}	
\setlength{\unitlength}{1mm} \begin{picture}(45,30)(0,0) \put(20,17){\oval(30,15)[tl]} \put(25,17){\oval(30,15)[tr]} \put(20,12){\oval(30,15)[bl]} \put(25,12){\oval(30,15)[br]} \end{picture}	
\setlength{\unitlength}{1mm} \begin{picture}(40,25)(0,0) \put(10,12){\oval(14,14)} \put(32,15){\oval(14,14)[t]} \put(32,10){\oval(14,14)[b]} \end{picture}	
\setlength{\unitlength}{1mm} \begin{picture}(25,25)(0,0) \put(10,15){\oval(14,14)[tl]} \put(15,15){\oval(14,14)[tr]} \put(10,10){\oval(14,14)[bl]} \put(15,10){\oval(14,14)[br]} \end{picture}	

Some applications of \oval()[] are shown in Table 10.5, where a box is put in a specified base point through the \put(){} command. As before, the base points

by small solid circles along with their coordinates are also shown for illustrative purpose. Since no option for any part (i.e., the optional `part`) is provided, the very first **\put(){\oval()}** in each of the first and third examples draws a complete box with the base point as its center coordinate. Any other command in Table 10.5 draws a part of a box due to the presence of one of the options as given in Table 10.4. Note that, whether a complete box or a part of it is opted, the complete size of the box (l_x, l_y) is to be provided to **\oval()[]**, which draws the box or its part with the base point, specified through **\put(){}**, as the center coordinate of the complete box. It is observed that for equal values of l_x and l_y, in some small amount (a maximum of around 14 mm), **\oval()[]** draws a circle-like box or a part of it. Some such examples are also shown in the last two examples of Table 10.5.

10.5 Texts in Figures*

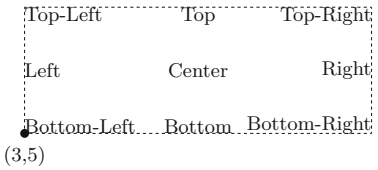
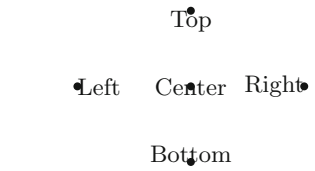
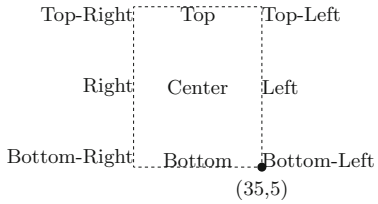
If required, some texts can be inserted in a figure through **\put(x,y){atext}** in the **picture** environment, where `atext` is the texts to be inserted and `(x,y)` is its lower left coordinate. Additionally, there are three box making commands, **\makebox(**l_x**,**l_y**)[pos]{atext}**, **\framebox(**l_x**,**l_y**)[pos]{atext}** and **\dashbox{dsize}(**l_x**,**l_y**)[pos]{atext}**, for conveniently inserting texts in a rectangular box positioned by **\put(){}**. The fields l_x and l_y are respectively the horizontal and vertical lengths of a box, `atext` is the texts to be inserted in the figure, and `dsize` is the size of the dashes in a dashed box. The optional argument `pos` is the vertical and horizontal positions of `atext` in the box, whose permissible values are given in Table 10.6.

Table 10.6 Settings of the **\makebox()[]{}**, **\framebox()[]{}** and **\dashbox{}()[]{}** commands for positioning texts in figures

Argument	Meaning
t	Vertically top aligned and horizontally centered.
b	Vertically bottom aligned and horizontally centered.
c	Centered both vertically and horizontally (default value).
l	Horizontally left aligned and vertically centered.
r	Horizontally right aligned and vertically centered.
s	Horizontally aligned on both the edges and vertically centered.
tl or **lt**	Top-left corner of the box.
tr or **rt**	Top-right corner of the box.
bl or **lb**	Bottom-left corner of the box.
br or **rb**	Bottom-right corner of the box.

No visible box is produced by **\makebox()[]{}**, it only reserves a rectangular area for printing texts. In the first example in Table 10.7 on the next page, a box of size 55 mm × 20 mm is created a number of times at point (3, 5) by **\put(){\makebox()[]{}}** and variations in the positions of texts with the `pos` option are shown. The base coordinate (3, 5) and a dotted box are also shown for illustrative purpose. The **\makebox()[]{}** command has two special properties. First, a zero-dimension box

Table 10.7 Texts in figures through the \makebox( )[]{} command

LaTeX input	Output
\setlength{\unitlength}{1mm} \begin{picture}(60,30)(0,0) \put(3,5){\makebox(55,20)[c]{Center}} \put(3,5){\makebox(55,20)[t]{Top}} \put(3,5){\makebox(55,20)[b]{Bottom}} \put(3,5){\makebox(55,20)[l]{Left}} \put(3,5){\makebox(55,20)[r]{Right}} \put(3,5){\makebox(55,20)[tl]{Top-Left}} \put(3,5){\makebox(55,20)[tr]{Top-Right}} \put(3,5){\makebox(55,20)[bl]{Bottom-Left}} \put(3,5){\makebox(55,20)[br]{Bottom-Right}} \end{picture}	
\setlength{\unitlength}{1mm} \begin{picture}(50,30)(0,0) \put(25,25){\makebox(0,0)[t]{Top}} \put(25,15){\makebox(0,0)[c]{Center}} \put(25,5){\makebox(0,0)[b]{Bottom}} \put(10,15){\makebox(0,0)[l]{Left}} \put(40,15){\makebox(0,0)[r]{Right}} \end{picture}	
\setlength{\unitlength}{1mm} \begin{picture}(50,30)(0,0) \put(35,5){\makebox(-20,25)[c]{Center}} \put(35,5){\makebox(-20,25)[t]{Top}} \put(35,5){\makebox(-20,25)[b]{Bottom}} \put(35,5){\makebox(-20,25)[l]{Left}} \put(35,5){\makebox(-20,25)[r]{Right}} \put(35,5){\makebox(-20,25)[tl]{Top-Left}} \put(35,5){\makebox(-20,25)[tr]{Top-Right}} \put[35,5){\makebox(-20,25)[bl]{Bottom-Left}} \put(35,5){\makebox(-20,25)[br]{Bottom-Right}} \end{picture}	

can also be created, in which case a piece of texts is positioned with respect to the base coordinate of \makebox()[]{}, some examples of which are shown in the second part of Table 10.7. Second, a box of a negative dimension can also be created, which means the negative side of the base point. As shown in the third part of Table 10.7, a box of size (−20, 25 mm) means 20 mm left and 25 mm above the base point. Such a negative-dimensional box may be useful if a piece of texts is to be inserted on the left or below the base point.

Unlike \makebox()[]{}, \framebox()[]{} produces a visible box of a specified size. Moreover, no negative or zero-dimensional box is permitted in \framebox()[]{}. If no size is provided, a box of an arbitrary size is created. All other issues of \framebox()[]{} are the same with those of \makebox()[]{}. The \dashbox{}()[]{} command is similar with \framebox()[]{}, with the only difference that it produces a box by dashed lines. Some applications of \framebox()[]{} and \dashbox{}()[]{} are shown in Table 10.8 on the next page, where the base coordinates of the boxes are marked by small solid circles for illustrative purpose.

Apart from the above three commands, the \parbox[pos]{l_x}{atext} command can also be used for printing texts in the picture environment. The meanings of the

Table 10.8 Boxed texts in figures using the \framebox( )[]{} and \dashbox{}( )[]{} commands

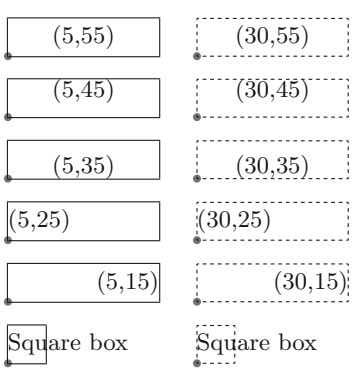
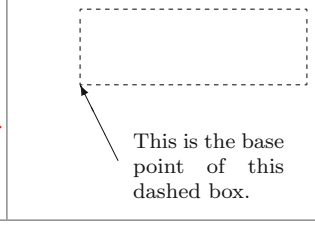
LaTeX input	Output
\setlength{\unitlength}{1mm} \begin{picture}(55,60)(0,0) \put(5,55){\framebox(20,5)[c]{(5,55)}} \put(5,45){\framebox(20,5)[t]{(5,45)}} \put(5,35){\framebox(20,5)[b]{(5,35)}} \put(5,25){\framebox(20,5)[l]{(5,25)}} \put(5,15){\framebox(20,5)[r]{(5,15)}} \put(5,5){\framebox(5,5)[l]{Square box}} % \put(30,55){\dashbox{0.5}(20,5)[c]{(30,55)}} \put(30,45){\dashbox{0.5}(20,5)[t]{(30,45)}} \put(30,35){\dashbox{0.5}(20,5)[b]{(30,35)}} \put(30,25){\dashbox{0.5}(20,5)[l]{(30,25)}} \put(30,15){\dashbox{0.5}(20,5)[r]{(30,15)}} \put(30,5){\dashbox{0.5}(5,5)[l]{Square box}} \end{picture}	

Table 10.9 Multi-line texts in a figure through the \parbox[]{}{} command

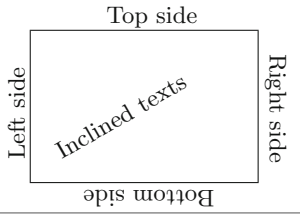
LaTeX input	Output
\setlength{\unitlength}{1mm} \begin{picture}(40,35)(0,0) \put(5,20){\dashbox{0.6}(30,10){}} \put(10,10){\vector(-1,2){5}} \put(12,5){\parbox[b]{2.0cm}{This is the base point of this dashed box.}} \end{picture}	

arguments of \parbox[]{}{} are the same as mentioned above. Like \makebox( )[]{}, \parbox[]{}{} also does not produce any visible box. As shown in Table 10.9, an advantage of using \parbox[]{}{} is that a long piece of texts is split over multiple lines, if the size of the box is not sufficient to hold the entire texts in a single line[2].

All the above commands print texts in the horizontal direction. Some texts in a figure may need to be printed in an inclined direction also. The command for the same is \rotatebox{arotate}{atext} defined in the rotating package, which rotates atext by angle arotate (in degree) from the horizontal (a positive value of arotate rotates in the counter-clockwise direction, while a negative value rotates in the clockwise direction). Applications of \rotatebox{}{} are shown in Table 10.10 on the next page.

[2]The \parbox[]{}{} command splits its textual argument over multiple lines, if the size of the box is not sufficient to hold the entire texts in a single line.

Table 10.10 Rotated texts in figures through the \rotatebox{}{} command

LaTeX input	Output
\setlength{\unitlength}{1mm} \begin{picture}(40,37)(0,5) \put(5,15){\framebox(30,20){}} \put(2,18){\rotatebox{90}{Left side}} \put(15,36){\rotatebox{0}{Top side}} \put(36,32){\rotatebox{-90}{Right side}} \put(12,14){\rotatebox{-180}{Bottom side}} \put(8,18){\rotatebox{30}{Inclined texts}} \end{picture}	

Note that the commands of Table 17.1 on page 161 can also be used in the **picture** environment through the positioning command \put(){} for printing texts in boxes.

10.6 Compound Figures*

Once commands for drawing different geometric figures are known, they can be combined for forming a compound figure also. Some examples of such figures are given in Table 10.11, where the application of \multiput()(){}{} is also shown while drawing parallel lines of the rectangle and parallelogram.

Table 10.11 Compound figures through multiple commands

Name	LaTeX Coding	Figure
Triangle	\begin{picture}(25,25)(0,0) \put(0,0){\line(1,0){20}} \put(0,0){\line(1,2){10}} \put(20,0){\line(-1,2){10}} \end{picture}	
Rectangle	\begin{picture}(45,25)(0,0) \multiput(10,0)(30,0){2}{\line(0,1){20}} \multiput(10,0)(0,20){2}{\line(1,0){30}} \end{picture}	
Parallelogram	\begin{picture}(50,40)(0,0) \multiput(5,5)(10,20){2}{\line(1,0){40}} \multiput(5,5)(40,0){2}{\line(1,2){10}} \end{picture}	
Quadrilateral	\begin{picture}(45,25)(0,0) \curve(10,0, 35,4) \curve(35,4, 40,15) \curve(40,15, 15,20) \curve(15,20, 10,0) \end{picture}	
Decay curve	\begin{picture}(85,25)(0,0) \curve(0,20, 5,40, 10,20) \curve(10,20, 15,0, 20,20) \curve(20,20, 25,35, 30,20) \curve(30,20, 35,5, 40,20) \curve(40,20, 45,30, 50,20) \curve(50,20, 55,10, 60,20) \curve(60,20, 65,25, 70,20) \curve(70,20, 75,15, 80,15) \end{picture}