LaTeX - Importing graphics

Sayak Haldar

sayakhaldar@ymail.com

Shuvam Bosana

shuvambosana0705@gmail.com

Subham Banerjee

subhambansphs@gmail.com

Students of Computer Science & Technology

Bengal Engineering & Science University,Shibpur.

Abstract

This documentation is how to import graphics(pictures) to your document and how to treat them as a float

Contents

1 Introduction:

There are two possibilities to include graphics in your document. Either create them with some special code, a topic which will be discussed in the Creating Graphics part, (see Introducing Procedural Graphics) or import productions from third party tools, which is what we will be discussing here.

But LaTeX cannot manage pictures directly: in order to introduce graphics within documents, LaTeX just creates a box with the same size as the image you want to include and embeds the picture, without any other processing. This means you will have to take care that the images you want to include are in the right format to be included. This is not such a hard task because LaTeX supports the most common picture formats around.

Now, as we stated before, LaTeX can't manage pictures directly, so you will need some extra help: you have to load the graphicx package in the preamble of your document:

```
\usepackage{graphicx}
```

2 Document options:

The graphics and graphicx packages recognize the draft and final options given in the \documentclass[...]{...} command at the start of the file. (See Document Classes.) Using draft as the option will suppress the inclusion of the image in the output file and will replace the contents with the name of the image file that would have been seen. Using final will result in the image being placed in the output file. The default is final

3 Supported image format:

As explained before, the image formats you can use depend on the driver that graphicx is using but, since the driver is automatically chosen according to the compiler, then the allowed image formats will depend on the compiler you are using.

We will discuss about this topic later.But right now what you have to know Using pdflatex will be usually much more simple for graphics inclusion as it supports widespread formats such as PDF, PNG and JPG. As knowing it is more than enough for now.

4 Including graphics:

Now that you have seen which formats you can include and how to manage those formats, it's time to learn how to include them in our document. After you have loaded the graphicx package in your preamble, you can include images with \includegraphics, whose syntax is the following:

```
\includegraphics[attr1=val1, attr2=val2, ..., attrn=valn]
{imagename}
```

You have to know that the arguments in the square brackets are optional,whereas arguments in the curly brackets are compulsory.

Variety of possible attributes we can do with a graphics/figure:

scale=xx	Scales the image by the desired scale factor. e.g, 0.5 to reduce by half, or 2 to double.
width=xx	Specify the preferred width of the imported image to xx.Whre xx=magnitude of length with proper unit.
height=xx	Specify the preferred height of the imported image to xx.
totalheight=xx	To specify height plus depth of the figure(in case the picture is rotated). In particular if the figure has been rotated by $-90°$ then it will have zero height but large depth.
keepaspectratio	This can be set to either true or false. When true, it will scale the image according to both height and width, but will not distort the image, so that neither width nor height are exceeded.
angle=xx	This option can rotate the image by xx degrees (counter-clockwise)

+continued to next page

continued from previous page

trim=l b r t	This option will crop the imported image by l from the left, b from the bottom, r from the right, and t from the top. Where l, b, r and t are lengths.
clip	For the trim option to work, you must set clip=true.It is a boolean type parameter, i.e. can be set to either true or false.
page=x	If the image file is a pdf file with multiple pages, this parameter allows you to use a different page than the first.
draft	It can also be set to either true or false. If draft is set to the true option then prevents figure from being imported, but created a named box with the dimensions of the figure (this option is used to speed up processing).
clip	It can also be set to either true or false.By default it is set to true. It Clips the graphic to the bounding box.
bb = llx lly urx ury	enters the bounding box coordinates, which are given by default in points (1/72 inch), manually (the bounding box might be missing or you might want to alter it)
viewport = llx lly urx ury	specifies bounding box w.r.t. bottom left of existing bounding box; used with clip to select a part of the image (or to clear unwanted margins. An example: to 'view' the 1in square in the bottom left hand corner of the area specified by the bounding box, use the argument viewport=0 0 72 72.
• hiresbb	Boolean valued key. It reads the bounding box information from the line %%HiResBoundingBox in figure file. If set to true (just specifying hiresbb is equivalent to hiresbb=true) then TEX will look for %%HiResBoundingBox lines. rather than %%BoundingBox. It may be set to false to overrule a default setting of true set by the hiresbb package option.

In order to use more than one option at a time, simply separate each with a comma. The order you give the options matters. E.g you should first rotate your graphic (with angle) and then specify its width.

Included graphics will be inserted just there, where you placed the code, and the compiler will handle them as "big boxes". As we will see in the floats section, this can disrupt the layout; you'll probably want to place graphics inside floating objects. Also note that the trim option does not work with XeLaTex.

Be careful using any options, if you are working with the chemnum-package. The labels defined by \cmpdref{<label name>} might not behave as expected. Scaling the image for instance may be done by **\scalebox** instead. The star version of the command will work for .eps files only. For a more portable solution, the standard way should take precedence. The star command will take the crop dimension as extra parameter:

```
\includegraphics*[100,100][300,300]{mypicture}
```

5 Examples:

5.1 Importing a picture without any option specification:

This command will only insert a picture without doing anything and this will enter the picture at the cursor position:

```
\includegraphics{Chick}
```

This will produce output like:

Now, if you the seme picture to be inserted in the middle of the next line Simply use **center** environment and wrap the **\includegraphics** command within it.

```
\begin{center}
\includegraphics{Chick}
\end{center}
```

It will produce output like:

Now, we are going to discuss the use of some common specification parameters like:scale,width etc. . . (all these are optional parameters)

5.2 Examples of using the specification parameters:

5.2.1 scale:

To scale down a larger image,(which may be larger than the page size) we need to scale it down. And to magnify a samller image, we need to scale it up.(the picture may not look proper after doing it because of the resolution factor). For both purposes we need to use the optional parameter "scale":

```
\includegraphics[scale=0.5]{Chick}
```

This will produce output like:

5.2.2 width:

To specify the width of the picture we need to use the optional parameter ”width”.

```
\includegraphics[width=0.2\textwidth]{Chick}
```

This will produce output like:

One can specify the width with respect to the width of a line in the local environment (**\linewidth**), the width of the text on a page (**\textwidth**) or the height of the text on a page (**\textheight**) as well as to a length unit(like 2cm).

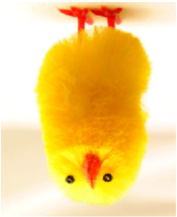
Note: In both the examples,we did not mention the file extension name of the picture, i.e the picture format:-whether it is .jpg,.png or .pdf. However, mentioning the file extension name is not necessary.

5.2.3 angle:

To rotate a picture in 180°:

```
\includegraphics[scale=0.5, angle=180]{Chick}
```

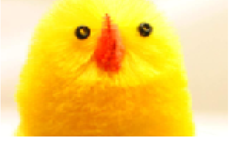
It will produce output like:



Here, we use multiple arguments. We rotate it to 180° as well as scale it to half.

5.2.4 trim:

```
\includegraphics[trim = 5mm 30mm 5mm 5mm, clip, width=3cm]{Chick}
```

The output would be like: 

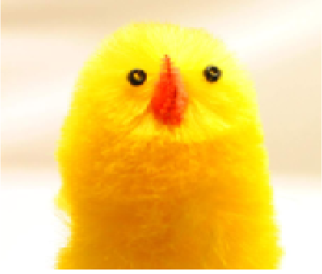
Here, we also specify the width of picture as 3cm.

5.2.5 viewport:

The viewport optional command of the **\includegraphics** command tells what portion of the image file to include. The first two coordinates are the (x,y) coordinates in pixels of the lower left corner of the portion of the image file you want to include. The second two coordinates are the upper right values of (x,y). i.e viewport crops a picture to the desired size.

```
\includegraphics*[viewport=0 60 120 180]{Chick.png}
```

It will produce output like:



Notice that, here we use the starred version of the command.

5.3 Spaces in names:

If the image file were called ”Chick picture.png” instead of ”Chick”,you need to include the full filename when importing the image,i.e you have to use:

```
\includegraphics{Chick picture.png}
```

instead of

```
\includegraphics{Chick}
```

One option is to not use spaces in file names while you are importing a picture is to replace space with underscore,i.e to use the command:

```
\includegraphics{Chick_picture.png}
```

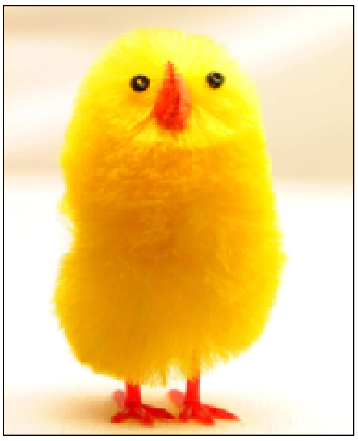
5.4 Borders:

It is possible to have LaTeX create a border around your image by using **\fbox**. you can also control the border padding with the **\setlength \fboxsep{0pt}** command. opt for the case to avoid any padding, so that the border will be placed tightly around the image. However you can also set it to 0.5pt,1pt or 2pt or to some other options according to our choice. The thickness of the border is adjusted by **\setlength\fboxrule{0.5pt}** command. however youcan also change 0.5pt accoring to your choice.

Now we will illustrate two examples for this portion. First, with a border which will be placed tightly around the image and second, with a border which will not be placed tightly around the image. So that, we could understand this topic properly.

```
\setlength\fboxsep{0pt}
\setlength\fboxrule{0.5pt}
\fbox{\includegraphics{Chick}}
```

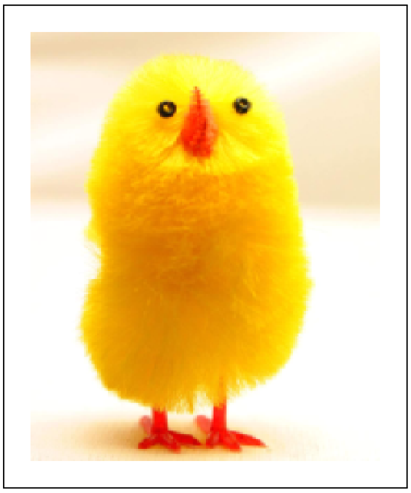
The output with respect to these commands would be like:



Now, clearly this picture has a tight border. Now, if we write:

```
\setlength\fboxsep{10pt}
\setlength\fboxrule{0.5pt}
\fbox{\includegraphics{Chick}}
```

The output would be like:



We have used 10pt here. So that, we could understand the difference properly.

5.5 Additonal features:

5.5.1 Reflecting a picture:

Here, first we will show you the original picture, then the reflected one:

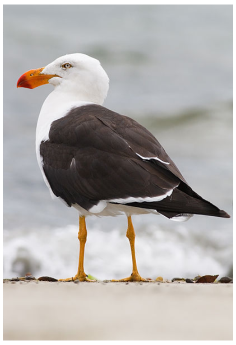
Original picture:

The original picture looks like:

Reflected picture:

```
\begin{center}
\reflectbox{\includegraphics[scale=0.7]{Gull}}
\end{center}
```

It will produce output like:



6 Treating picture as a float:

First, we need to know about float.

6.1 Float:

Floats are containers for things in a document that cannot be broken over a page. LaTeX by default recognizes ”table” and ”figure” floats, but you can define new ones of your own (see Custom Floats below). Floats are there to deal with the problem of the object that won’t fit on the present page, and to help when you really don’t want the object here just now.

Floats are not part of the normal stream of text, but separate entities, positioned in a part of the page to themselves (top, middle, bottom, left, right, or wherever the designer specifies). They always have a caption describing them and they are always numbered so they can be referred to from elsewhere in the text. LaTeX automatically floats Tables and Figures, depending on how much space is left on the page at the point that they are processed. If there is not enough room on the current page, the float is moved to the top of the next page. This can be changed by moving the Table or Figure definition to an earlier or later point in the text, or by adjusting some of the parameters which control automatic floating.

6.2 Figures:

To create a figure that floats, use the figure environment.

```
\begin{figure}[placement specifier]
... figure contents ...
\end{figure}
```

The previous section mentioned how floats are used to allow LaTeX to handle figures, while maintaining the best possible presentation. However, there may be times when you disagree, and a typical example is with its positioning of figures. The placement specifier parameter exists as a compromise, and its purpose is to give the author a greater degree of control over where certain floats are placed.

Now, we need to know about placement specifiers.

Definition of Placement specifier

It is a parameter exists as a compromise, and its purpose to give us the greater degree of control over where certain floats are placed.

Specifier	Permission
h	Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot)
t	Position at the top of the page.
b	Position at the bottom of the page.
p	Put on a special page for floats only.
!	Override internal parameters LaTeX uses for determining ”good” float positions.
H	Places the float at precisely the location in the LaTeX code. Requires the float package. This is somewhat equivalent to h!.

For doing some operations to a float, (Like using H as a placement specifier for a figure) we need to include the float package, i.e we have to mention the following command in the preamble of our document:

```
\usepackage{float}
```

Treating a picture as a float i.e as a figure also provides you the option to add a list of figures at the beginning of the document:

```
\listoffigures
```

6.3 Captions:

It is always good practice to add a caption to any figure or table. Fortunately, this is very simple in LaTeX . All you need to do is use the following command:

```
\caption{''text''}
```

in the float environment. Because of how LaTeX deals sensibly with logical structure, it will automatically keep track of the numbering of figures, so you do not need to include this within the caption text.

The location of the caption is traditionally underneath the float. However, it is up to you to therefore insert the caption command after the actual contents of the float (but still within the environment). If you place it before, then the caption will appear above the float.
Now, we will illustrate the process of giving a caption by providing an example:

```
\begin{figure}[h!]
\caption{A picture of a chick}
\centering
\includegraphics[scale=0.70]{Chick}
\end{figure}
```

Figure 1: A picture of a chick

This would be the output of the given code.

Note: Now the placement specifier h has changed to ht. Here we use the **\caption{}** before the **\includegraphics** command. So, We get the caption before the picture. If we use **\caption{}** after **\includegraphics** then we get the caption after the picture. Now, we use **\centering**(it works like the **center** environment) so that we get the picture at the middle of the page.In the example below, We will illustrate what will happen if we dont use **\centering**.

add the caption after the picture.

```
\begin{figure}[ht]
\centering
\includegraphics[scale=0.70]{Chick}
\caption{A picture of a chick}
\end{figure}
```

Figure 2: A picture of a chick

This will produce output like:

Now, we will show you how to add a caption after the picture:

Figure 3: A picture of a chick

```
\begin{figure}[ht]
\centering
\includegraphics[scale=0.70]{Chick}
\caption{A picture of a chick}
\end{figure}
```

It will produce output like:

6.3.1 Side captions:

It is sometimes desirable to have a caption appear on the side of a float, rather than above or below. The **sidecap** package can be used to place a caption beside

Figure 4: A picture of a chick

a figure or table. The following example demonstrates this for a figure by using a SCfigure environment in place of the figure environment.

```
\begin{SCfigure}
  \centering
  \includegraphics[scale=0.60,width=0.5\textwidth]
    {Giraffe}% picture filename
  \caption{ The giraffe {\textit{(Giraffa camelopardalis)}} is an
African even-toed ungulate mammal, the tallest living terrestrial
animal and the largest ruminant. Its species name refers to its
camel-like appearance and the patches of color on its fur. Its
chief distinguishing characteristics are its extremely long neck
and legs, its horn-like ossicones, and its distinctive coat
patterns. It stands 5{6 m (16{20 ft) tall and has an average
weight of 1,600 kg (3,500 lb) for males and 830 kg (1,800 lb) for
females. It is classified under the family Giraffidae, along with
its closest extant relative, the okapi. The nine subspecies are
distinguished by their coat patterns.
}
\end{SCfigure}
```

If you load the **sidecap** package in your preamble, and use the given command, you will get an output like the one in the next page.

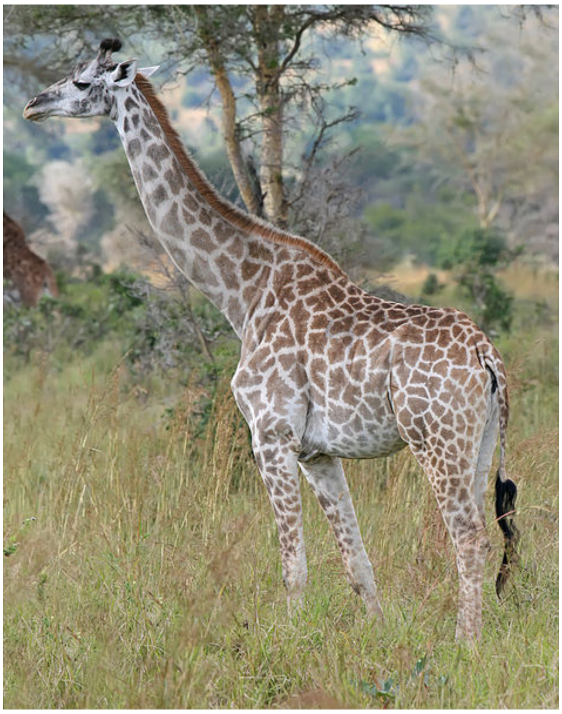


Figure 5: The giraffe *(Giraffa camelopardalis)* is an African even-toed ungulate mammal, the tallest living terrestrial animal and the largest ruminant. Its species name refers to its camel-like appearance and the patches of color on its fur. Its chief distinguishing characteristics are its extremely long neck and legs, its horn-like ossicones, and its distinctive coat patterns. It stands 5–6 m (16–20 ft) tall and has an average weight of 1,600 kg (3,500 lb) for males and 830 kg (1,800 lb) for females. It is classified under the family Giraffidae, along with its closest extant relative, the okapi. The nine subspecies are distinguished by their coat patterns.

Now, you already know how to provide caption to a picture. Now, you will know how to producce a list of figures at the beginning of your document.

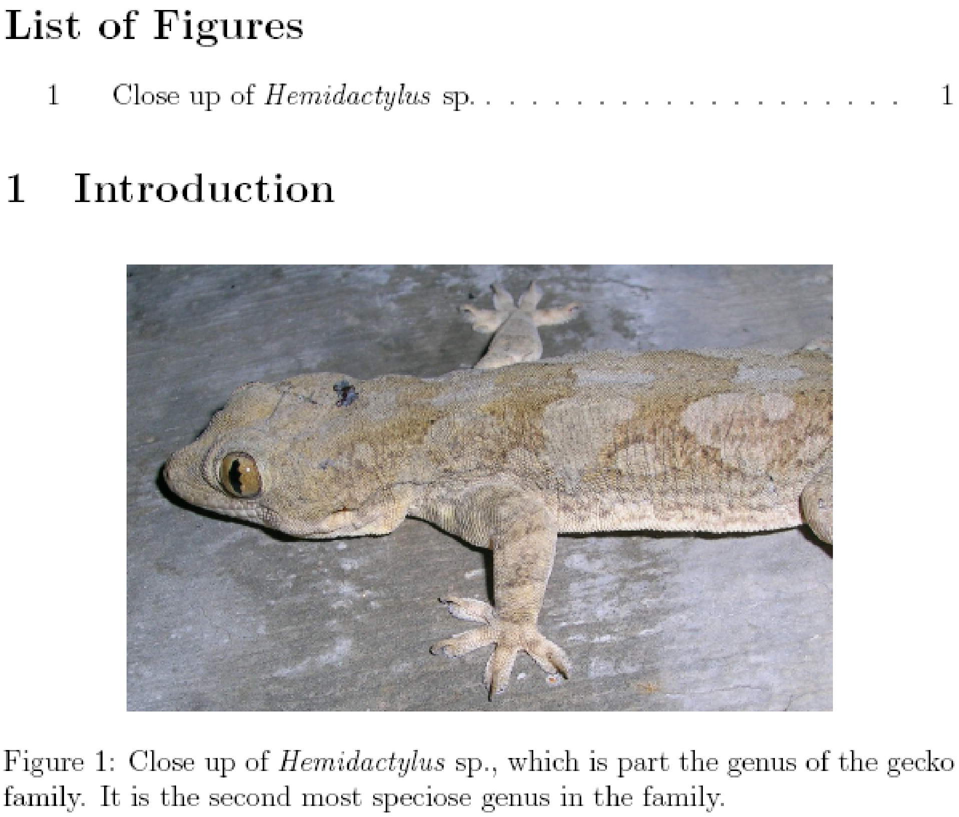
6.4 Lists of figures:

Captions can be listed at the beginning of a paper or report in a ”List of Figures” section by using the **\listoffigures** command, respectively. The caption used for each figure will appear in these lists, along with the figure numbers, and page numbers that they appear on.

The **\caption** command also has an optional parameter, **\caption[”short”]{”long”}** which is used for the List of Tables or List of Figures. Typically the short description is for the caption listing, and the long description will be placed beside the figure or table. This is particularly useful if the caption is long, and only a ”one-liner” is desired in the figure/table listing. Here is an example of this usage:

```
\documentclass[12pt]{article}
\usepackage{graphicx}

\begin{document}

\listoffigures

\section{Introduction}

\begin{figure}[hb]
  \centering
  \includegraphics[width=4in]{gecko}
  \caption[Close up of \textit{Hemidactylus} sp.]
   {Close up of \textit{Hemidactylus} sp., which is
   part the genus of the gecko family. It is the
   second most speciose genus in the family.}
\end{figure}
\end{document}
```

You will get a list of figures at the beginning of your document like the following picture.



List of Figures

1 Introduction

Figure 1: Close up of *Hemidactylus* sp., which is part the genus of the gecko family. It is the second most speciose genus in the family.

6.5 Labels and cross referencing:

If you want to label a figure so that you can reference it later, you have to add the label after the caption (inside seems to work in LaTeX) but inside the floating environment. If it is declared outside, it will give the section number.

If the label picks up the section or list number instead of the figure number, put the label inside the caption to ensure correct numbering. If you get an error when the label is inside the caption, use **\protect** in front of the **\label** command.You will learn more information about it in the 'Labels and Cross Referencing' section.

6.6 Wrapping text around figures:

Although not normally the case in academic writing, an author may prefer that some floats do not break the flow of text, but instead allow text to wrap around it. (Obviously, this effect only looks decent when the figure in question is significantly narrower than the text width.)

A word of warning: Wrapping figures in LaTex will require a lot of manual adjustment of your document. There are several packages available for the task, but none of them works perfectly. Before you make the choice of including figures with text wrapping in your document, make sure you have considered all the options. For example, you could use a layout with two columns for your documents and have no text-wrapping at all.

Anyway, we will look at the package **wrapfig**. Note that **wrapfig** may not come with the default installation of LaTeX ; you might need to install additional packages.

However like we mentioned to wrap text around figures you first have to load the **wrapfig** in your preamble by using the the following command in the preamble:

```
\usepackage{wrapfig}
```

This then gives you access to:

```
\begin{wrapfigure}[lineheight]{position}{width}
```

There are overall eight possible positioning targets:

Characters		Corresponding Targets
r	R	right side of the text
l	L	left side of the text
i	I	inside edge–near the binding (in a twoside document)
o	O	outside edge–far from the binding

The uppercase-character allows the figure to float, while the lowercase version means ”exactly here”. Now, we will use the same giraffe picture to illustrate an example of it.

The giraffe (*Giraffa camelopardalis*) is an African even-toed ungulate mammal, the tallest living terrestrial animal and the largest ruminant. Its species name refers to its camel-like appearance and the patches of color on its fur. Its chief distinguishing characteristics are its extremely long neck and legs, its horn-like ossicones, and its distinctive coat patterns. It stands 5–6 m (16–20 ft) tall and has an average weight of 1,600 kg (3,500 lb) for males and 830 kg (1,800 lb) for females. It is classified under the family Giraffidae, along with its closest extant relative, the okapi. The nine subspecies are distinguished by their coat patterns. The giraffe's scattered range extends from Chad in the north to South Africa in the south, and from Niger in the west to Somalia in the east. Giraffes usually inhabit savannas, grasslands, and open woodlands. Their primary food source is acacia leaves, which they browse at heights most other herbivores cannot reach. Giraffes are preyed on by lions, and calves are also targeted by leopards, spotted hyenas, and wild dogs. Adult giraffes do not have strong social bonds, though they do gather in loose aggregations if they happen to be moving in the same general direction. Males establish social hierarchies through "necking", which are combat bouts where the neck is used as a weapon. Dominant males gain mating access to females, which bear the sole responsibility for raising the young.The giraffe has intrigued various cultures, both ancient and modern, for its peculiar appearance, and has often been featured in paintings, books, and cartoons. It is classified by the International Union for Conservation of Nature as Least Concern, but has been extirpated from many parts of its former range, and some subspecies are classified as Endangered. Nevertheless, giraffes are still found in numerous national parks and game reserves.

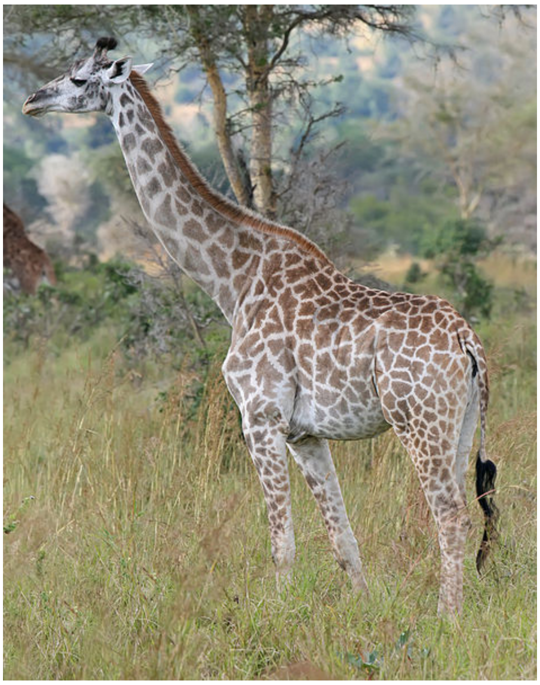


Figure 6: A giraffe

To create such thing, we use the following commands:

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \begin{center}
    \includegraphics[scale=0.60,width=0.48\textwidth]{Giraffe}
  \end{center}
  \caption{A giraffe}
```

we placed it between the text. We placed the used code between the first paragraph(which ends with ...are distinguished by their coat patterns) and second paragraph(which starts with The giraffe's scattered range ...).

Note that we have specified a size for both the wrapfigure environment and the image we have included. We did it in terms of the text width: it is always better to use relative sizes in LaTeX, let LaTeX do the work for you! The ”wrap” is slightly bigger than the picture, so the compiler will not return any strange warning and you will have a small white frame between the image and the surrounding text. You can change it to get a better result, but if you don't keep the image smaller than the ”wrap”, you will see the image over the text.

The **wrapfig** package can also be used with user-defined floats with float package.

Tip for figures with too much white space: It happens that you'll generate figures with too much (or too little) white space on the top or bottom. In such a case, you can simply make use of the optional argument [lineheight]. It specifies the height of the figure in number of lines of text. Also remember that the environment center adds some extra white space at its top and bottom; consider using the command **\centering** instead. Another possibility is adding space within the float using the **vspace{...}** command. The argument is the size of the space you want to add, you can use any unit you want, including pt, mm, in, etc. If you provide a negative argument, it will add a negative space, thus removing some white space. Using **\vspace** tends to move the caption relative to the float while the [lineheight] argument does not. Here is an example using the **\vspace** command, the code is exactly the one of the previous case, we just added some negative vertical spaces to shrink everything up:

Now cosider our previous example while we wrap some text around the picture of a giraffe. Suppose the wrapping process leaves some unwanted whitespace.(Though it soes not!)Then we could use the following command to shrink that extra white space:

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \vspace{-20pt}
  \begin{center}
    \includegraphics[scale=0.60,width=0.48\textwidth]{Giraffe}
  \end{center}
  \vspace{-20pt}
  \caption{A giraffe}
  \vspace{-17pt}
\end{wrapfigure}
```

it will produce output like what you see in the next page.
In this case it may look too shrunk, but you can manage spaces the way you like. In general, it is best not to add any space at all: let LaTeX do the formatting work!

(In this case, the problem is the use of **\begin{center}** to center the image. The center environment adds extra space that can be avoided if **\centering** is used instead.) You can use intextsep parameter to control additional space above and below the figure: **\setlength\intextsep{0pt}**
Alternatively you might use the picins package instead of the wrapfig package which produces a correct version without the excess white space out of the box without any hand tuning. There is also an alternative to wrapfig: the package **floatflt**.

The giraffe (*Giraffa camelopardalis*) is an African even-toed ungulate mammal, the tallest living terrestrial animal and the largest ruminant. Its species name refers to its camel-like appearance and the patches of color on its fur. Its chief distinguishing characteristics are its extremely long neck and legs, its horn-like ossicones, and its distinctive coat patterns. It stands 5–6 m (16–20 ft) tall and has an average weight of 1,600 kg (3,500 lb) for males and 830 kg (1,800 lb) for females. It is classified under the family Giraffidae, along with its closest extant relative, the okapi. The nine subspecies are distinguished by their coat patterns. The giraffe's scattered range extends from Chad in the north to South Africa in the south, and from Niger in the west to Somalia in the east. Giraffes usually inhabit savannas, grasslands, and open woodlands. Their primary food source is acacia leaves, which they browse at heights most other herbivores cannot reach. Giraffes are preyed on by lions, and calves are also targeted by leopards, spotted hyenas, and wild dogs. Adult giraffes do not have strong social bonds, though they do gather in loose aggregations if they happen to be moving in the same general direction. Males establish social hierarchies through "necking", which are combat bouts where the neck is used as a weapon. Dominant males gain mating access to females, which bear the sole responsibility for raising the young.The giraffe has intrigued various cultures, both ancient and modern, for its peculiar appearance, and has often been featured in paintings, books, and cartoons. It is classified by the International Union for Conservation of Nature as Least Concern, but has been extirpated from many parts of its former range, and some subspecies are classified as Endangered. Nevertheless, giraffes are still found in numerous national parks and game reserves.

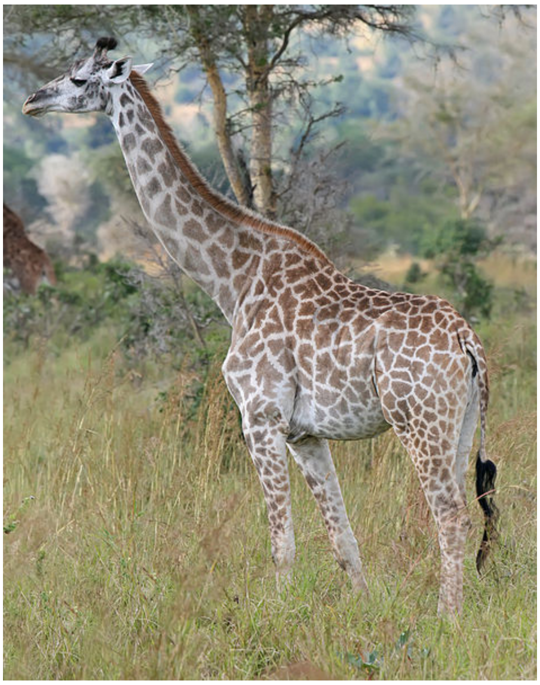


Figure 7: A giraffe

6.7 Sub-float:

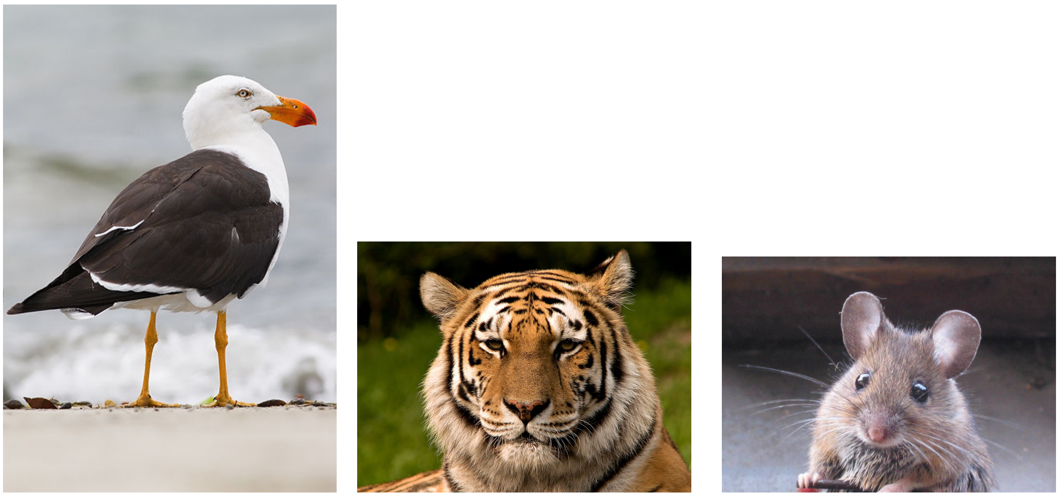
6.7.1 Use of subcaption package:

Use of subfigure environment: A useful extension is the **subcaption** package (the **subfigure** and **subfig** packages are deprecated and shouldn't be used any more), which uses subfloats within a single float. This gives the author the ability to have subfigures within figures, or subtables within table floats. Subfloats have their own caption, and an optional global caption. An example will best illustrate the usage of this package:

```
\begin{figure}
  \centering
   \begin{subfigure}[b]{0.3\textwidth}
    \includegraphics[scale=0.40,width=\textwidth]
{Gull}
     \caption{A gull}
     \label{fig:gull}
      \end{subfigure}%
~ %add desired spacing between images, e. g. ~, \quad,
\qquad etc.
%(or a blank line to force the subfigure onto a new line)
        \begin{subfigure}[b]{0.3\textwidth}
       \includegraphics[scale=0.40,width=\textwidth]
{Tiger}
        \caption{A tiger}
        \label{fig:tiger}
        \end{subfigure}
~ %add desired spacing between images, e. g. ~, \quad,
\qquad etc.
%(or a blank line to force the subfigure onto a new line)
        \begin{subfigure}[b]{0.3\textwidth}
        \includegraphics[scale=0.40,width=\textwidth]
{Mouse}
       \caption{A mouse}
      \label{fig:mouse}
     \end{subfigure}
    \caption{Pictures of animals
    \label{fig:animals}
```

```
\end{figure}
```

If you use the given code, you will get an output like what you are seeing in the next page. Now, the width and height of the pictures are different. So, we get an output like that. If you have pictures which are of same height and width you get a better output.



(a) A gull

(b) A tiger

(c) A mouse

Figure 8: Pictures of animals

Use of minipage envireonment: However, you could get the same output by using **minipage** environment. Though you have to use the same packages, i.e **caption & subcaption**.If we want the same output using the **minipage** environment,we have to use the following command:

```
\begin{figure}[ht]
\centering
\begin{minipage}[b]{0.30\linewidth}
 \includegraphics[scale=0.40,width=\textwidth]{Gull}
 \caption{A Gull}
 \label{fig:gull}
 \end{minipage}
 \quad
```

```
\begin{minipage}[b]{0.30\linewidth}
\includegraphics[scale=0.40,width=\textwidth]{Tiger}
\caption{A tiger}
\label{fig:tiger}
\end{minipage}
```

continued to next page

continued from previous page

```
 \begin{minipage}[b]{0.30\linewidth}
 \includegraphics[scale=0.40,width=\textwidth]{Mouse}
 \caption{A mouse}
\label{fig:mouse}
 \end{minipage}
\caption{Pictures of animals}
\label{fig:animals}
\end{figure}
```

It will produce output like:

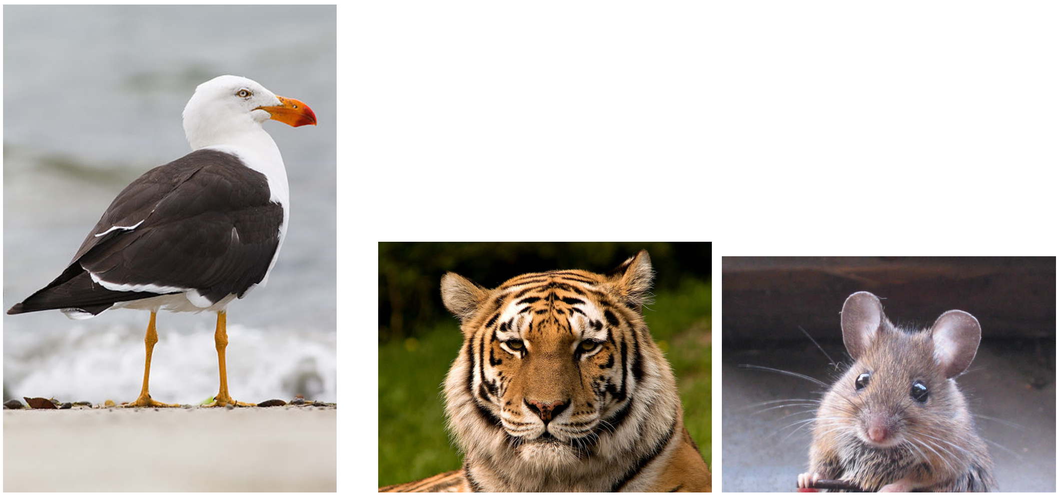


Figure 9: A Gull

Figure 10: A tiger

Figure 11: A mouse

Figure 12: Pictures of animals

Note:- The numberings in caption can be omitted by not using **label** and by using **caption*** instead of **caption**. Now we need to know the meanings to **linewidth & textwidth**.
linewidth: the width of the current line in a paragraph.
textwidth: the normal horizontal dimension of the body of the page.

If you intend to cross-reference any of the subfloats, see where the label is inserted; **caption** outside the subfigure-environment will provide the global caption.

subcaption will arrange the figures or tables side-by-side providing they can fit, otherwise, it will automatically shift subfloats below. This effect can be added manually, by putting the newline command (\\) before the figure you wish to move to a newline.

Horizontal spaces between figures are controlled by one of several commands, which are placed in between **\begin{subfigure}** and **\end{subfigure}**:

- A non-breaking space (specified by   as in the example above) can be used to insert a space in between the subfigs.
- Math spaces: **\qquad, \quad, \;**, and **\,**
- Generic space: **\hspace{"length"}** Automatically expanding/contracting space: **\hfill**

6.8 Wide figures in two column documents:

If you are writing a document using two columns (i.e. you started your document with something like **\documentclass**[**twocolumn**]**{article}**), you might have noticed that you can't use floating elements that are wider than the width of a column (using a LaTeX notation, wider than **0.5\textwidth**), otherwise you will see the image overlapping with text. If you really have to use such wide elements, the only solution is to use the "starred" variants of the floating environments, that are **{figure*}** and **{table*}**. Those "starred" versions work like the standard ones, but they will be as wide as the page, so you will get no overlapping.

textbackslashTo prevent the figures from being placed out-of-order with respect to their "non-starred" counterparts, the package fixltx2e should be used (e.g. **\usepackage{fixltx2e}**).

6.9 Creating a sidewaysfigure:

The package **rotating** provides you the following facility.

This package provides you the environment **sidewaysfigure** which make a float sideways so that the caption is also rotated.

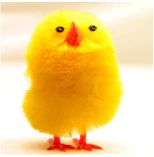
7 Seamless text integration:

The drawback of importing graphics that were generated with a third-party tool is that font and size will not match with the rest of the document. There are still some workarounds though.

The easiest solution is to use the picture environment and then simply use the ”put” command to put a graphics file inside the picture, along with any other desired LaTeX element. For example:

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
\put(3.5,0.4){$\displaystyle
s:=\frac{a+b+c}{2}$}
\put(1,1){\includegraphics[
  width=2cm,height=2cm]{Chick.png} }
\end{picture}
```

it will produce output like:



$$s := \frac{a+b+c}{2}$$

Note that the border around the picture in the above example was added by using **\fbox**, so the contents of the border is the picture as generated by the above code. Tools like Inkscape or Xfig have a dedicated LaTeX export feature that will let you use correct font and size for text in vector graphics. See #Third-party graphics tools.

For a perfect integration of graphics, you might consider procedural graphics capabilities of some LaTeX packages like TikZ or PSTricks. It lets you draw from within a document source. While the learning curve is steeper, it is worth it most of the time.

8 References:

- `http://en.wikibooks.org/wiki/LaTeX`
- `tex.stackexchange.com`
- graphics bundle documentation by **D. P. Carlisle**
- graphicx package documentation by **T. J. Mahoney**