# LaTeX: Online module 6

Venkata Manem

Univ. of Waterloo

August 4, 2011

# Topics to be covered

- Introduction
- Image formats
- Importing graphics
- Wrap figure
- Errors
- Exercises

# Introduction

- Presenting figures in a word document becomes challenging which might lead to unnecessary page breaks
- This requires to tune the document manually
- But, figures automatically move to suitable locations and the positioning of figures is taken care off by LaTeX
- LaTeX cannot handle figures directly, so we have to use the graphicx package
- Use the command \usepackage{graphicx} in the preamble of your document

# Figure environment

```
\begin{figure}[hbtp]
\caption{Figure name}
\begin{center}
\includegraphics{filename with extension}
\end{center}
\label{For further references}
\end{figure}
```

# contd...

Figure environment should always start and end with the following commands:

```
\begin{figure}
⋮
\end{figure}
```

# Figure position

- User can use the following options to specify the position of the figure to LaTeX:
  1. h: places the figure in the text where the figure command is located (note that this option cannot be executed if there is not enough space left on the page)
  2. t: places the figure on top of a page
  3. b: places the figure at the bottom of a page
  4. p: places the figure on a page containing only floats

# contd...

- By default LaTeX takes the options in the following order: h, b, t, p
- If you want the figure to appear in the current position of your LaTeX file then we use the option h only:

$$\backslash\text{begin}\{\text{figure}\}[h]$$
$$\vdots$$
$$\backslash\text{end}\{\text{figure}\}$$

# contd...

- Centering commands should be enclosed within the figure environment. One of the following commands can be used:

$$\backslash\text{begin}\{\text{center}\}$$
$$\vdots$$
$$\backslash\text{end}\{\text{center}\}$$

or, you can use the following command after $\backslash\text{begin}\{\text{figure}\}[h]$ :

$$\backslash\text{centering}$$

# Example

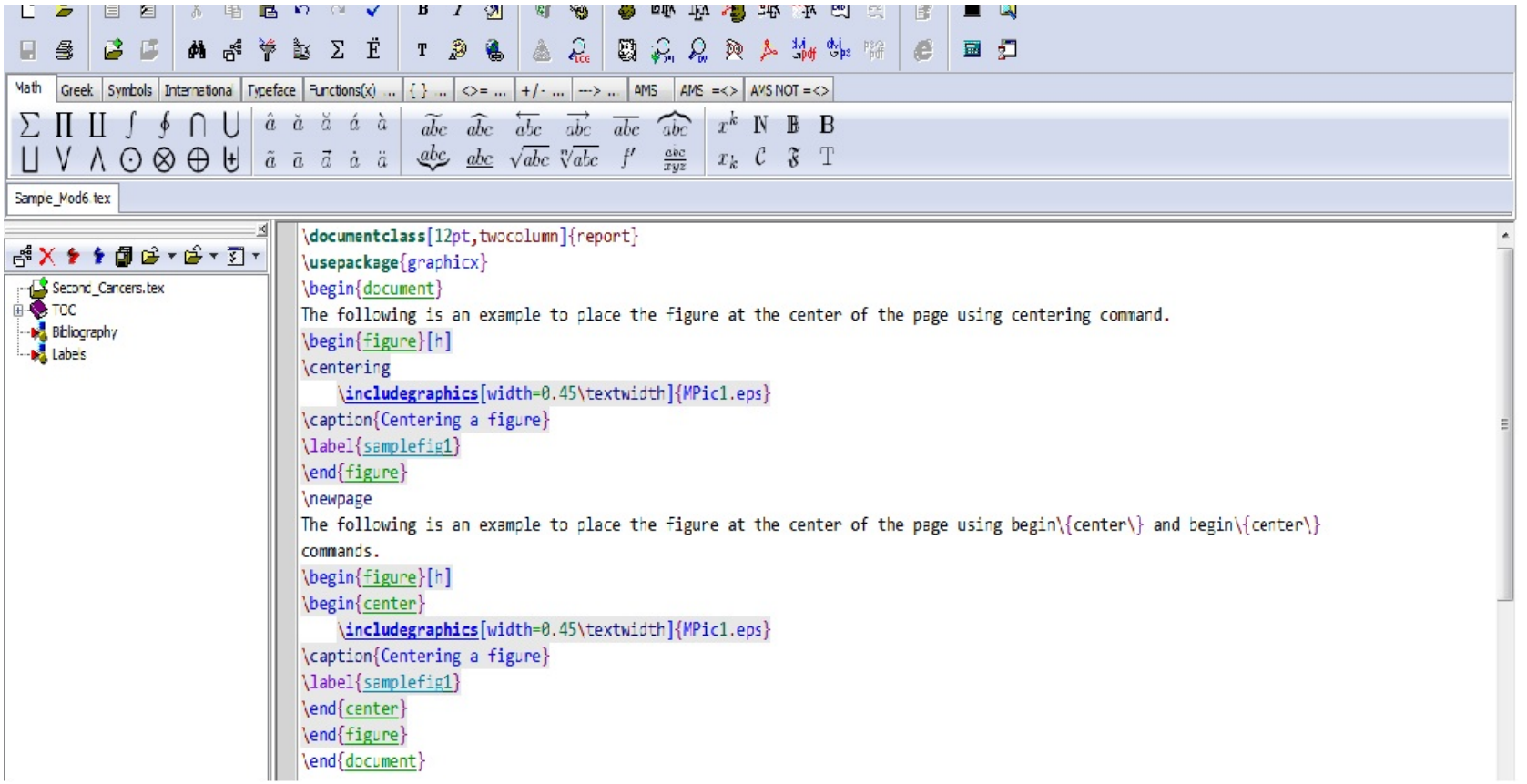


Fig A: Example for using Centering, h options in figure environment

# contd...

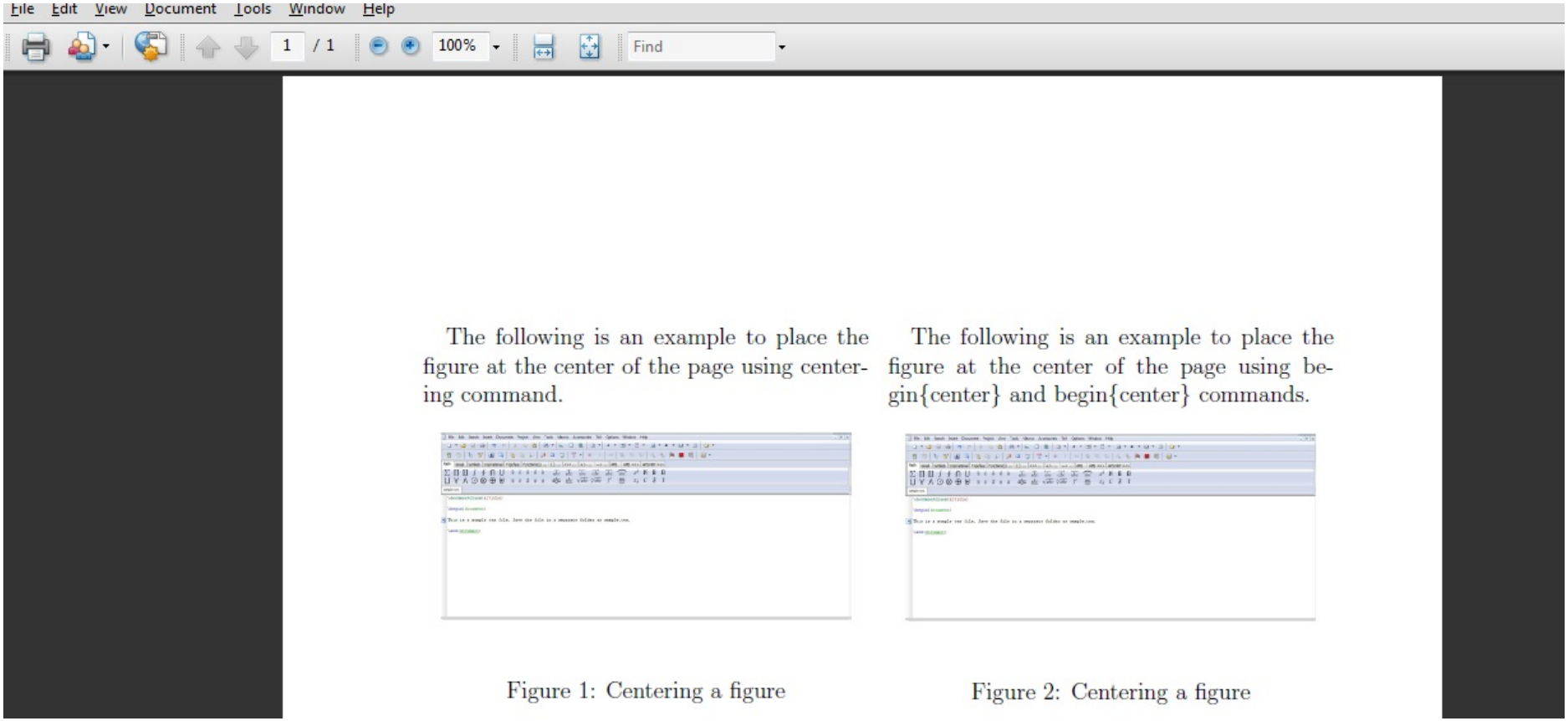


Fig B: Output of the previous slide using different commands to center a figure (note that two column format is used)

# Command: \include{graphics}

- This command inserts the figure at the current location where the code is written
- User can use the following options within the command:

  \include{graphics}[options]{filename}
- Options can be any one of the following:
  1. width
  2. height
  3. scale
  4. angle

# Sample code to define width

Specifies the width of a figure

```
\begin{figure}[h]
\begin{center}
    \includegraphics[width=0.75\textwidth]{MPic1.eps}
\caption{Defining width of a figure}
\label{samplefig1}
\end{center}
\end{figure}
```

# contd...

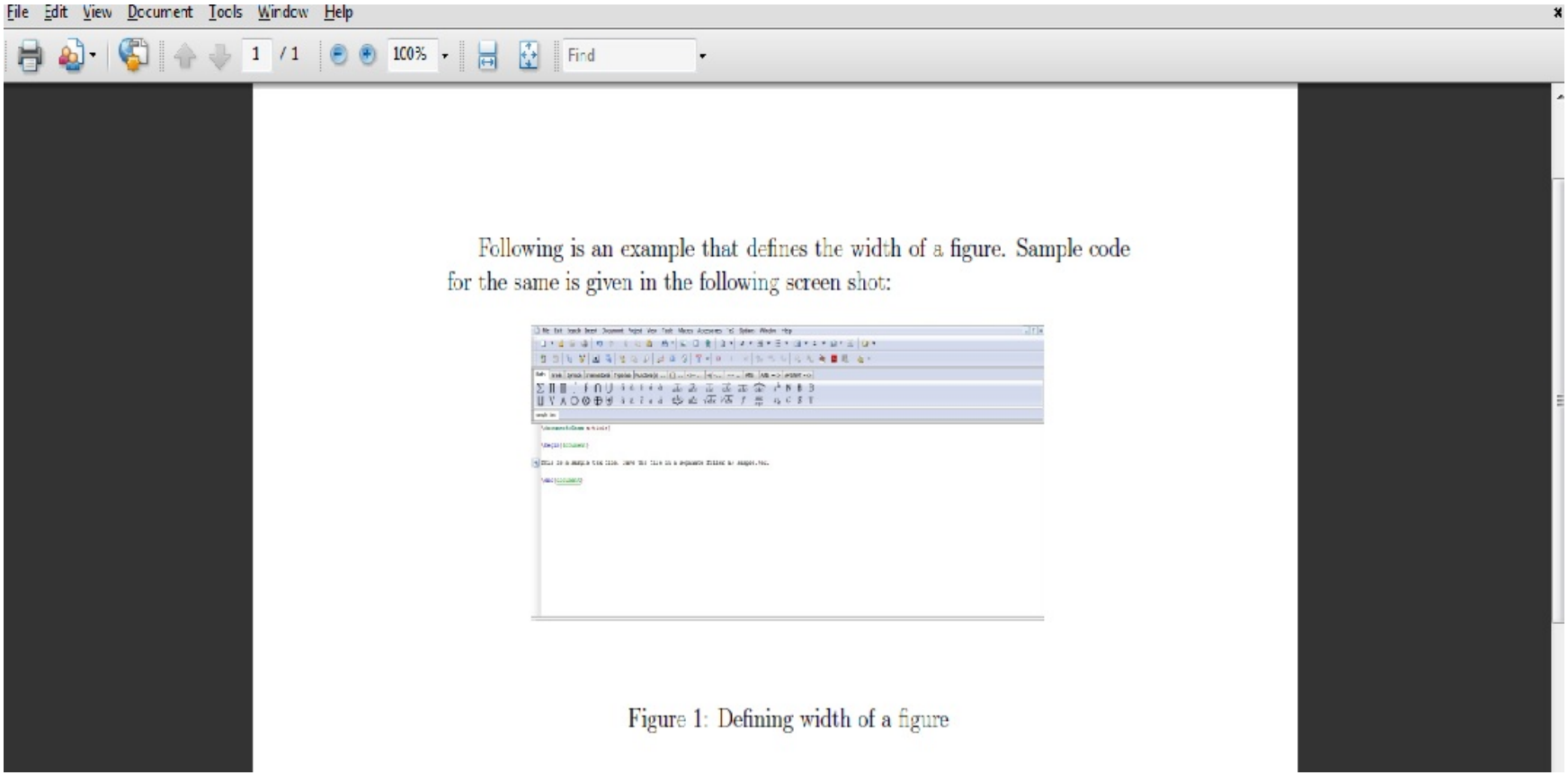


Fig C: Output of the previous slide that defined width of a figure

# Sample code to define height

Specifies the height of a figure

```
\begin{figure}[h]
\centering
    \includegraphics[height=3in]{MPic1.eps}
\caption{Defining height of a figure}
\label{samplefig1}
\end{figure}
```

# contd...

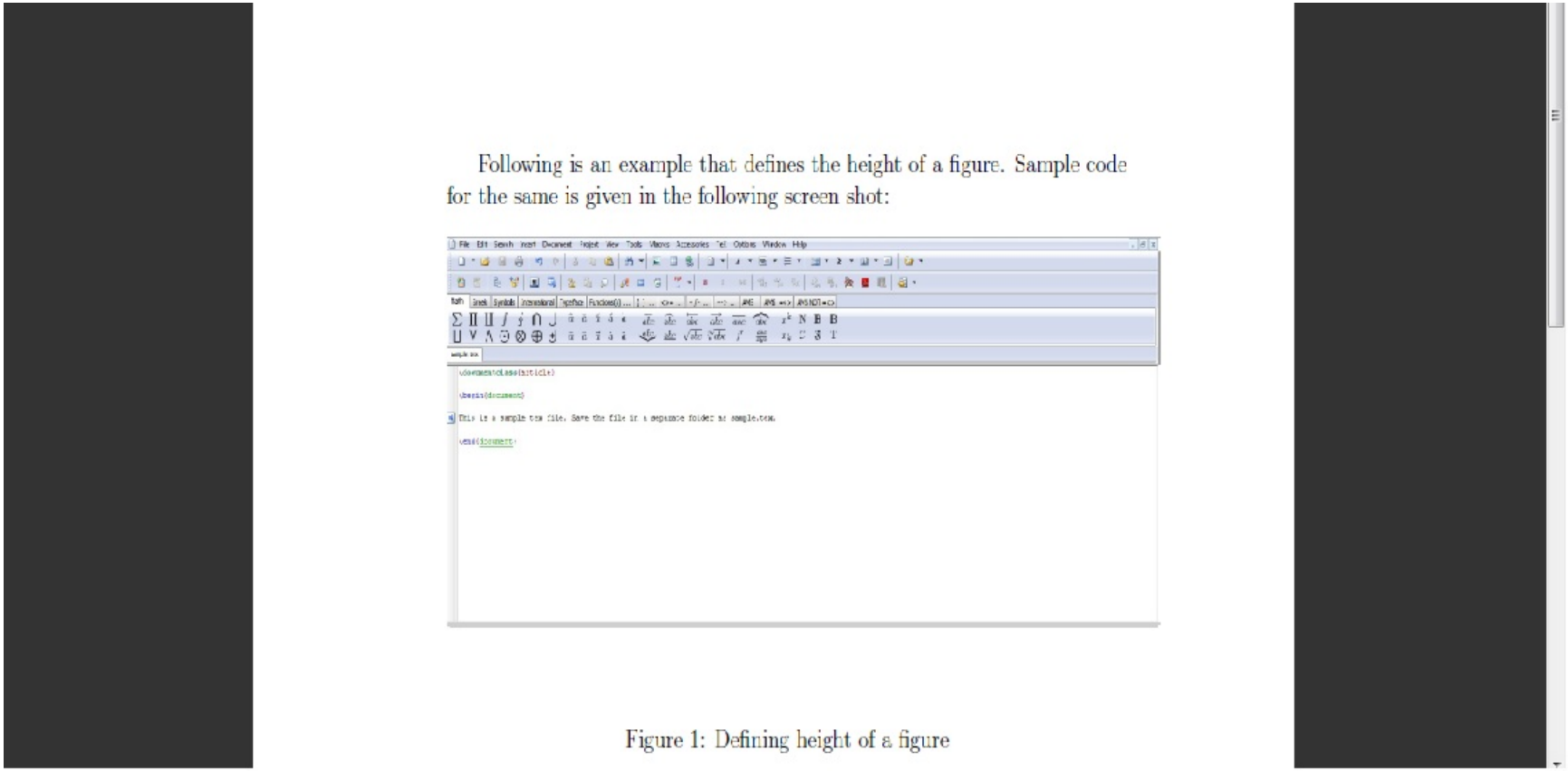


Fig D: Output of the previous slide that defined height of a figure

# Sample code to define scale

Specifies the scale of a figure. This command acts as a scale factor for the figure

```
\begin{figure}[h]
\centering
    \includegraphics[scale=0.5]{MPic1.eps}
\caption{Defining scale of a figure}
\label{samplefig1}
\end{figure}
```

**Note**: One should be very cautious in using this command as the figure may extend over its boundaries

# contd...

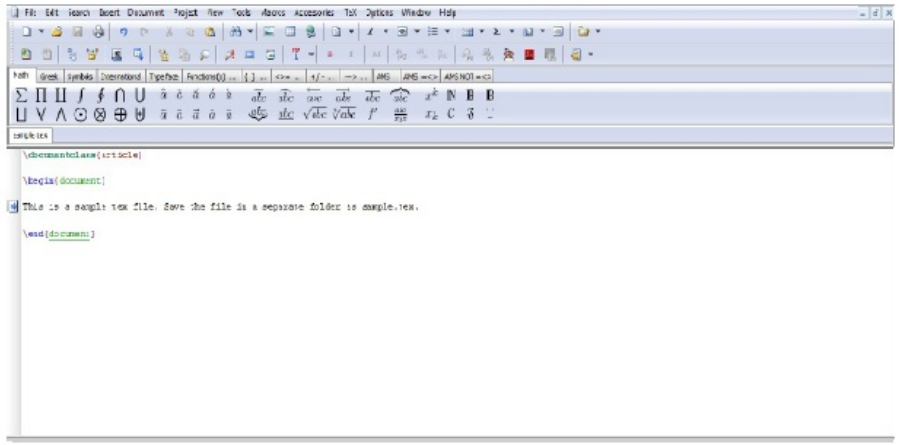

Figure 1: Defining scale of a figure

Fig E: Output of the previous slide that defined scale of a figure

# Sample code to define angle of rotation

User can specify the angle of rotation in degrees. Below is a sample code that displays the figure rotated by 45 degrees

```
\begin{figure}[h]
\centering
    \includegraphics[height=2in,angle=45]{MPic1.eps}
\caption{Defining angle of rotation of a figure}
\label{samplefig1}
\end{figure}
```

# contd...

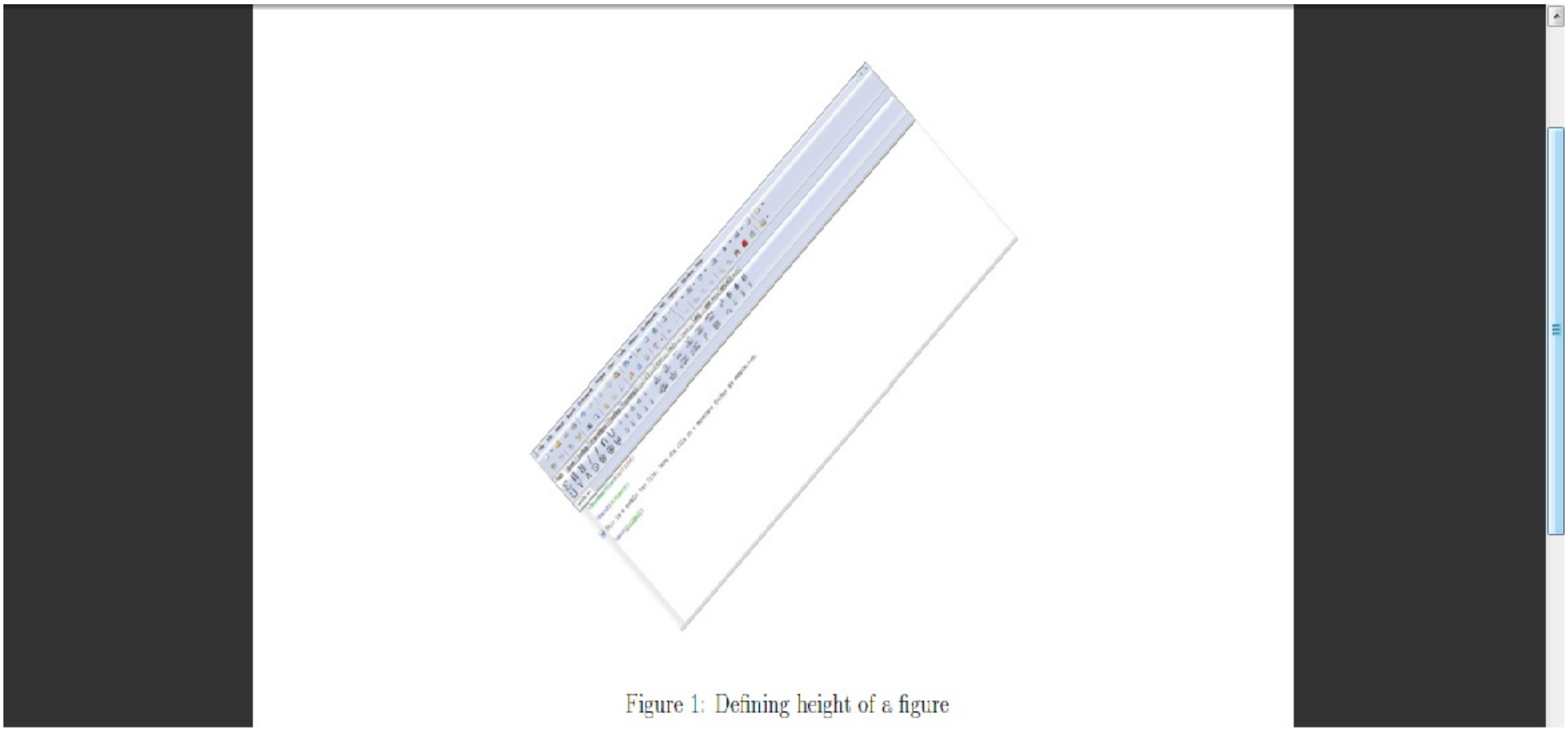


Fig F: Output of the previous slide that defined angle of a figure

# Caption and label

- \caption{figure name}: Name of the figure can be specified
- \label{reference name}: A label is defined by the user so that it can be used for reference in any part of the document

# Border lining

- If the user wishes to have a figure enclosed within a thin border line, following packages along with few commands have to be included in the preamble of the document:

```
\usepackage{float}
\floatstyle{boxed}
\restylefloat{figure}
```

# Sample code to to draw a border line

```
\documentclass[12pt]{report}
\usepackage{graphicx}
\usepackage{float}
\floatstyle{boxed}
\restylefloat{figure}
\begin{document}
\begin{figure}[h]
\centering
    \includegraphics[height=2in,angle=45]{MPic1.eps}
\caption{Defining height of a figure}
\label{samplefig1}
\end{figure}
\end{document}
```

# contd...

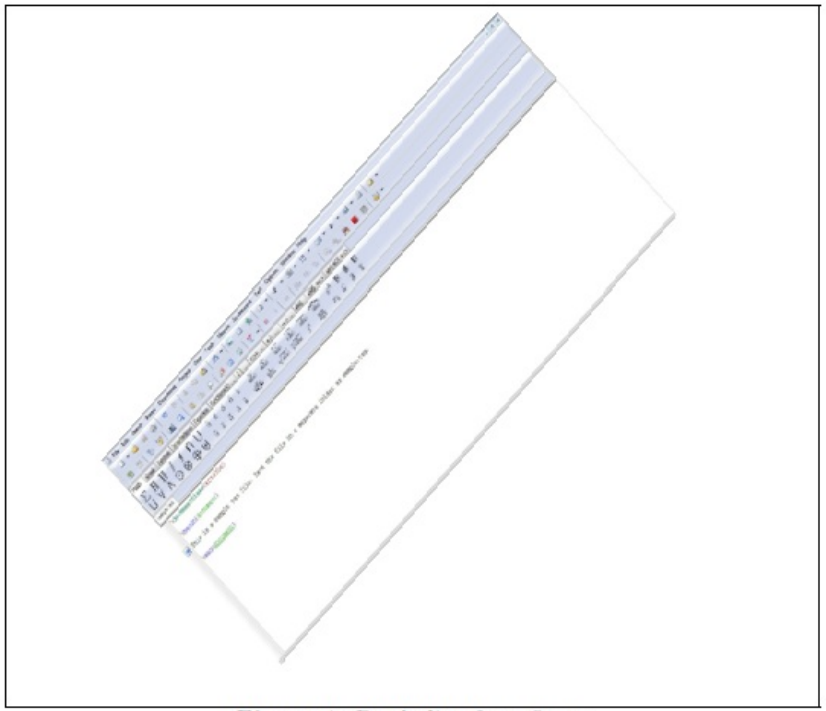

Figure 1: Borderline for a figure

Fig G: Output of the previous slide that defined border for a figure

# Generating PDF documents

- There are two approaches in generating a PDF document from a LaTeX file that has figures of various formats
- Extensions for figures can be any of the following: .eps, .jpg, .png, .pdf, *.pstex_t*
- Depending on the type of figure that we include in the LaTeX file, we use two approaches in generating a PDF document

# Approach 1

- Figures should be in .eps (encapsulated post script) for **compiling** a LaTeX document
- This .eps figure format was defined by Adobe to import graphics into LaTeX documents
- An .eps file declares its own size for the figure which is an advantage for LaTeX

# contd...

- Most of the software packages handle .eps figure formats (as this is one of the extension that Adobe has provided)
- If any of softwares does not handle the .eps extension then you can do the following to convert any figure to an .eps format
- In linux **convert** is the command used to convert any figure format to an .eps extension format. In the following example, convert command takes input as sample.jpg and converts it to sample.eps

  *convert sample.jpg sample.eps*
- Approach to generate pdf documents is shown in the following slides

# contd...

Following are the steps to generate a pdf document when .eps figures are included in your LaTeX file:

- Compile the file using LaTeX option available on the tool bar
- User can view the document in DVI format by clicking on the DVI button on the tool bar
- A PDF document is then generated by clicking on DVIPDF button

Compile$\dashrightarrow$View DVI$\dashrightarrow$DVIPDF

Compile$\dashrightarrow$DVIPDF

(user can directly generate a PDF without looking at DVI format)

# Example

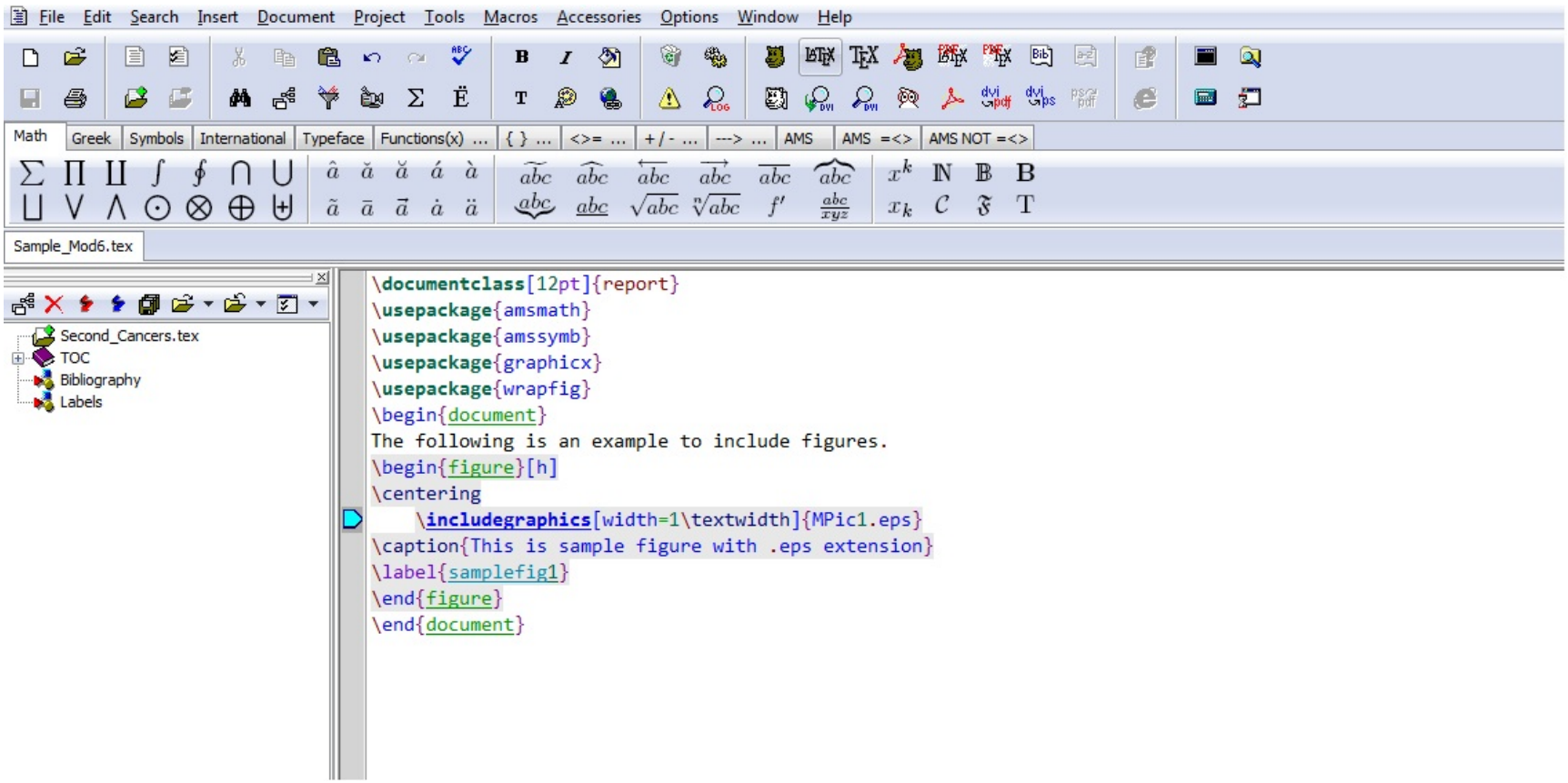


Fig 1: Compile option is enabled in the above snap shot

# contd...

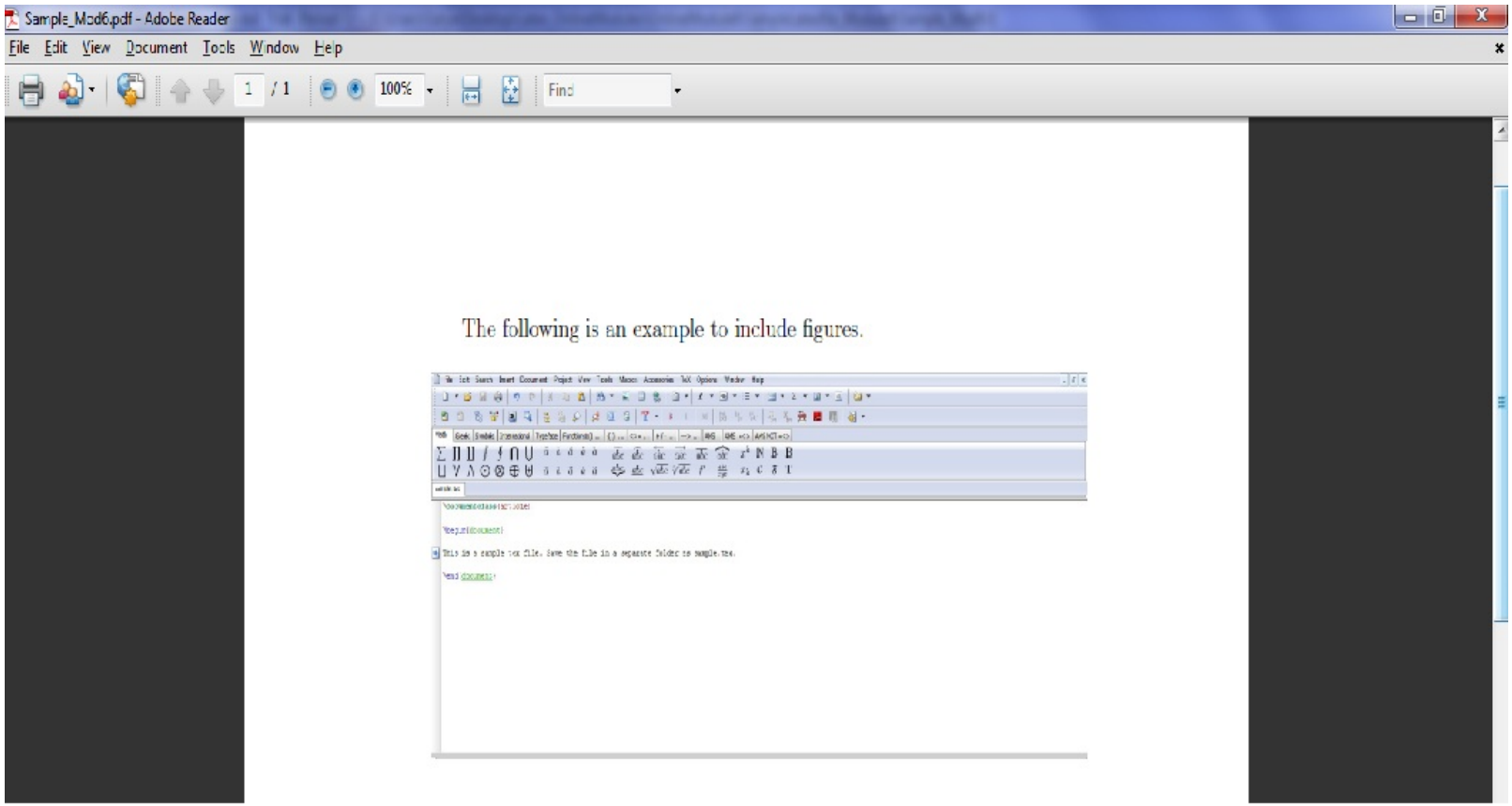


Fig 2: Output is generated by clicking on *dvipdf* button on the tool bar

# Approach 2

Following are the steps to generate a pdf document when figures with .jpg, .png extensions are included in your LaTeX file:

- Click on the *pdflatex* button on the tool bar
- A PDF document is then generated by clicking on Adobe button (notice that DVI format is not seen here)
- This approach is more flexible in handling lots more image formats than compiling a LaTeX file to generate a pdf document
- Observe that LATEX option (used in approach 1 for compilation) is replaced by PDFLATEX in this approach

PDFLATEX$\dashrightarrow$ADOBE

# Example

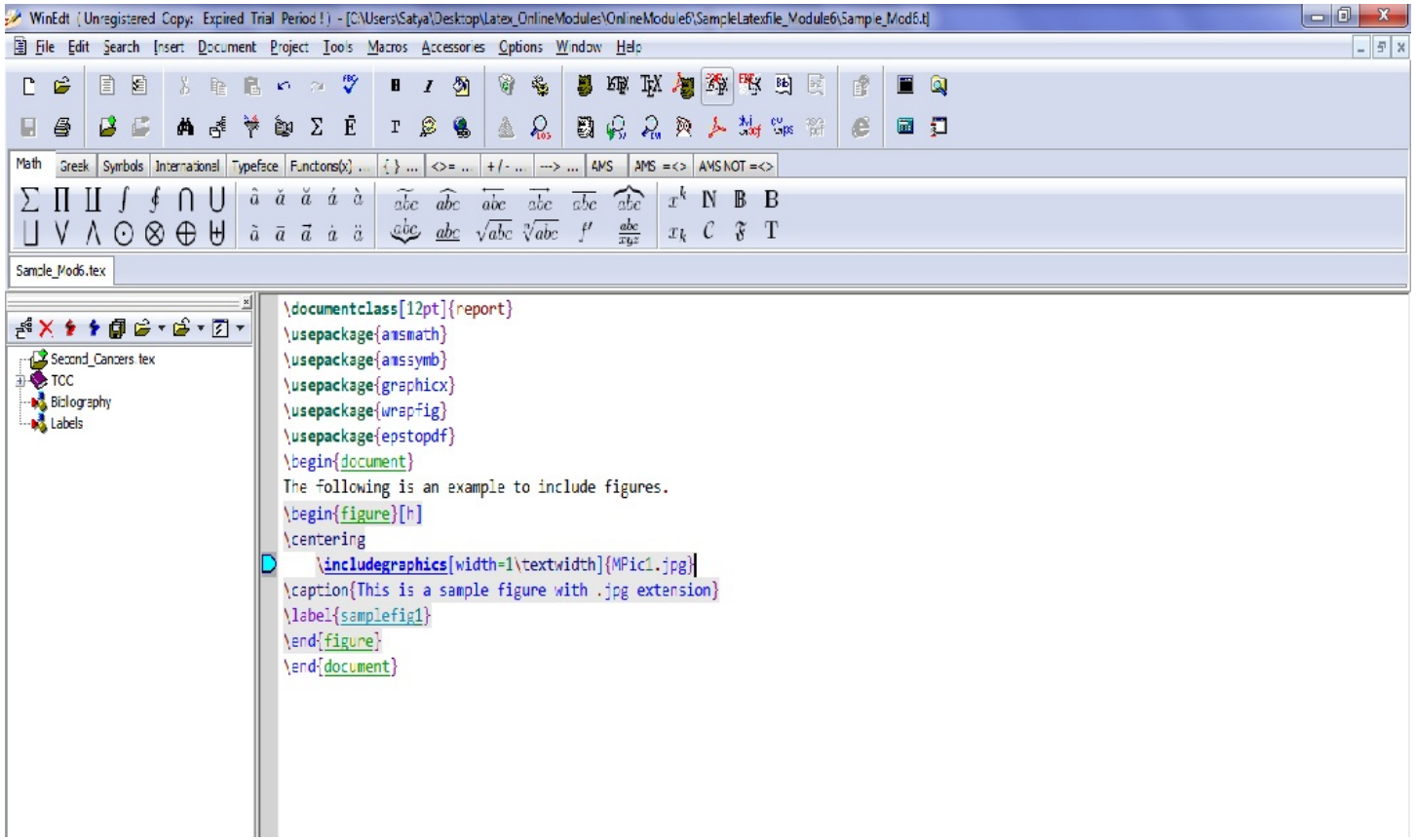


Fig 3: *pdflatex* button is enabled in the above Snap shot (for .jpg format)

# contd...

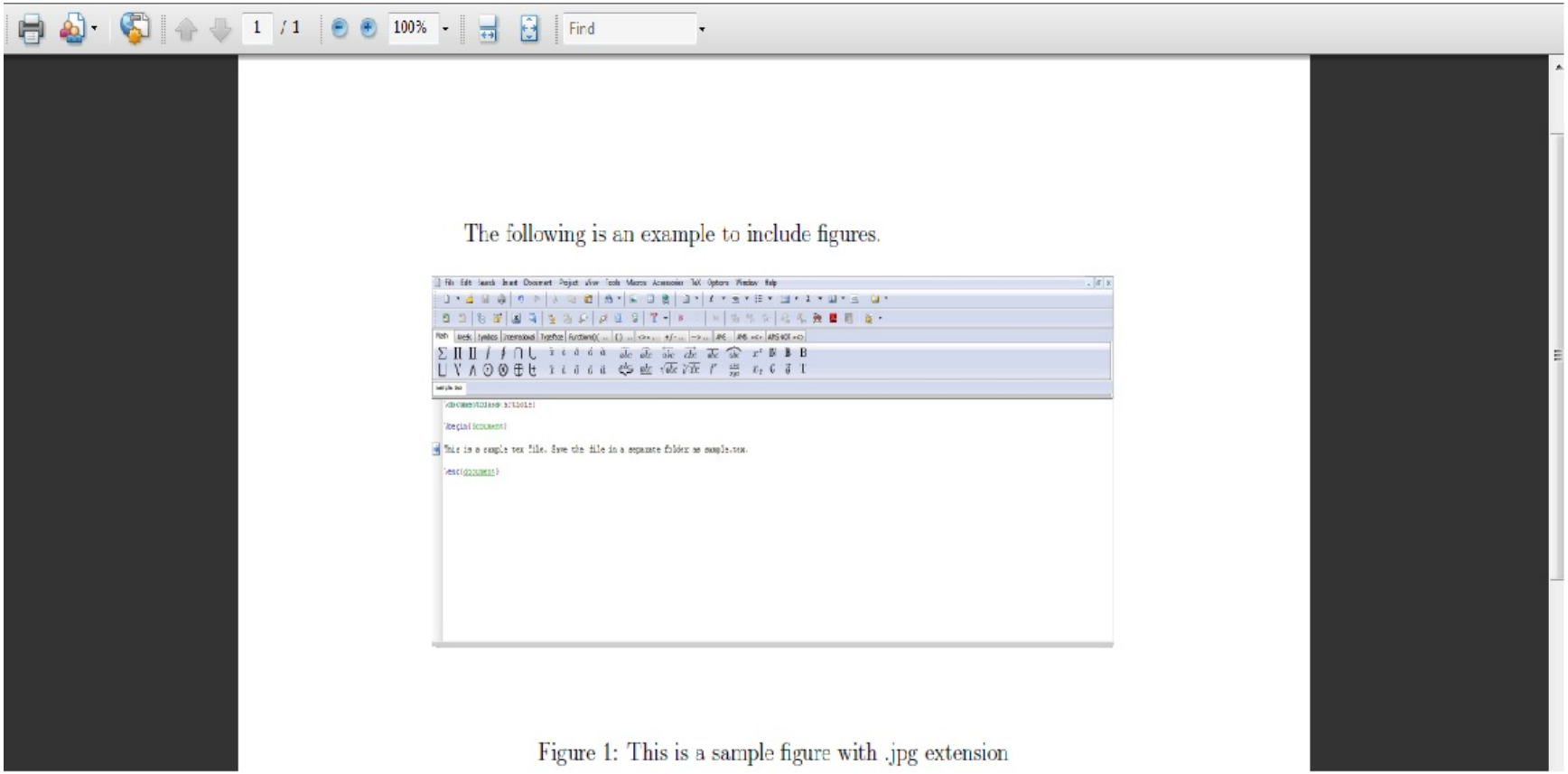


Fig 4: Output is generated by clicking on *adobe* button on the tool bar

# Conversion to various formats

- If the user is comfortable in using approach 1 to generate PDF documents but have a .jpg or a .png file, then use the following commands to convert the image format:
  1. On Linux platform:

     *convert samplefig.jpg samplefig.eps*
  2. On Windows platform: You can use the command line interface (cmd prompt) and type in the command

     *bmeps -c samplefig.png samplefig.eps*

  **Make sure the figure is located in your current working directory**

# Errors

Errors that one might encounter during compilation are the following:

- Error: Unknown graphics extension
  1. Cause1: Graphics file is not saved in the current working folder
  2. Solution1: Change the location of the graphics file to the current working folder
  3. Cause2: If your LaTeX file has a .eps file and you try to generate a PDF document using approach 2 (pdflatex)
  4. Solution2: Use approach 1: Compile--→DVIPDF

# Example

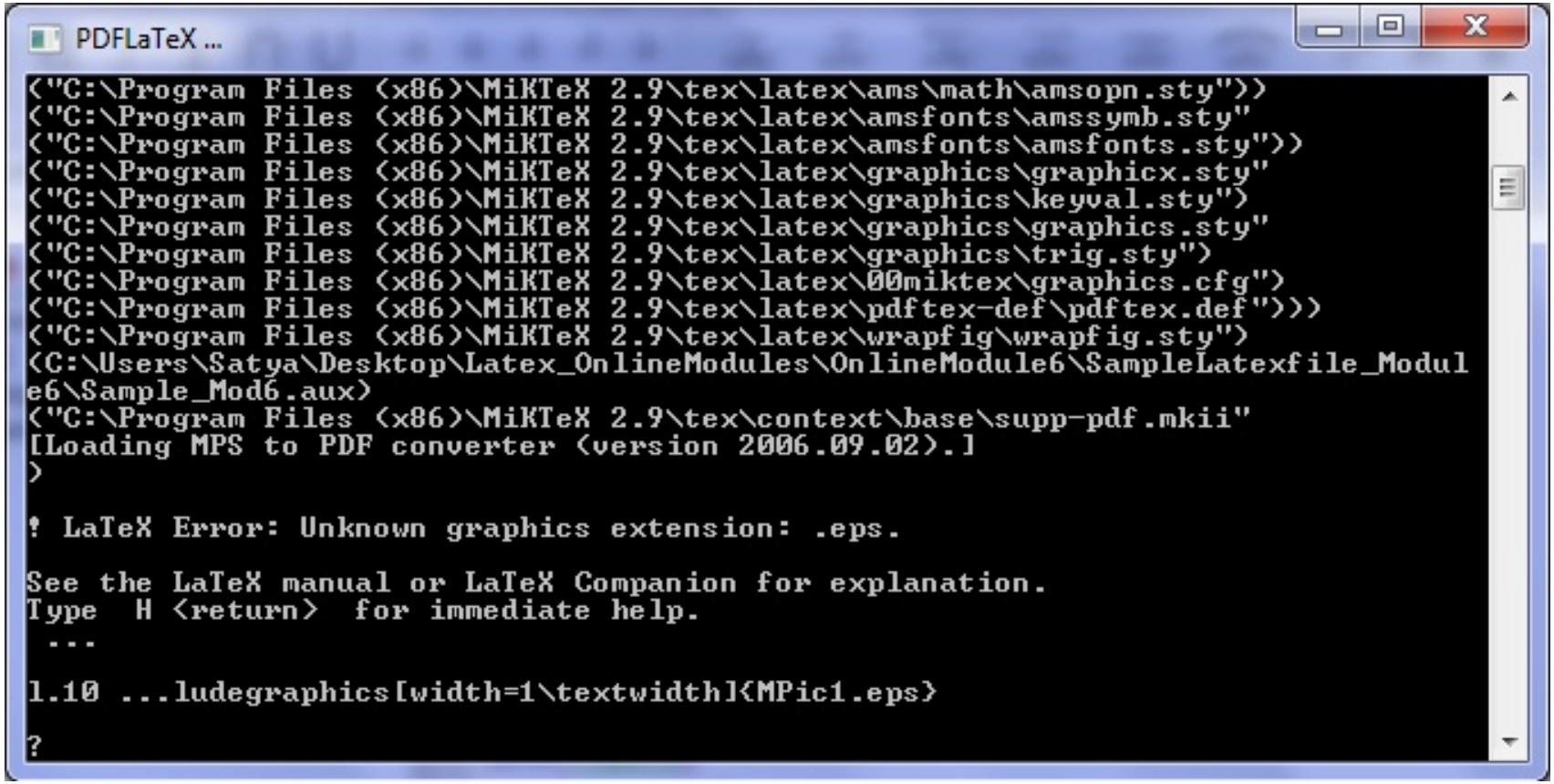


Fig 5: Snap shot of *Errors one can encounter during compilation*

# Wrap figure

- User might want to wrap some text with a figure on a page, then following package has to be added in the preamble of the document:

  \usepackage{*wrapfig*}

- Figure environment should be enclosed with the commands

  \begin{*wrapfigure*}{*alignment*}{*width*}
  \end{*wrapfigure*}

- Position of the figure has to be adjusted manually to fit exactly in the page (which might be tiresome at times if there is not enough space remaining on the page)

# Sample code to to draw a border line

Note: right alignment is used

```
\documentclass[12pt]{report}
\usepackage{graphicx}
\usepackage{wrapfig}
\begin{document}
\begin{wrapfigure}{r}{0.5\textwidth}
\centering
    \includegraphics[width=0.48\textwidth]{hagey.jpg}
    \caption{Example for wrap figure}
\vspace{-15pt}
\end{wrapfigure}
\noindent
The University of Waterloo celebrated its 50th anniversary in
2007,harking back to the first applied science classes offere
-d starting July 1, 1957, by the temporary Waterloo College..
\end{document}
```

# contd...

Observe that the command vspace is used in the sample code to manually adjust the content and figure on a page

The University of Waterloo celebrated its 50th anniversary in 2007, harking back to the first applied science classes offered starting July 1, 1957, by the temporary Waterloo College Associate Faculties. Gerald Hagey (left), president of the Lutheran-affiliated Waterloo College, had enlisted support from business leaders in Kitchener-Waterloo, a modest industrial city in midwestern Ontario: they would create a program to train engineers and technicians who were desperately needed for Canadas growing postwar economy. The cold war, the space race, and medical and scientific progress presented new needs for trained manpower and technical knowledge. For Hagey and his colleagues, the solution was not just classroom instruction but the cooperative program, which offered students alternating terms of paid work in industry to get practical experience.

Figure 1: Example for wrap figure

Fig 6: Output of the previous slide (with vspace)

# contd...

Below is a output of the sample code if vspace is removed (note that there is lots of space between content and caption of the figure)

The University of Waterloo celebrated its 50th anniversary in 2007, harking back to the first applied science classes offered starting July 1, 1957, by the temporary Waterloo College Associate Faculties. Gerald Hagey (left), president of the Lutheran-affiliated Waterloo College, had enlisted support from business leaders in Kitchener-Waterloo, a modest industrial city in midwestern Ontario: they would create a program to train engineers and technicians who were desperately needed for Canadas growing postwar economy. The cold war, the space race, and medical and scientific progress presented new needs for trained manpower and technical knowledge. For Hagey and his colleagues, the solution was not just classroom instruction but the co-operative program, which offered students alternating terms of paid work in industry to get practical experience.

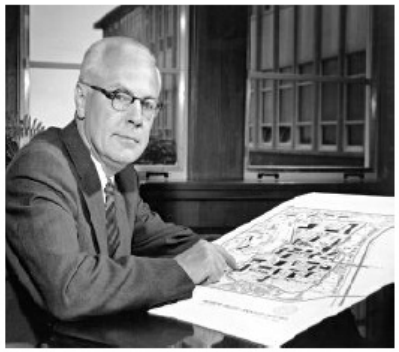

Figure 1: Example for wrap figure

Fig 7: Output of sample code (without vspace)

# Exercises

- Generate a pdf document with the following specifications (save a sample figure in .eps format):
  1. Which approach do you use to generate a pdf document?
  2. Figure should be centered
  3. Use the option **t**
  4. Use two column page format
- Repeat the same exercise by replacing the option **t** with **h**
- Convert the .eps figure to a different format and try to generate a pdf document using the above mentioned specifications

# contd...

- Try to generate a pdf document that has 4 figures (can be of any extension): use width, height, scale, angle of rotation for each figure and try to notice the difference
- Repeat the same exercise by wrapping the figure (make sure that the figure is enclosed within a border line)