

Tools for Getting Graphics Into \LaTeX

Part 1

Dan Drake

KAIST

Korea Advanced Institute of Science and Technology

13 August 2008 / Sage Days 9

Outline

1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Outline

1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Outline

1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Outline

1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Outline

1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Before we get started. . .

Tomorrow I'll talk more about specific pieces of software.
Try to think of questions like:

- What's a good way to make graphics like X / for Y?
- Is graphics program X good for use with \LaTeX ?

I will take questions at the end and answer them tomorrow. (Or maybe today.)

How does T_EX handle graphics?

- It doesn't.
- T_EX uses the `\special` macro to put magic bits in the output file.
- Vaguely similar to putting images on web pages with an `img` tag.

You needn't ever worry about this, but in the bad old days, one had to select the proper “driver”.

How does T_EX handle graphics?

- It doesn't.
- T_EX uses the `\special` macro to put magic bits in the output file.
 - Vaguely similar to putting images on web pages with an `img` tag.

You needn't ever worry about this, but in the bad old days, one had to select the proper “driver”.

How does T_EX handle graphics?

- It doesn't.
- T_EX uses the `\special` macro to put magic bits in the output file.
- Vaguely similar to putting images on web pages with an `img` tag.

You needn't ever worry about this, but in the bad old days, one had to select the proper “driver”.

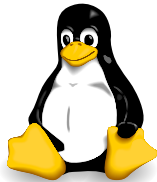
How does T_EX handle graphics?

- It doesn't.
- T_EX uses the `\special` macro to put magic bits in the output file.
- Vaguely similar to putting images on web pages with an `img` tag.

You needn't ever worry about this, but in the bad old days, one had to select the proper “driver”.

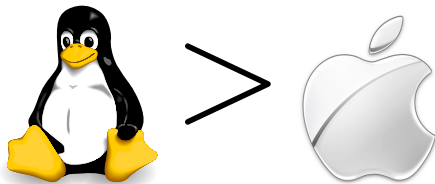
Admission of biases

(and level of knowledge)



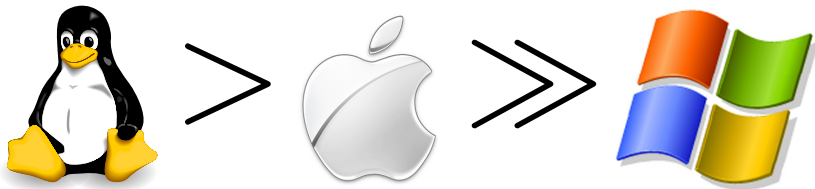
Admission of biases

(and level of knowledge)



Admission of biases

(and level of knowledge)



Two basic strategies

Two basic strategies for getting graphics into \LaTeX :

Strategy 1 use \LaTeX itself

Strategy *i* use something else

Outline

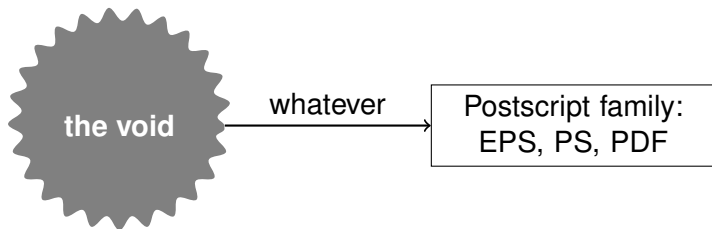
1 Introduction

2 Strategy i

3 Strategy 1

4 Wrap up

Strategy *i*: use something else to create the graphics



- Using `latex` to make a DVI file: only EPS files
- Using `pdflatex` to make PDFs: PDF files, as well as JPG and PNG.

Use “Postscript family” formats whenever possible

- Vector graphics are much better for mathematical/scientific graphics
- The $\text{T}_{\text{E}}\text{X}$ ecosystem is well-adapted to the Postscript family
- Conversion between EPS, PS, PDF is usually quite easy.

Use “Postscript family” formats whenever possible

- Vector graphics are much better for mathematical/scientific graphics
- The $\text{T}_{\text{E}}\text{X}$ ecosystem is well-adapted to the Postscript family
- Conversion between EPS, PS, PDF is usually quite easy.

Use “Postscript family” formats whenever possible

- Vector graphics are much better for mathematical/scientific graphics
- The $\text{T}_{\text{E}}\text{X}$ ecosystem is well-adapted to the Postscript family
- Conversion between EPS, PS, PDF is usually quite easy.

Aside: leave off extension when includegraphics'ing

A tip:

- When using `\includegraphics`, don't add the file extension; `latex` and `pdflatex` automatically look for EPS and PDF files, resp.
- Do `\includegraphics{foo}`, keep `foo.eps` and `foo.pdf` around. (Use `epstopdf` if necessary.)

Aside: leave off extension when includegraphics'ing

A tip:

- When using `\includegraphics`, don't add the file extension; `latex` and `pdflatex` automatically look for EPS and PDF files, resp.
- Do `\includegraphics{foo}`, keep `foo.eps` and `foo.pdf` around. (Use `epstopdf` if necessary.)

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Another aside:

From Wikipedia's article on "Vector graphics": "One of the first uses of vector graphic displays was the [US SAGE air defense system](#)."

SAGE, the Semi-Automatic Ground Environment, was an automated control system for tracking and intercepting enemy bomber aircraft used by NORAD from the late 1950s into the 1980s. [...]

By the time it was fully operational the Soviet bomber threat had been replaced by the Soviet missile threat, for which SAGE was entirely inadequate. Nevertheless, SAGE was tremendously important; it led to huge advances in online systems and interactive computing, real-time computing, and data communications using modems. It is generally considered to be one of the most advanced and successful large computer systems ever developed.

Outline

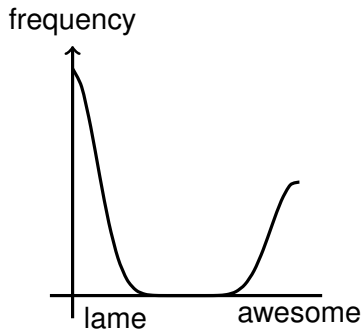
- 1 Introduction
- 2 Strategy i
- 3 Strategy 1**
- 4 Wrap up

Strategy 1: use \LaTeX to create the graphics

This uses packages that use \TeX or \LaTeX to insert graphics directly.

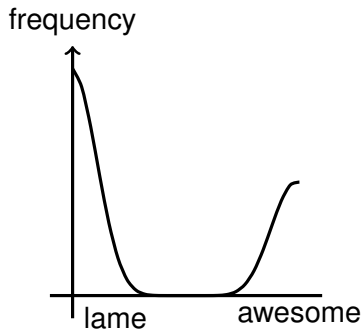
Strategy 1: use \LaTeX to create the graphics

This uses packages that use \TeX or \LaTeX to insert graphics directly.



Strategy 1: use \LaTeX to create the graphics

This uses packages that use \TeX or \LaTeX to insert graphics directly.



Unfortunately, awesomeness is strongly correlated with being difficult to use.

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

Big advantage: T_EX does the text!

Using T_EX means *text in the graphics matches the rest of the article.*

- can do math properly
- matching typeface gives visual and logical consistency

Two slogans from book titles:

- don't click the black T
- write like you give a damn

How to combine strategies 1 and i ?

We need strategy ...

- Overlay a `picture` environment over the graphics (`xfig`, `overpic`, `psfrag`)
- Render text as paths (PiScript, Inkscape, *et al.*)

How to combine strategies 1 and i ?

We need strategy ... $1 + i$!

Use something external to create the graphics, and use $\text{T}_{\text{E}}\text{X}$ to create and place the text.

- Overlay a `picture` environment over the graphics (`xfig`, `overpic`, `psfrag`)
- Render text as paths (PiScript, Inkscape, *et al.*)

How to combine strategies 1 and i ?

We need strategy ... $1 + i$!

Use something external to create the graphics, and use $\text{T}_{\text{E}}\text{X}$ to create and place the text.

Ways to do this:

- Overlay a `picture` environment over the graphics (`xfig`, `overpic`, `psfrag`)
- Render text as paths (PiScript, Inkscape, *et al.*)

How to combine strategies 1 and i ?

We need strategy ... $1 + i$!

Use something external to create the graphics, and use $\text{T}_{\text{E}}\text{X}$ to create and place the text.

Ways to do this:

- Overlay a `picture` environment over the graphics (`xfig`, `overpic`, `psfrag`)
- Render text as paths (PiScript, Inkscape, *et al.*)

How to combine strategies 1 and i ?

We need strategy ... $1 + i$!

Use something external to create the graphics, and use $\text{T}_{\text{E}}\text{X}$ to create and place the text.

Ways to do this:

- Overlay a `picture` environment over the graphics (`xfig`, `overpic`, `psfrag`)
- Render text as paths (PiScript, Inkscape, *et al.*)

Interlude

A disclaimer/cautionary tale/admission of hypocrisy. . .

Outline

- 1 Introduction
- 2 Strategy i
- 3 Strategy 1
- 4 Wrap up**

Colophon

A note about the graphics

What's a colophon? Quoth Wikipedia:

A brief description usually located at the end of a book, describing production notes relevant to the edition. In most cases it is a description of the text typography, often entitled “A note about the type”.

What about the graphics used in these slides?

About the graphics in these slides

- KAIST logo: downloaded an Adobe Illustrator file from KAIST website; used Inkscape to save as PDF.
- Windows and Apple logos: JPGs off the web, cropped in the GIMP.
- Tux: from the SVG file at Wikipedia, used Inkscape to convert to PDF. Tux SVG ©Larry Ewing, Simon Budig, Anja Gerwinski.
- Other graphics: TikZ

Thank you

Tomorrow: gritty details

These slides (source and PDF) will be available from the Sage Days 9 wiki page, and wiki.sagemath.org/DanDrake/Days9Talks