General presentation notes

Preparing higher quality figures

Taking a screenshot of your graph leads to a low resolution image file. These are generally harder to read and look unprofessional.

Fortunately Matplotlib provides a number of ways to create good quality files from your plots. The following code snippet shows an example of a good quality plot.

```
from matplotlib import pyplot as plt
import numpy as np

# generate some points for some trigonometric curves:
x = np.linspace(0, np.pi*2)
sin_points = np.sin(x)
cos_points = np.cos(x)

# Note that the lines are differentiated by lineweight and style.
# Thin coloured lines can be challenging to interpret, especially
# for people with colour vision.
plt.plot(x, cos_points, 'k-', linewidth='2') # Black heavy line
plt.plot(x, sin_points, 'k--', linewidth='1') # Black dashed line.
plt.axis([0, np.pi*2, -1, 1]) # Explicitly set the axis limits

# Add informative labels to the plot.
# Matplotlib can generate some symbols with the appropriate syntax
plt.xlabel('Angle (radians)')
plt.ylabel('Trigonometric function (arbitrary units)')
plt.title('Comparison of sine and cosine functions on [0,$2\pi$]')
# Add a legend.
# The first argument to legend() is a list of strings,
# which match the order the lines are plotted.
plt.legend(('Cosine', 'Sine'), loc='lower left')

# And save it to a high resolution png file.
plt.savefig('trigonometric_examples.png', dpi=300)
```

Including code in your submission

The source code of a program is just a text document - you should present and format it as such when including code in your submission. Taking a screenshot of text is a fundamentally inappropriate way to present this information, and is generally much less readable.

The simplest way to include your code in a printed document is to print it directly from your favourite text editor, and staple it together at the back of your assignment. Add a note somewhere near the beginning and be sure to refer to the appendix.

Including the code in a digital document depends on your word processing environment.

If using Microsoft Word, you can include the text file directly: Under the insert toolbar, click the arrow next to 'object' and select 'Text from File'. Then select the text file (.py in this case) to include. If you click the arrow next to insert on the file selection dialog, you can include the text as a link to the file, allowing you to update your document when the text file changes by pressing F9 on the text field. Similar functionality exists in other word processing packages such as openoffice.

If using Latex, we recommend the listings package.