Intro to Research Computing with Python: Visualization

Erik Spence

SciNet HPC Consortium

21 November 2013

Today's class

Today we will discuss the following topics:

- Basics of visualization and how to get started.
- How to make your work presentable and professional.
- Advanced plotting techniques.
- Animations.

SciNet compute • calcul CANADA

Plotting

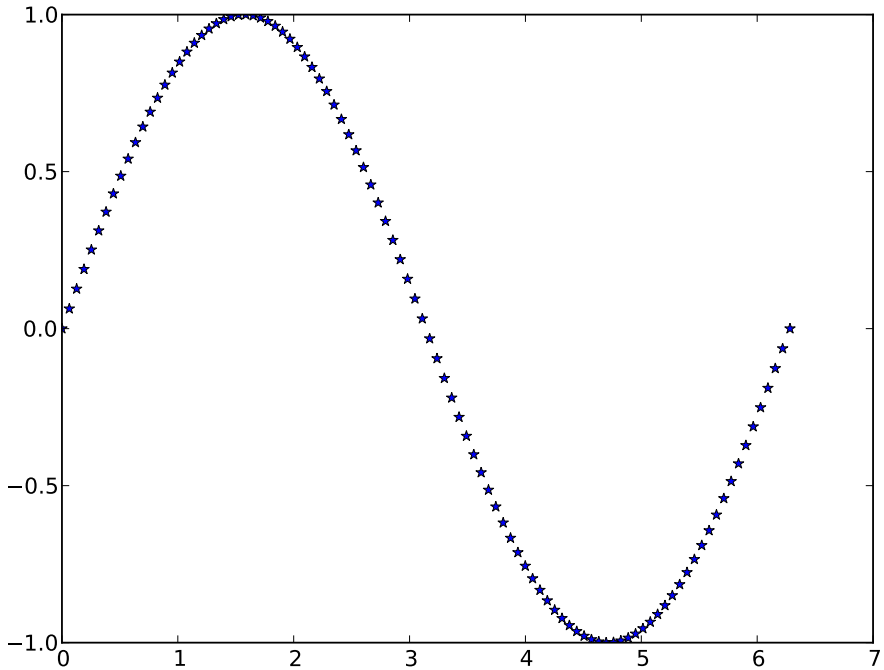
```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]:
```

Plotting

```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]:
```

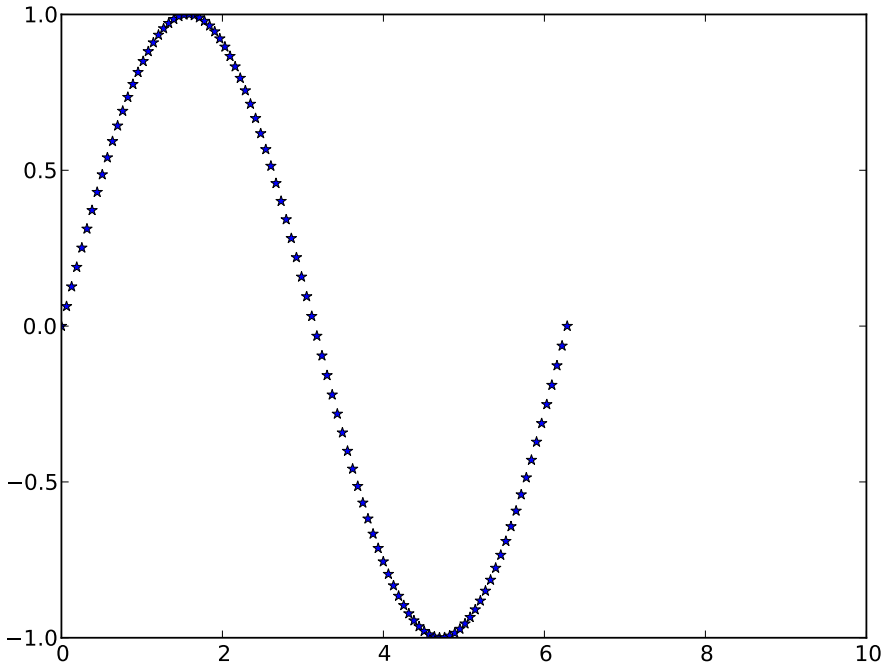
Plotting

```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]:
```



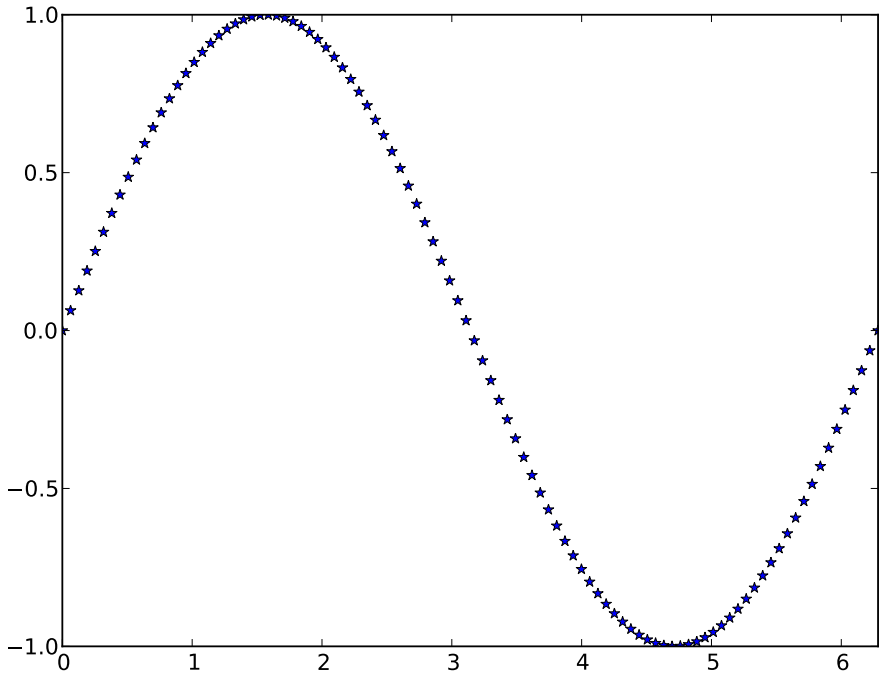
Plotting

```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]:
```



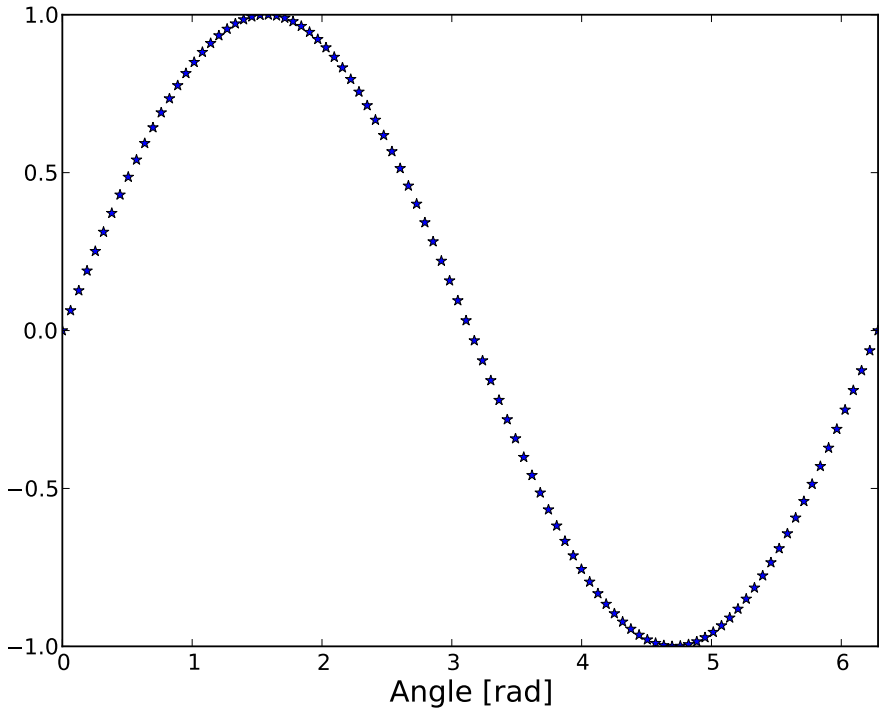
Plotting

```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]: xlim(0, 2 * pi)
Out[6]: (0, 6.2831853071795862)

In [7]:
```



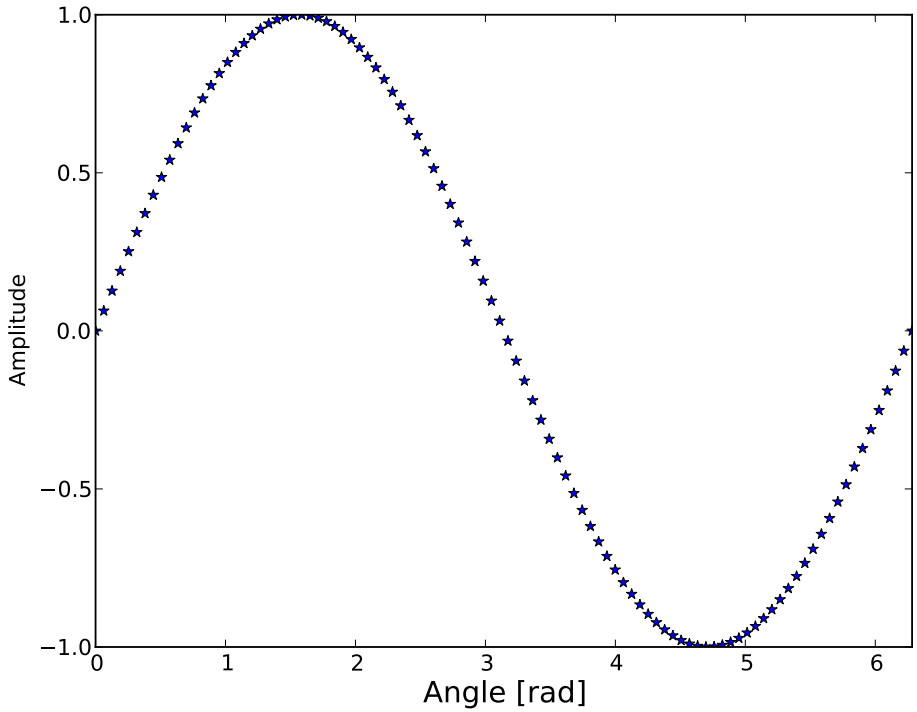
Plotting

```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]: xlim(0, 2 * pi)
Out[6]: (0, 6.2831853071795862)

In [7]: xlabel('Angle [rad]',
  fontsize = 16)

In [8]:
```



Plotting

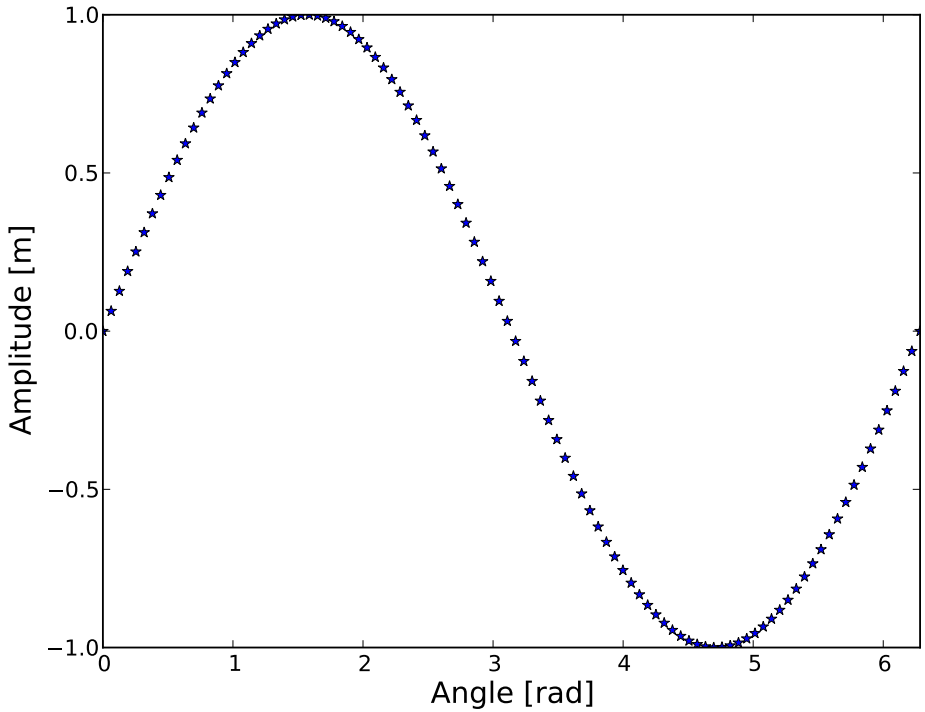
```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]: xlim(0, 2 * pi)
Out[6]: (0, 6.2831853071795862)

In [7]: xlabel('Angle [rad]',
  fontsize = 16)

In [8]: ylabel('Amplitude')
```




```
In [9]:
```

Plotting

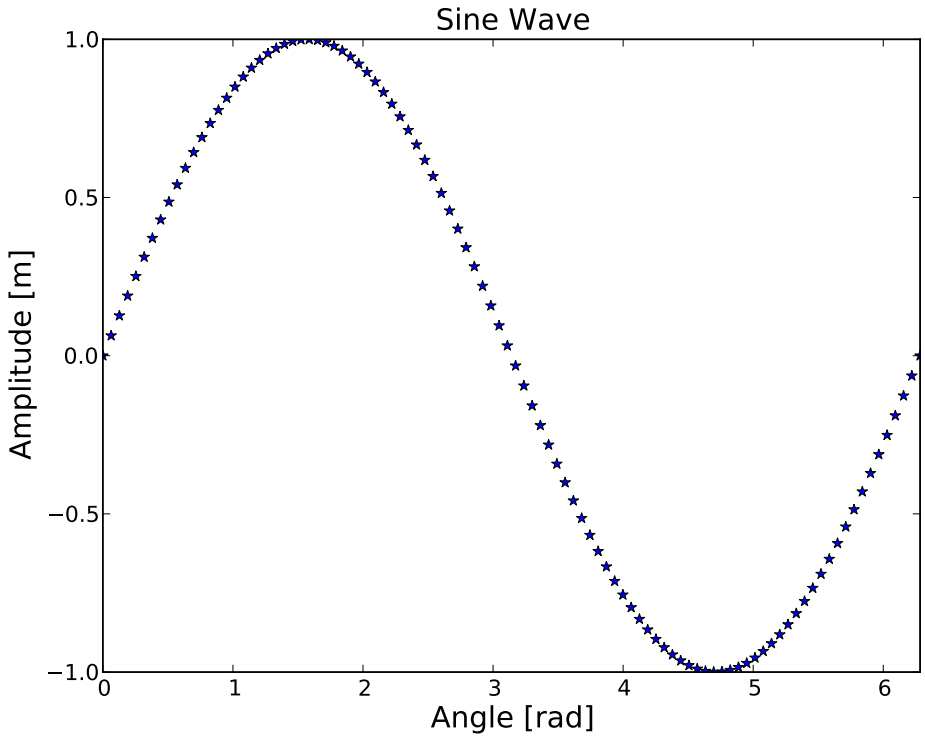
```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]: xlim(0, 2 * pi)
Out[6]: (0, 6.2831853071795862)

In [7]: xlabel('Angle [rad]',
  fontsize = 16)

In [8]: ylabel('Amplitude')
```




```
In [9]: ylabel('Amplitude [m]',
  fontsize = 16)

In [10]:
```

Plotting

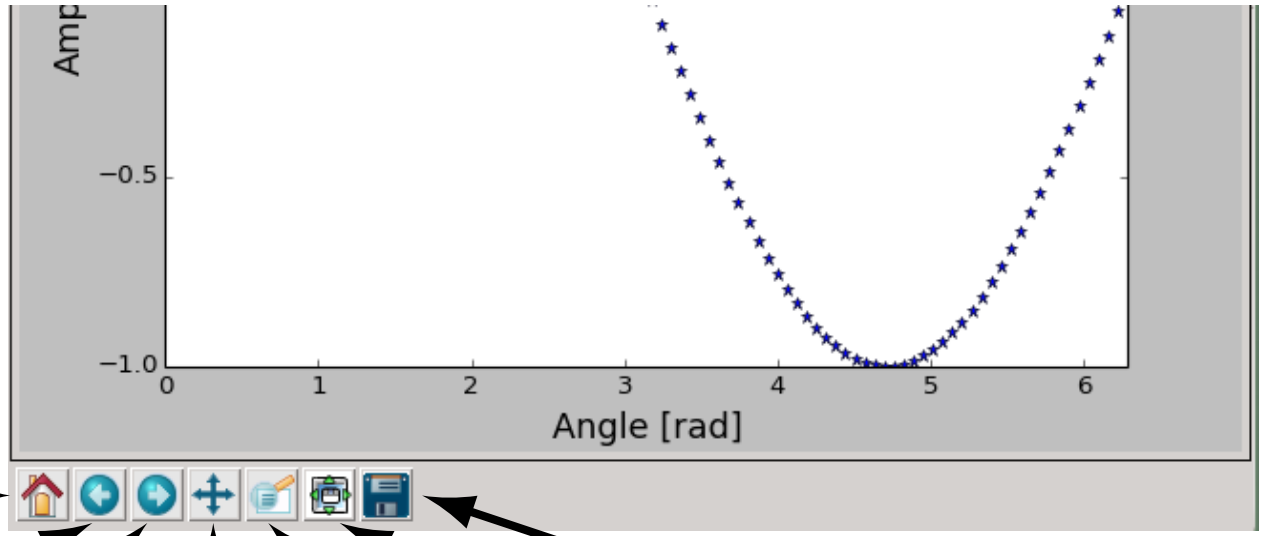
```
In [1]: from numpy import pi

In [2]: from matplotlib import plot,
  xlim, ylim, xlabel, ylabel, title

In [3]: x = linspace(0, 2 * pi, 100)

In [4]: plot(x, sin(x), '*')

In [5]: xlim(0, 10)
Out[5]: (0, 10)

In [6]: xlim(0, 2 * pi)
Out[6]: (0, 6.2831853071795862)

In [7]: xlabel('Angle [rad]',
  fontsize = 16)

In [8]: ylabel('Amplitude')
```




```
In [9]: ylabel('Amplitude [m]',
  fontsize = 16)

In [10]: title('Sine Wave',
  fontsize = 16)
```

SciNet compute • calcul CANADA

What are those buttons?



Return to where you started.

Go back a step.

Go forward a step.

Grab the plot and move it around.

Zoom in on a section.

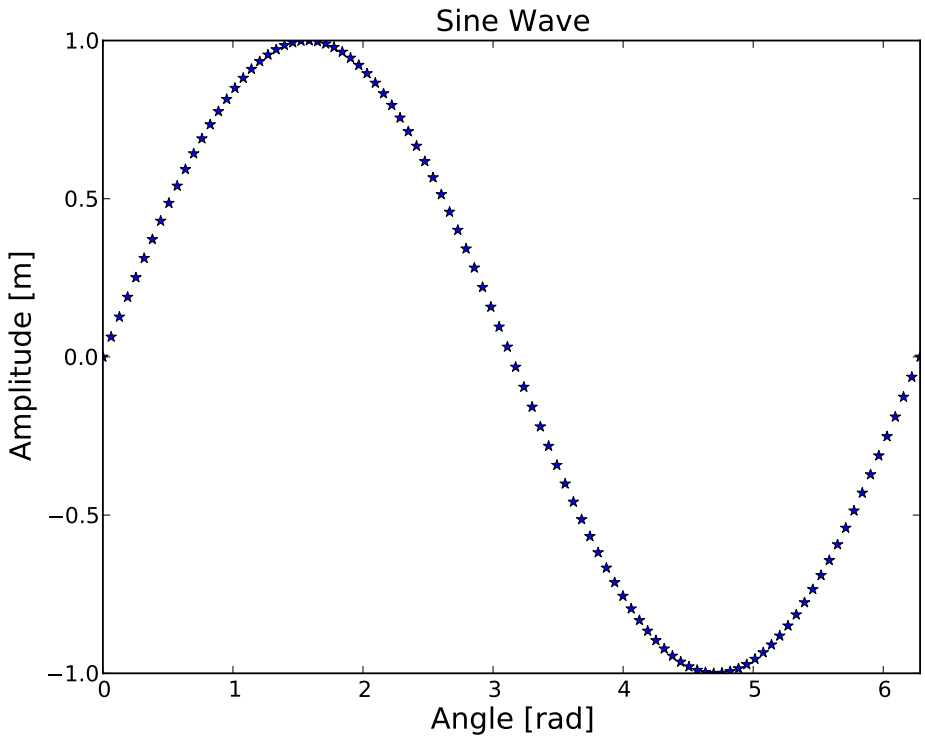
Adjust spacing between plots.

Save the figure.

SciNet compute • calcul CANADA

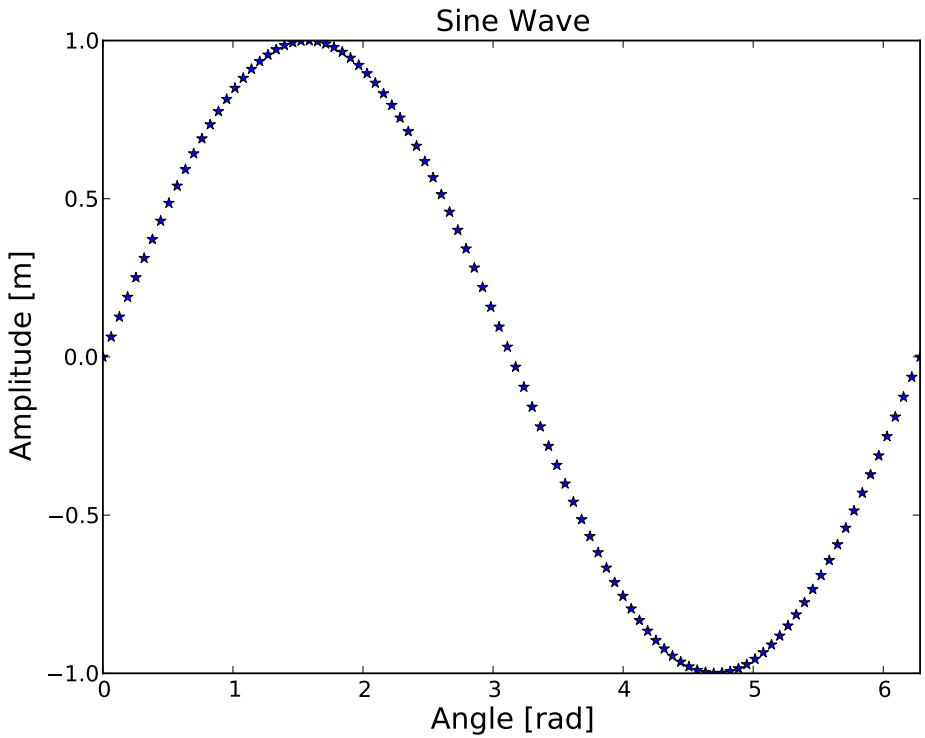
Playing with subplots

```
In [11]:
```



Playing with subplots

```
In [11]: from pylab import subplot

In [12]:
```

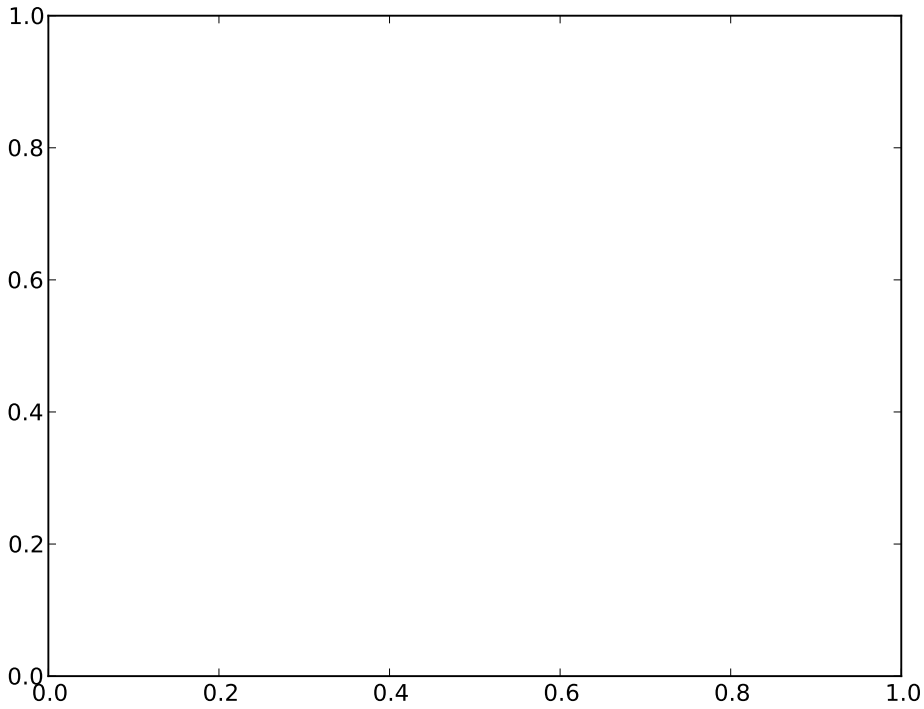


Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]:
```

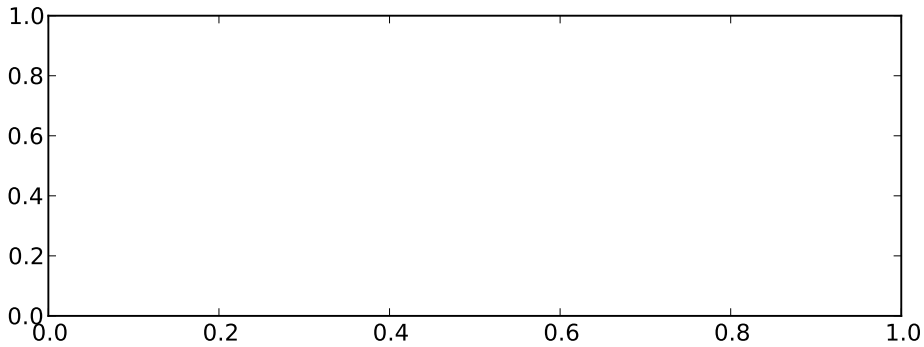
Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]:
```



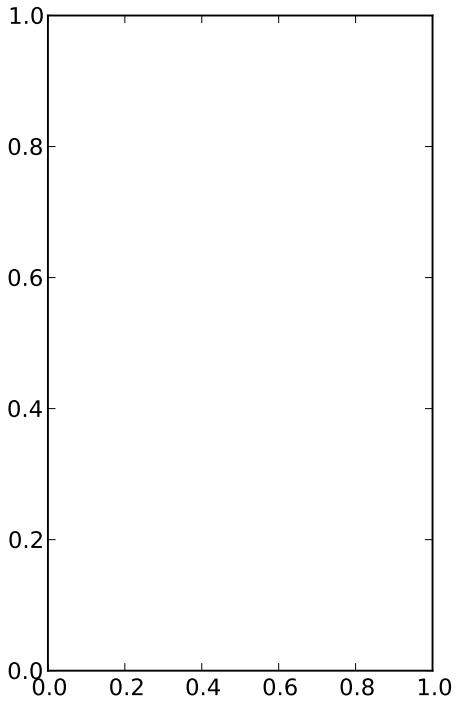
Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]:
```



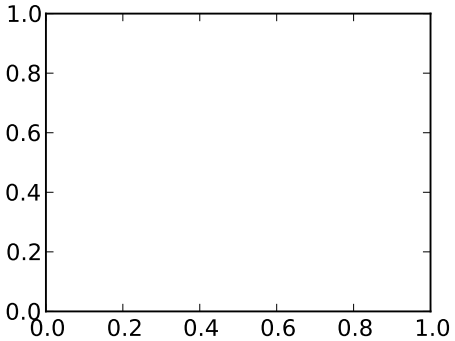
Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]:
```



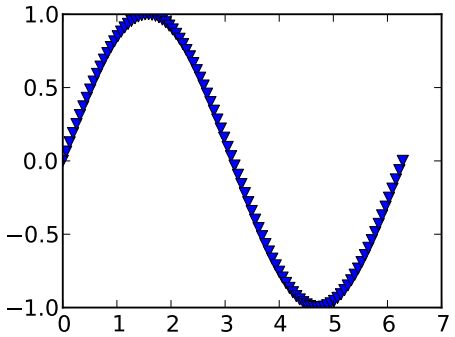
Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]:
```



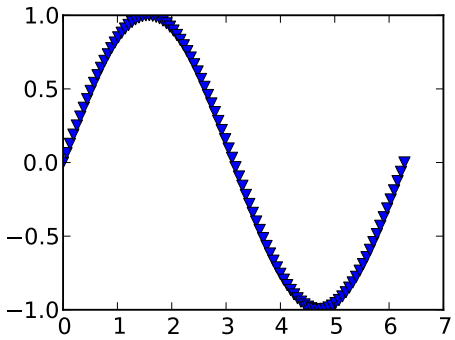
Playing with subplots

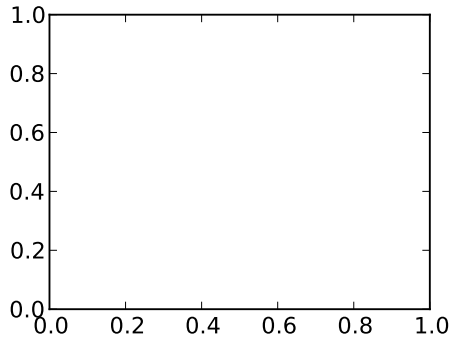
```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]:
```



Playing with subplots

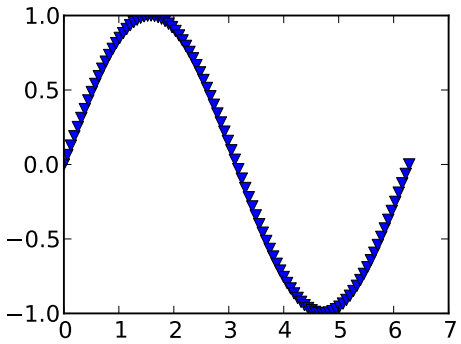
```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]:
```

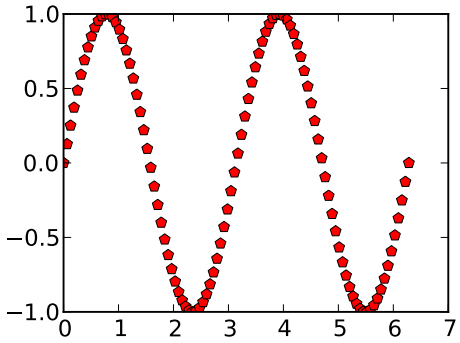




Playing with subplots

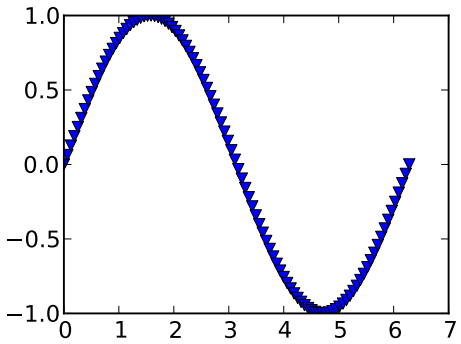
```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]: plot(x, sin(2 * x), 'rp')

In [20]:
```

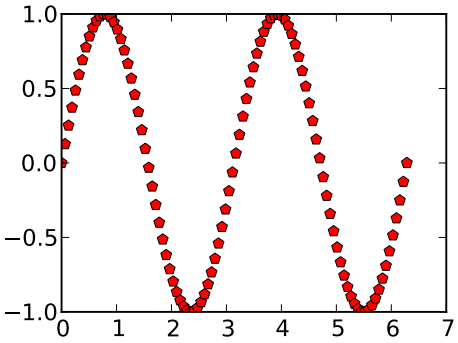




Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]: plot(x, sin(2 * x), 'rp')

In [20]: subplot(2,2,1)
```

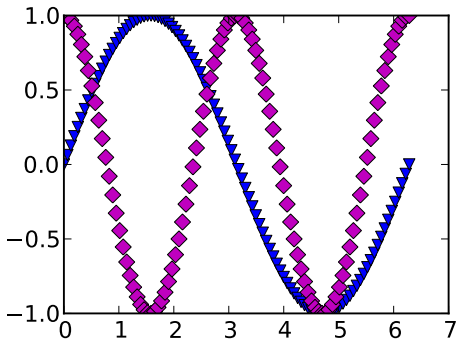


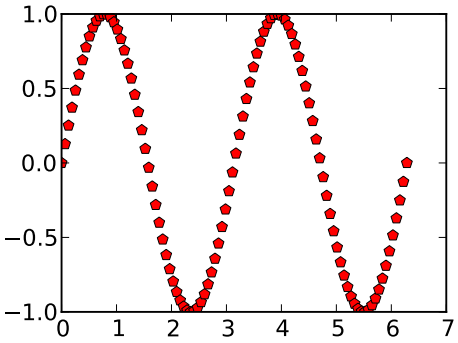



```
In [21]:
```

Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]: plot(x, sin(2 * x), 'rp')

In [20]: subplot(2,2,1)
```



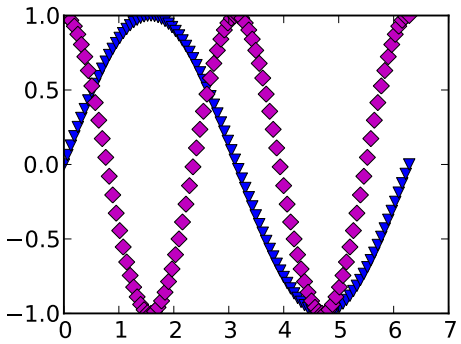


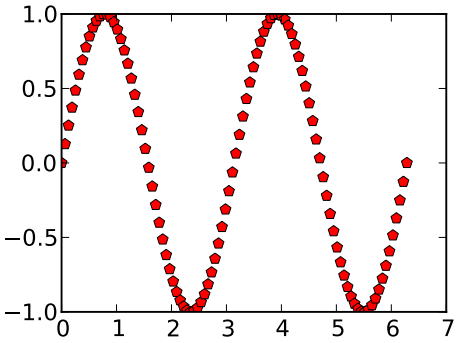

```
In [21]: plot(x, cos(2 * x), 'mD')

In [22]:
```

SciNet compute • calcul CANADA

Playing with subplots

```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]: plot(x, sin(2 * x), 'rp')

In [20]: subplot(2,2,1)
```

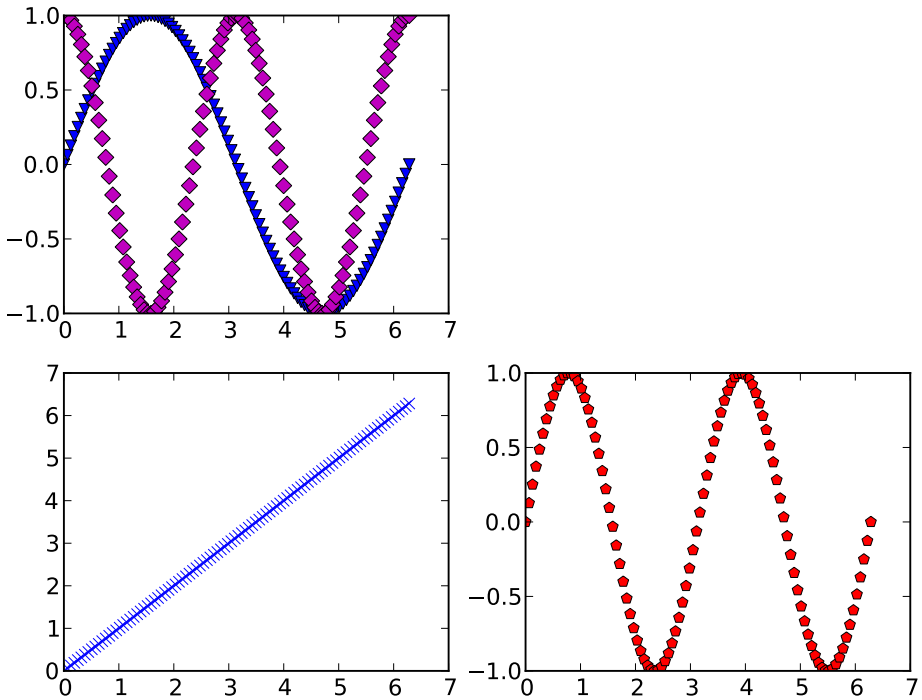





```
In [21]: plot(x, cos(2 * x), 'mD')

In [22]: subplot(2,2,3)

In [23]:
```

Playing with subplots

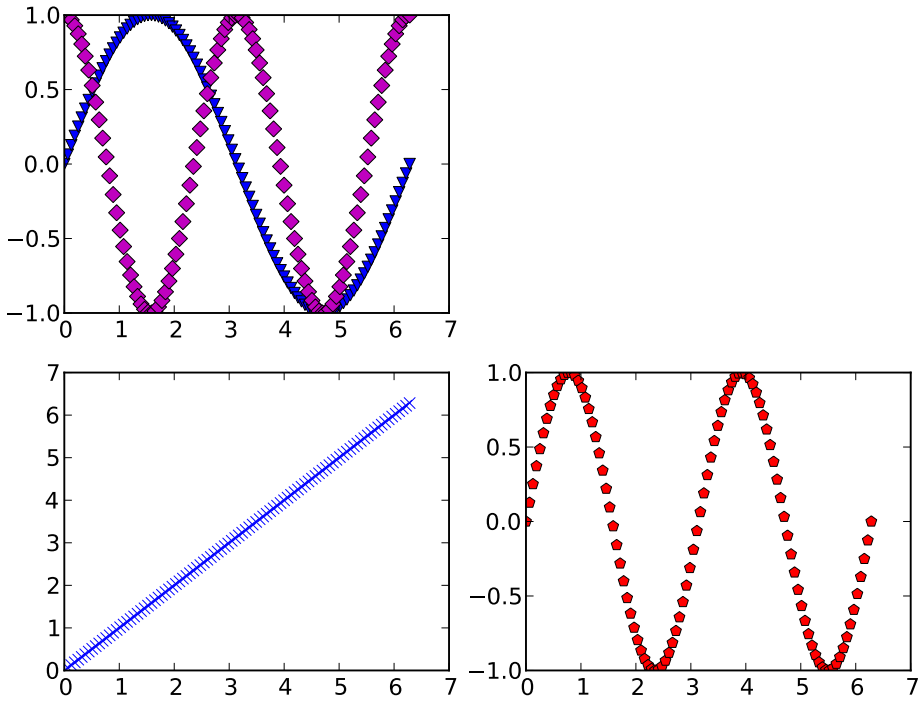
```
In [11]: from pylab import subplot

In [12]: clf()

In [13]: subplot(1,1,1)

In [14]: subplot(2,1,1)

In [15]: subplot(1,2,1)

In [16]: subplot(2,2,1)

In [17]: plot(x, sin(x), 'v')

In [18]: subplot(2,2,4)

In [19]: plot(x, sin(2 * x), 'rp')

In [20]: subplot(2,2,1)
```



```
In [21]: plot(x, cos(2 * x), 'mD')

In [22]: subplot(2,2,3)

In [23]: plot(x, x, 'x')
```

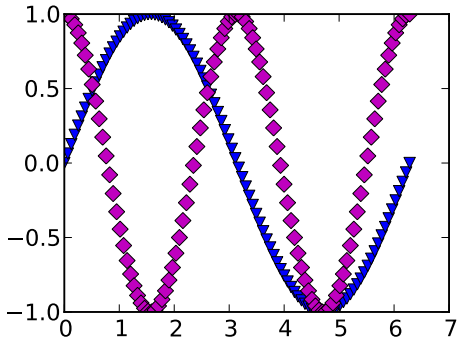
Using error bars

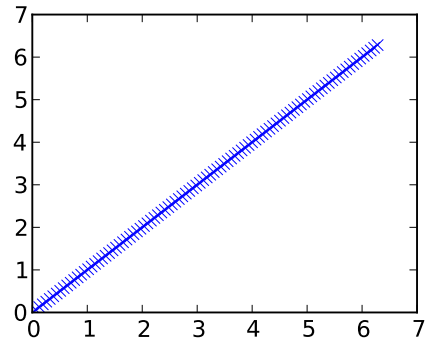
```
In [24]:
```

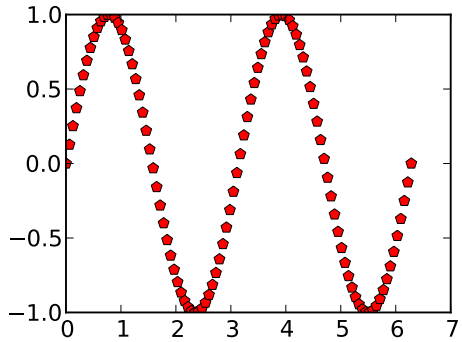


Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]:
```

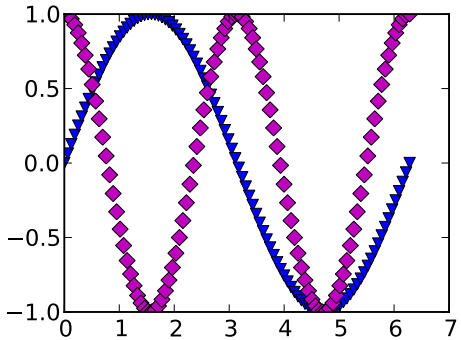


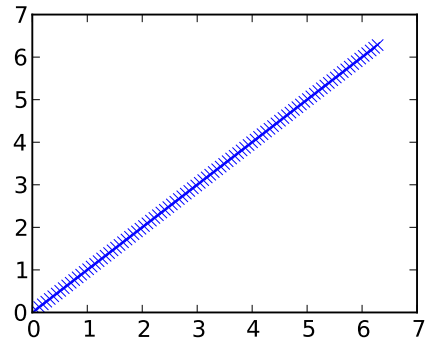


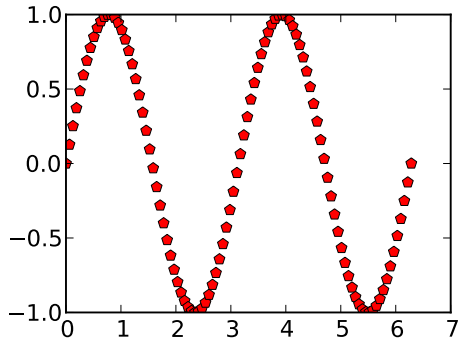


Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]:
```







Using error bars

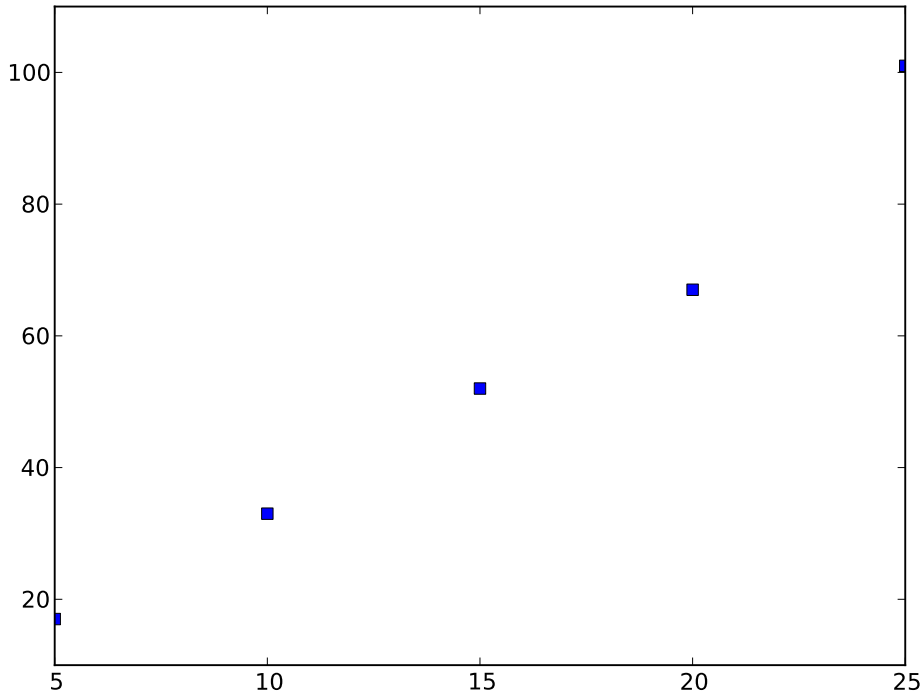
```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]:
```

Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]:
```

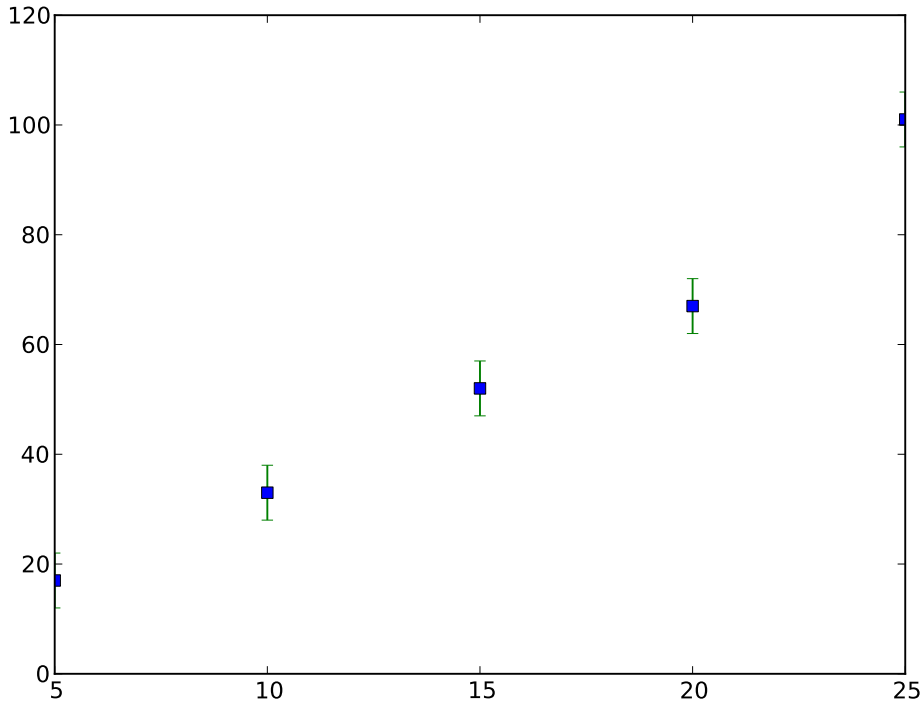
Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]:
```



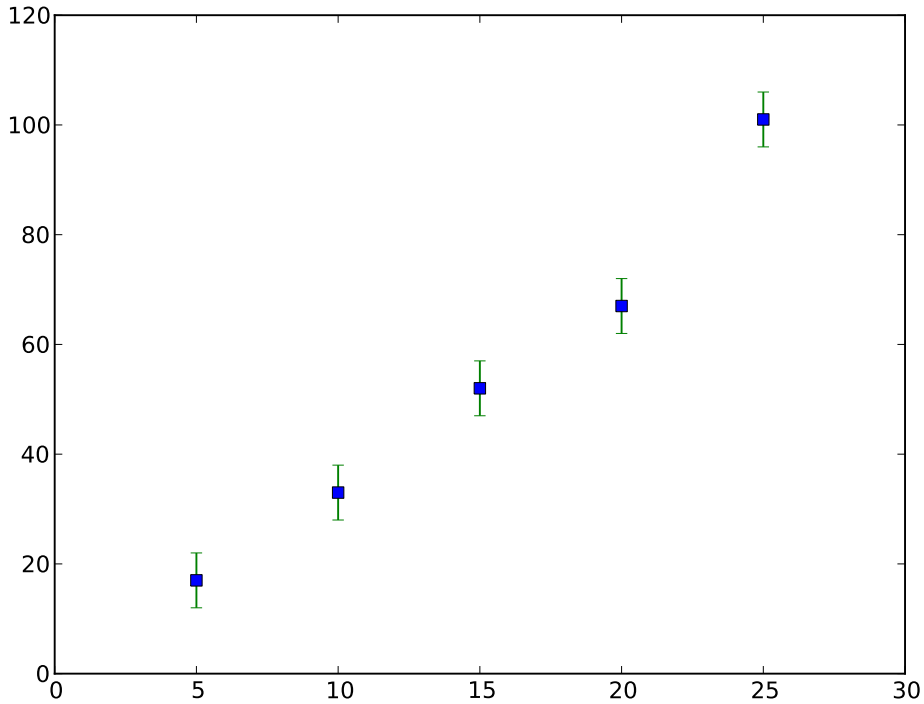
Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]:
```



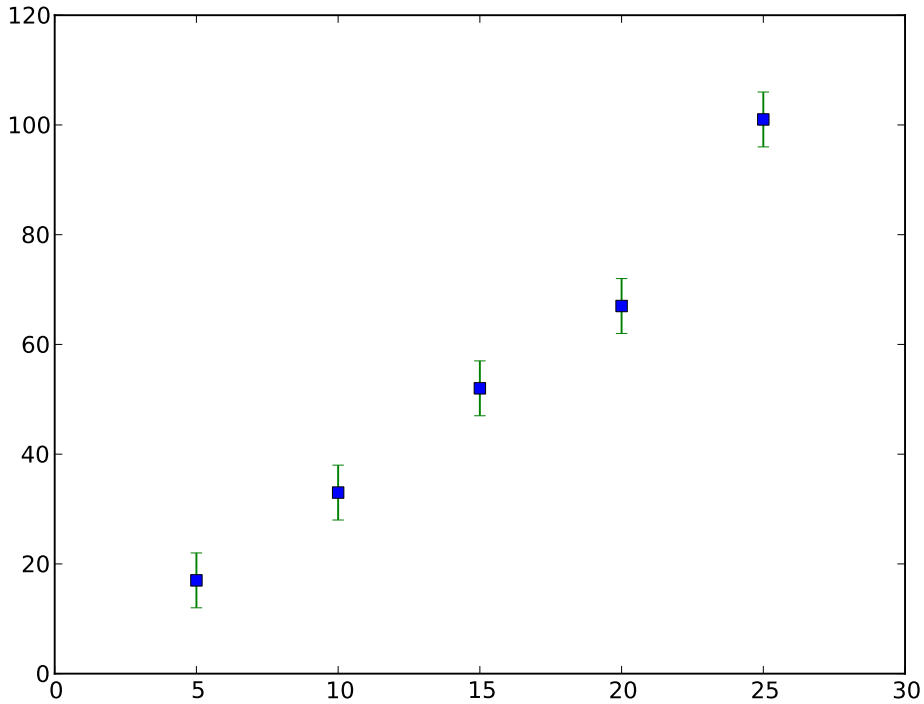
Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]: xlim(0, 30)

In [32]:
```



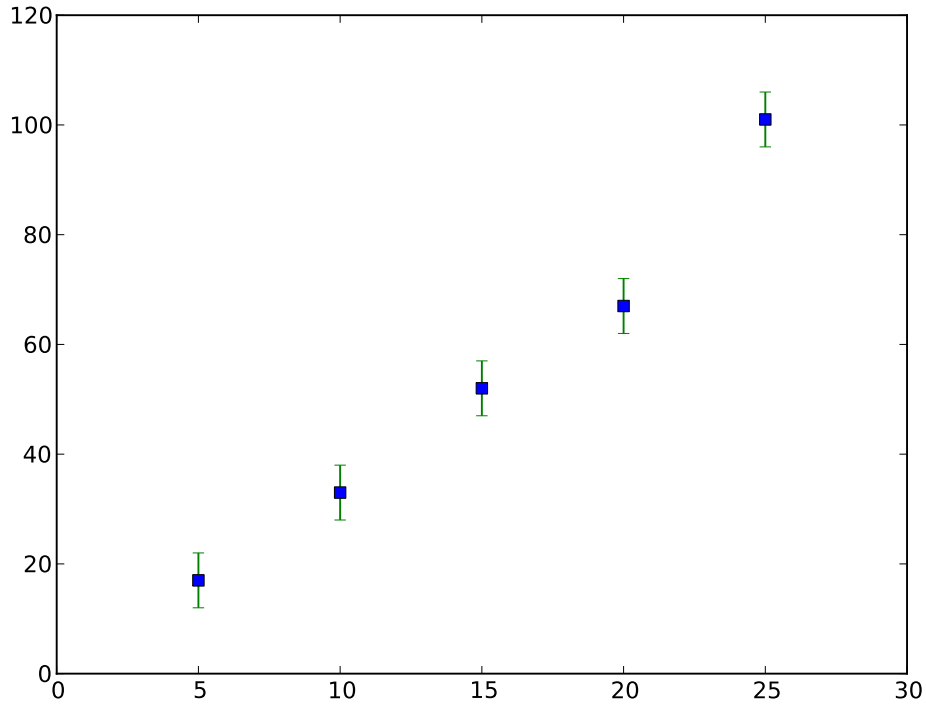
Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]: xlim(0, 30)

In [32]: fit = polyfit(o2, o1, 1)

In [33]:
```



Using error bars

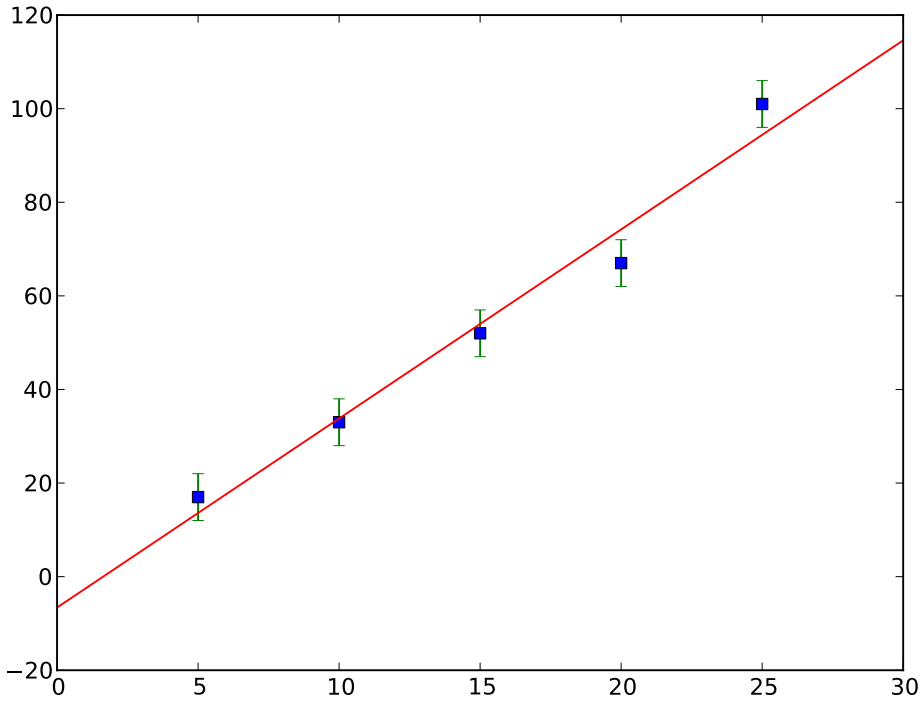
```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]: xlim(0, 30)

In [32]: fit = polyfit(o2, o1, 1)

In [33]: x = linspace(0, 30, 100)
```




```
In [34]:
```

Using error bars

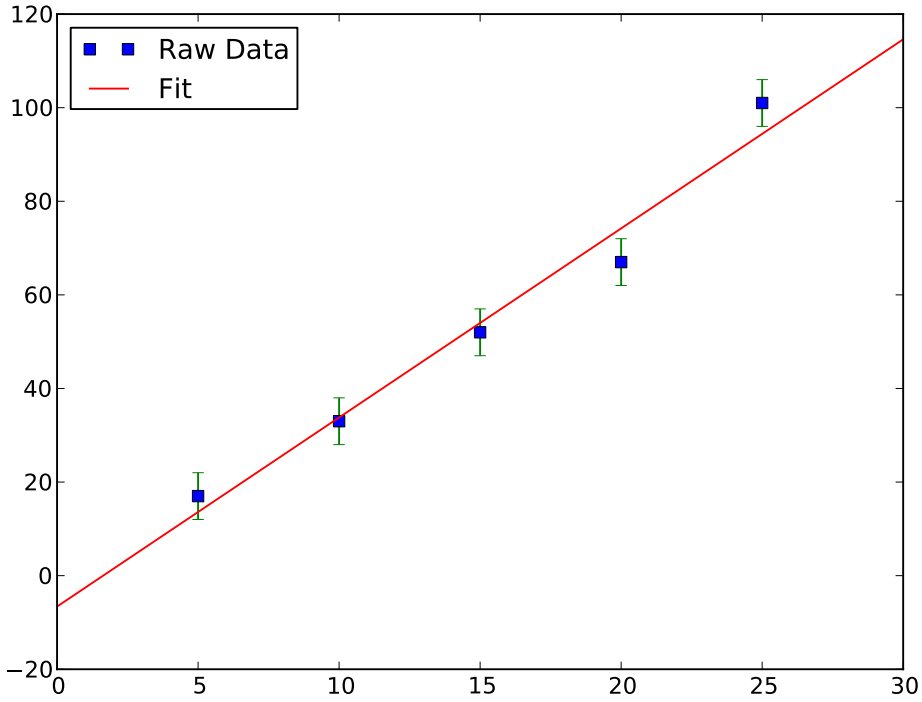
```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]: xlim(0, 30)

In [32]: fit = polyfit(o2, o1, 1)

In [33]: x = linspace(0, 30, 100)
```




```
In [34]: plot(x, polyval(fit,x),
  label='Fit')

In [35]:
```

Using error bars

```
In [24]: from pylab import errorbar,
  legend
In [25]: from matplotlib import label

In [26]: clf()

In [27]: o2 = arange(5, 26, 5)
In [28]: o1 = [17, 33, 52, 67, 101]

In [29]: plot(o2, o1, 's',
  label = 'Raw Data')

In [30]: errorbar(o2, o1, yerr = 5,
  fmt = None)

In [31]: xlim(0, 30)

In [32]: fit = polyfit(o2, o1, 1)

In [33]: x = linspace(0, 30, 100)
```

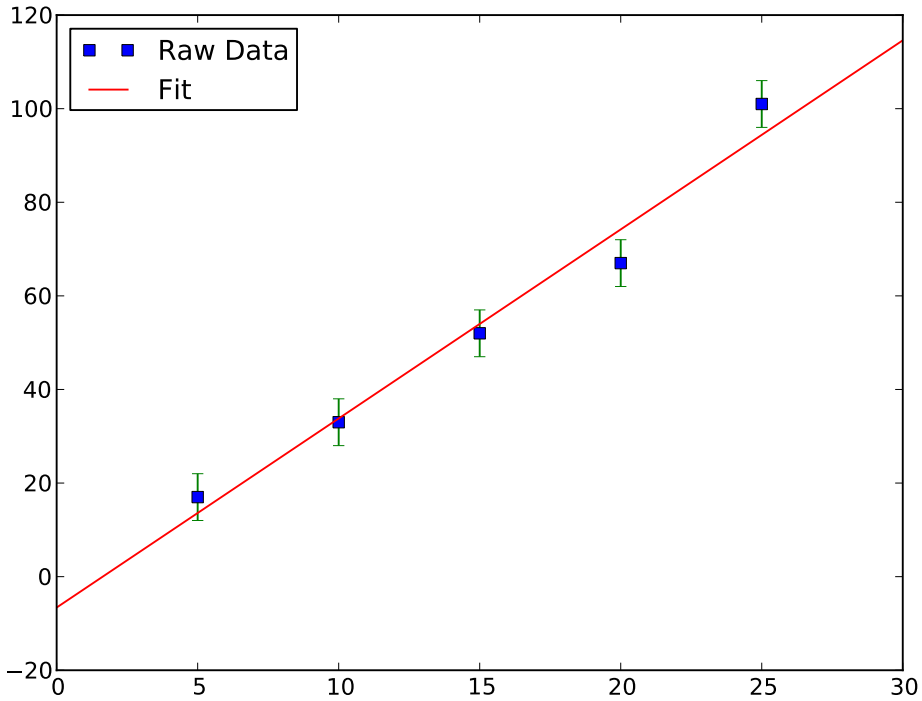



```
In [34]: plot(x, polyval(fit,x),
  label='Fit')

In [35]: legend(loc = 'best')
```

SciNet compute • calcul CANADA

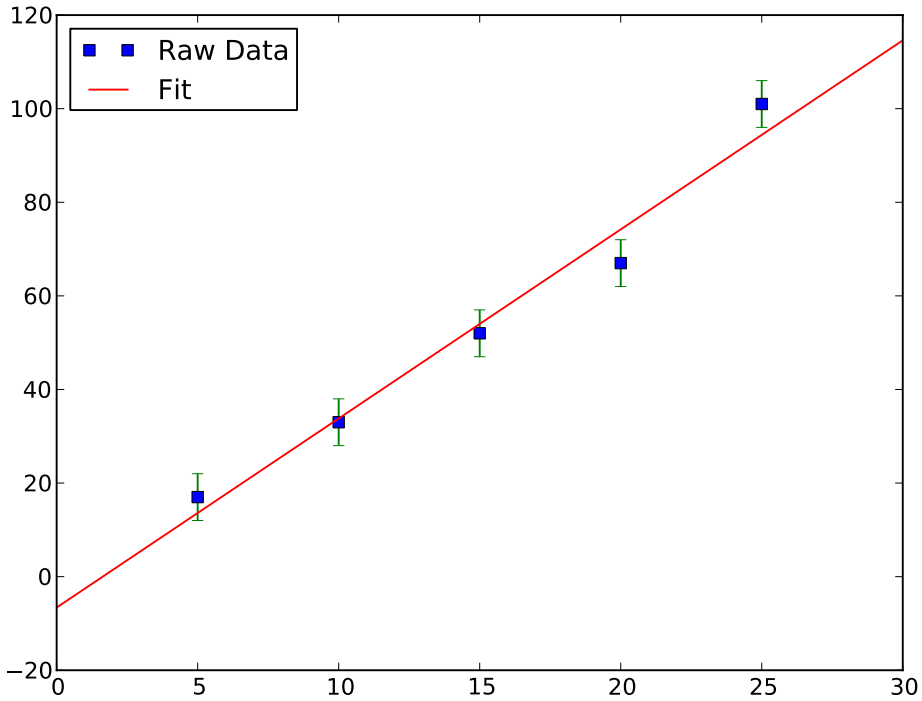
2D plotting

```
In [36]:
```



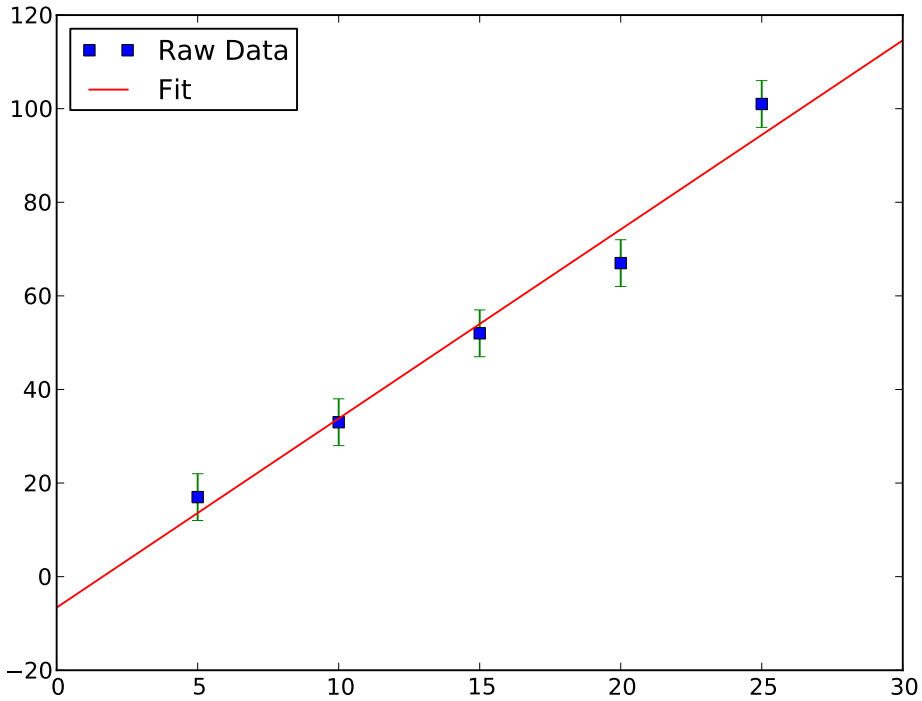
2D plotting

```
In [36]: from numpy import mgrid

In [37]:
```



2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]:
```



2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]:
```

2D plotting

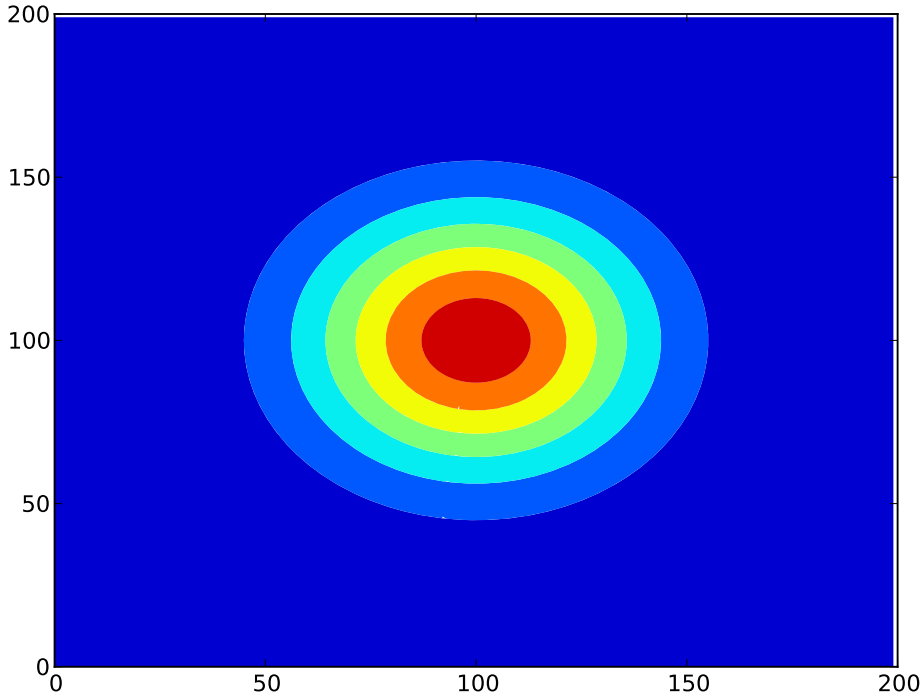
```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]:
```

2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]:
```

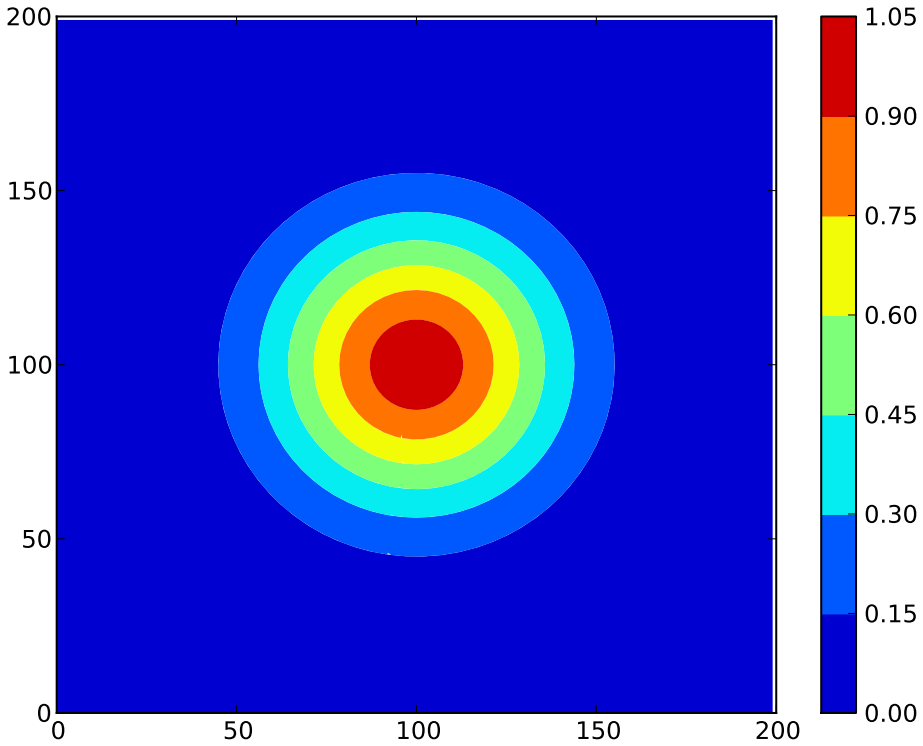
2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]:
```



2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]: colorbar()

In [43]:
```



2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]: colorbar()

In [43]: clf()

In [44]:
```

2D plotting

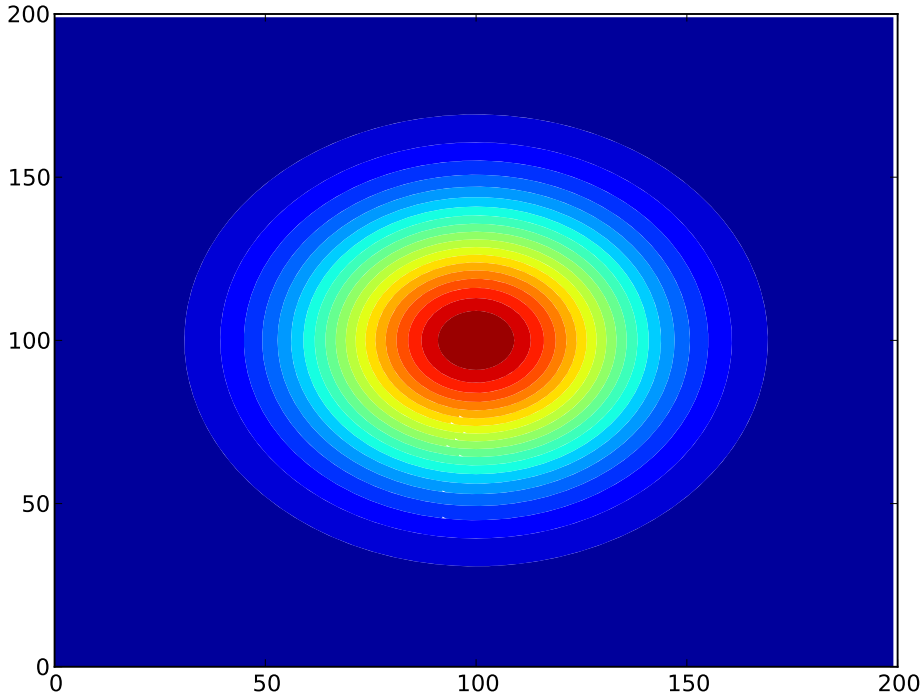
```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]: colorbar()

In [43]: clf()

In [44]: V = linspace(g.min(),
  g.max(), 21)
```

```
In [45]:
```

SciNet compute • calcul CANADA

2D plotting

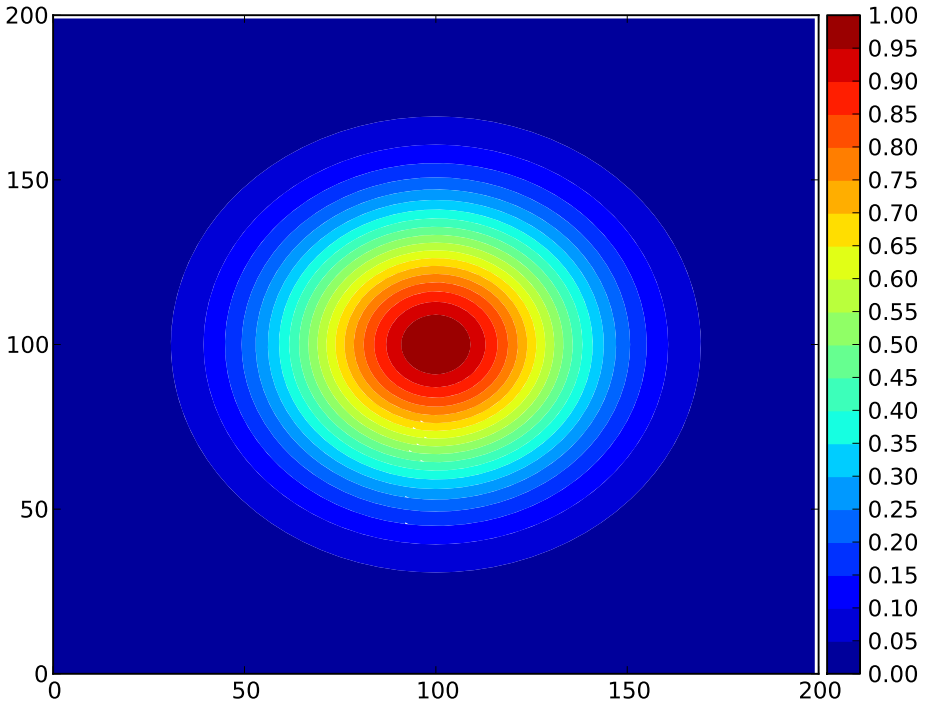
```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]: colorbar()

In [43]: clf()

In [44]: V = linspace(g.min(),
  g.max(), 21)
```




```
In [45]: contourf(g, V)

In [46]:
```

2D plotting

```
In [36]: from numpy import mgrid

In [37]: from pylab import contourf,
  colorbar

In [38]: clf()

In [39]: x, y = mgrid[-10:10:0.1,
  -10:10:0.1]

In [40]: g = exp(-(x**2 + y**2)/16)

In [41]: contourf(g)

In [42]: colorbar()

In [43]: clf()

In [44]: V = linspace(g.min(),
  g.max(), 21)
```




```
In [45]: contourf(g, V)

In [46]: colorbar(format = '%.2f',
  pad = 0.01, fraction = 0.09,
  ticks = V)
```

SciNet compute • calcul CANADA

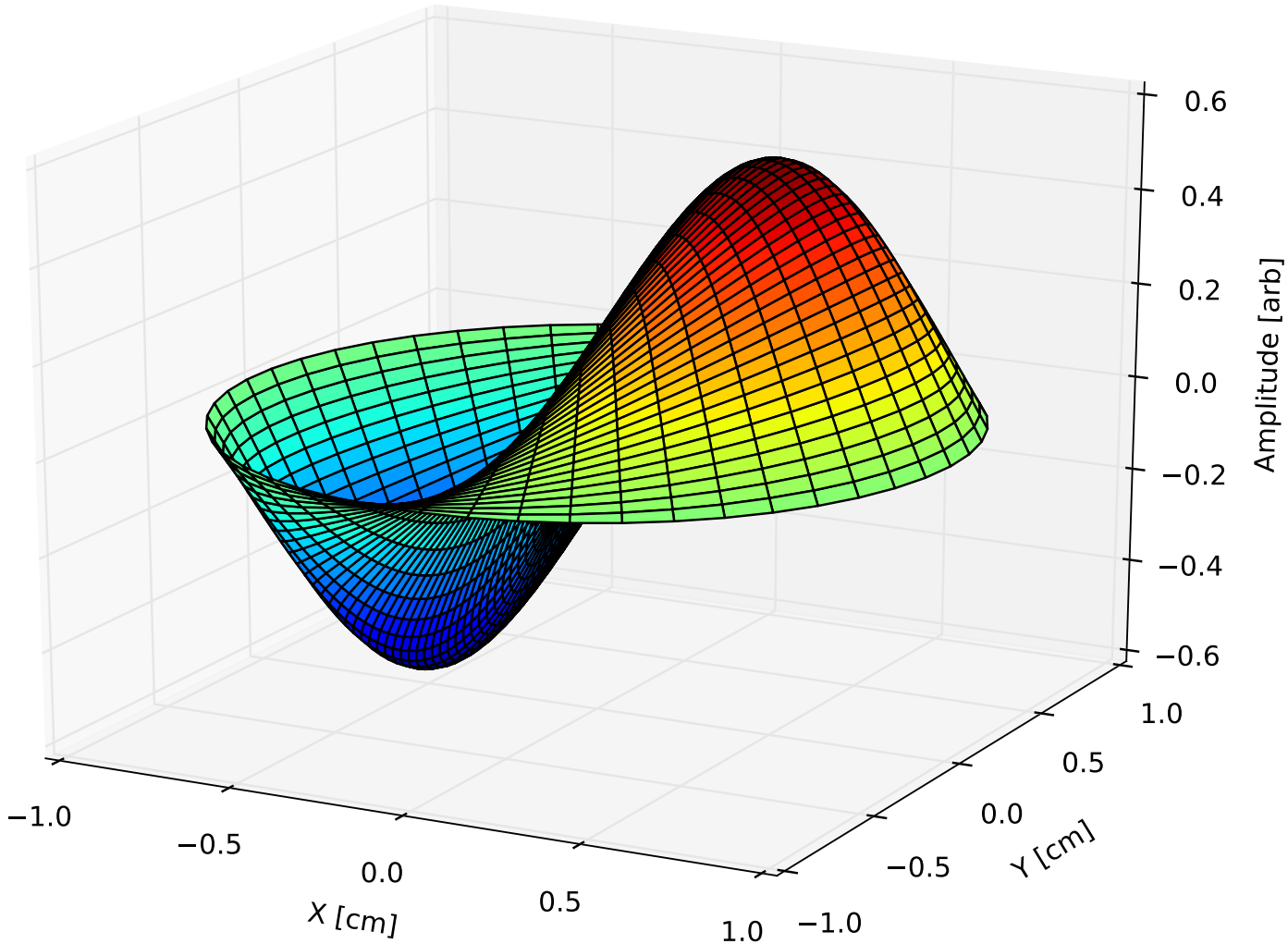
3D plotting

Plotting in 3 dimensions is a little more complicated than in 2.

```
# drum_plot.py
from numpy import *
import pylab as p
from scipy.special import jn,
  jn_zeros
from mpl_toolkits.mplot3d import
  Axes3D

# Define the drumhead function.
def drumhead_height(r, theta):
  nth_zero = jn_zeros(1, 1)
  return cos(theta) *
    jn(1, r * nth_zero)

# Coordinates on the drum head.
theta = linspace(0, 2 * pi, 50)
radius = linspace(0, 1, 50)
```

```
x = zeros([50,50]); y = zeros([50,50])
z = zeros([50,50]); i = 0
for r in radius:
  x[i,:] = r * cos(theta)
  y[i,:] = r * sin(theta)
  z[i,:] = drumhead_height(r, theta)
  i += 1

# Create figure; invoke 3D axis.
fig = p.figure()
ax = Axes3D(fig)

ax.plot_surface(x, y, z, rstride = 1,
  cstride = 1, cmap = p.cm.jet)
ax.set_xlabel('X [cm]')
ax.set_ylabel('Y [cm]')
ax.set_zlabel('Amplitude [arb]')

# Show the plot.
p.show()
```

3D plot


0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Amplitude [arb]
−1.0
−0.5
0.0
0.5
1.0
X [cm]
1.0
0.5
0.0
−0.5
−1.0
Y [cm]

Plotting for professional scientists

Scientists must make their work presentable. That means making a serious effort to make your results easy for your audience to understand.

Suggestions:

- DO label *everything*: axes, lines, fits, data.
- DO put units on your axis labels.
- DO use legends, where appropriate.
- DO adjust the font size of axis and tick labels so that they can actually be read.
- DO NOT put a title label over your plot (for talks and papers).
- DO NOT use colours that cannot be read on a white background (yellow, orange, light green). This is especially important for figures used in talks.
- DO make your data fill the plot.

SciNet compute • calcul CANADA

More plotting for professional scientists

It's also important to consider file types and sizes.

More suggestions:

- DO set the image size and resolution to that requested by the journal you're submitting to.
- DO NOT use bitmap or stroke fonts for your plot. These cannot be rescaled properly, which is often needed for publication. Use vector fonts (the default for Python).
- If possible, DO NOT use image file types that cannot be scaled (bitmap, jpeg). Use EPS or PDF.
- DO NOT leave a bunch of white space around the outside of your plots.
- DO make a script (in Python or whatever language), whose sole purpose is to make that plot for your paper.

SciNet compute • calcul CANADA

Advanced plotting

In [47]:

Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]:
```

Advanced plotting

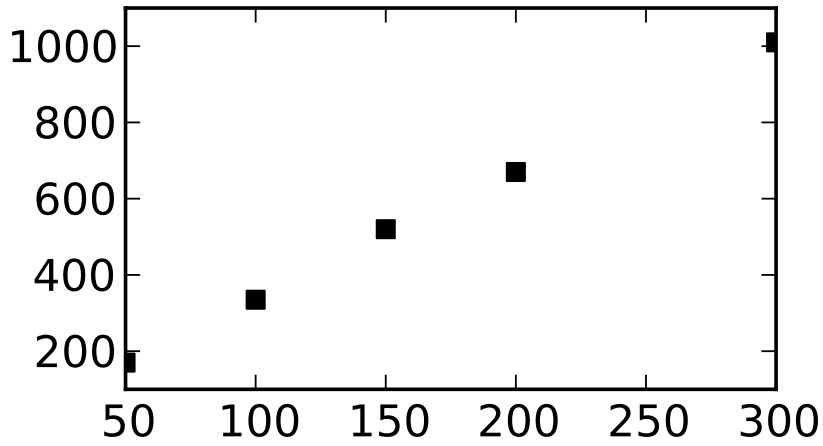
```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]:
```

Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]:
```

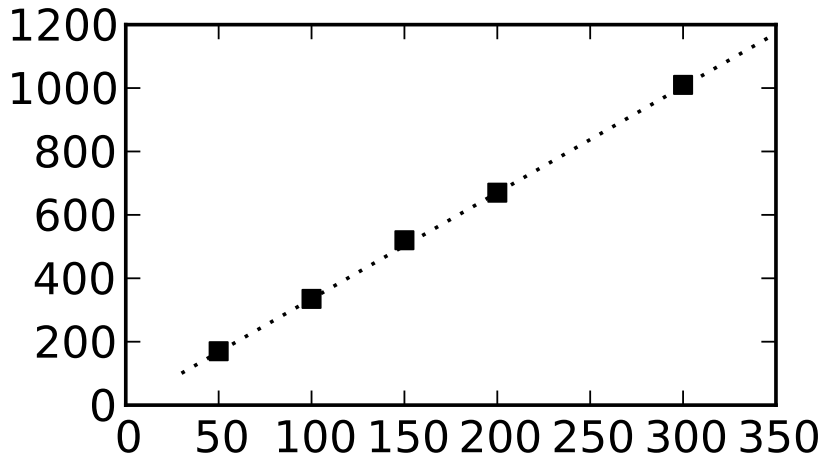
Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]:
```



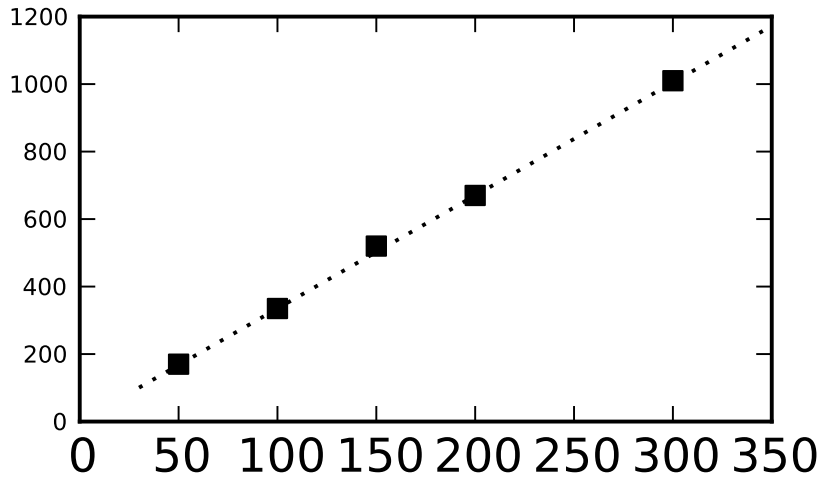
Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]:
```



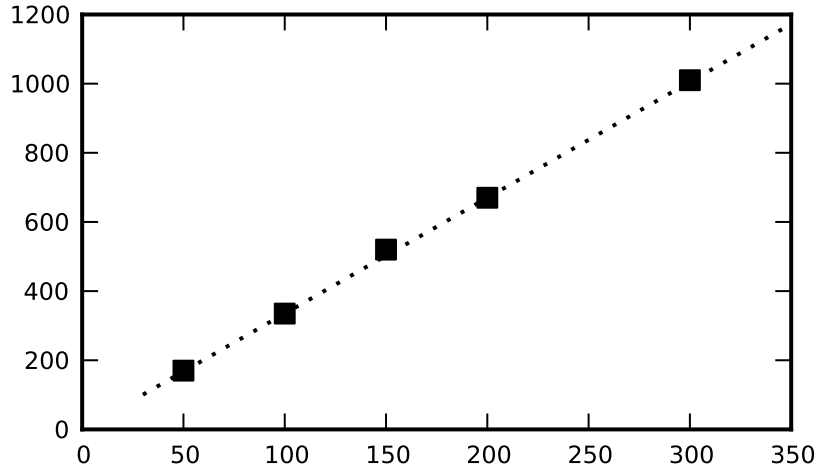
Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [55]:
```



Advanced plotting

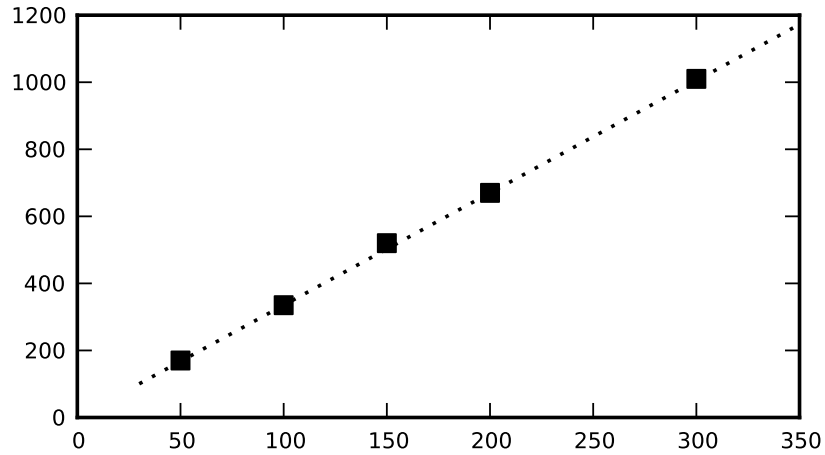
```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [55]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```




```
In [56]:
```

Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [55]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```

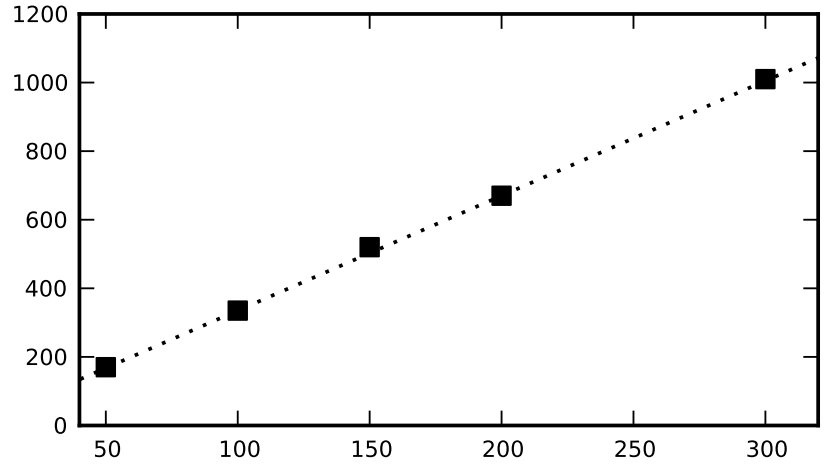



```
In [56]: fig.subplots_adjust(
  top = 0.97, right = 0.98,
  left = 0.115, bottom = 0.12)

In [57]:
```

compute • calcul CANADA

Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [55]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```

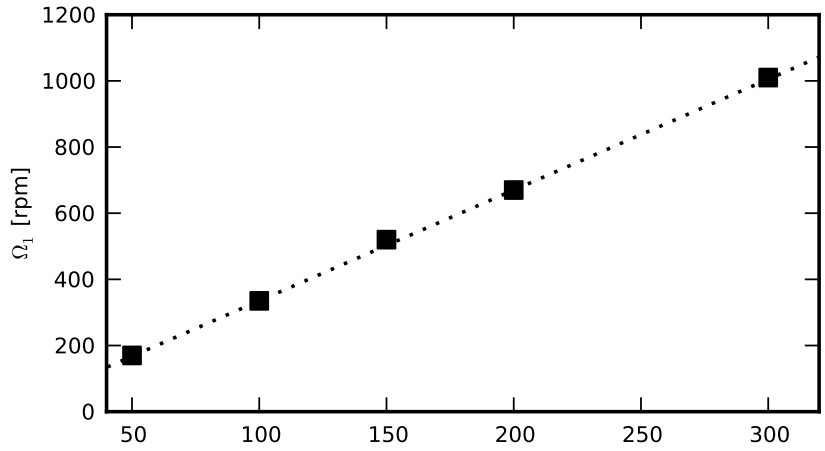



```
In [56]: fig.subplots_adjust(
  top = 0.97, right = 0.98,
  left = 0.115, bottom = 0.12)

In [57]: xlim(40, 320)
Out[57]: (40, 320)

In [58]:
```

SciNet compute • calcul CANADA

Advanced plotting

```
In [47]: o2 = [50, 100, 150, 200, 300]
In [48]: o1 = [170, 335, 520, 670, 1010]
In [49]: o = linspace(30, 350, 100)

In [50]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [51]: a = fig.add_subplot(1,1,1)

In [52]: a.plot(o2,o1,'ks',markersize=5)
In [53]: a.plot(o, 3.35*o, ':',
  color='k')

In [54]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [55]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```

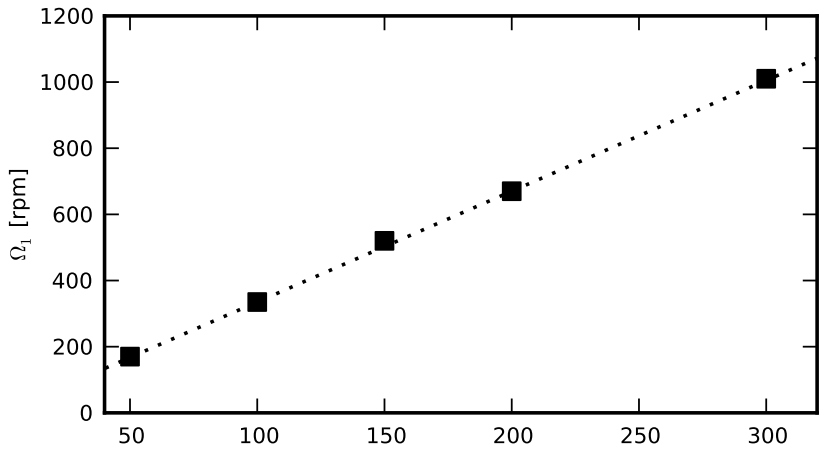



```
In [56]: fig.subplots_adjust(
  top = 0.97, right = 0.98,
  left = 0.115, bottom = 0.12)

In [57]: xlim(40, 320)
Out[57]: (40, 320)

In [58]: ylabel(r'$\Omega_1$ [rpm]',
  fontsize = 6,
  horizontalalignment='center')
```

SciNet compute • calcul CANADA

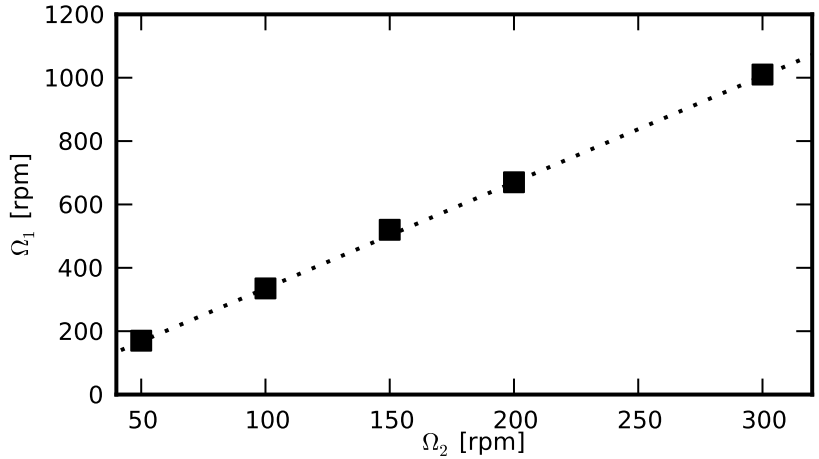
Advanced plotting, continued

```
In [59]:
```



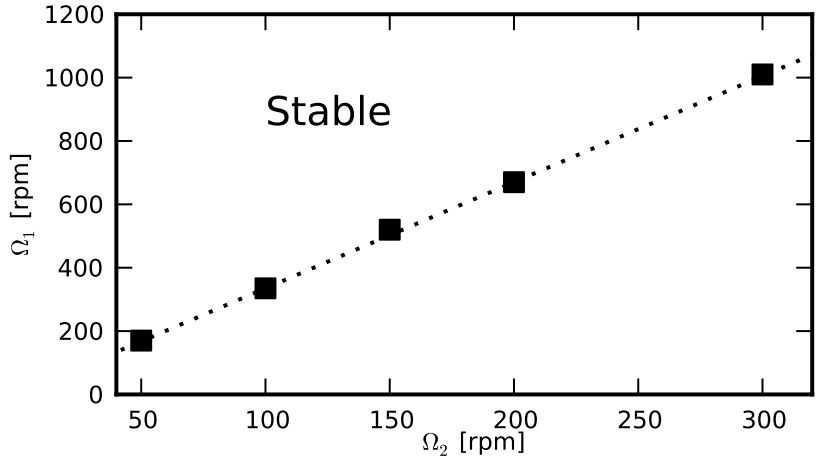
Advanced plotting, continued

```
In [59]: text(163, -180.,
  r'$\Omega_2$ [rpm]', fontsize = 6)

In [60]:
```



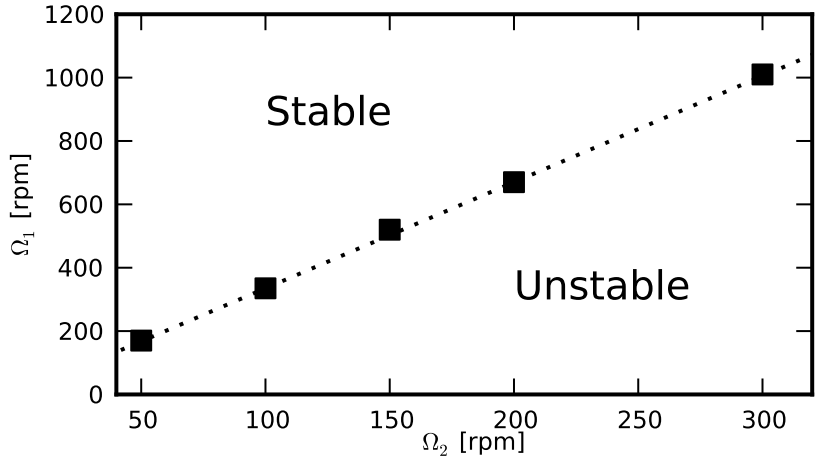
Advanced plotting, continued

```
In [59]: text(163, -180.,
  r'$\Omega_2$ [rpm]', fontsize = 6)

In [60]: text(100, 850., 'Stable',
  fontsize = 10)

In [61]:
```



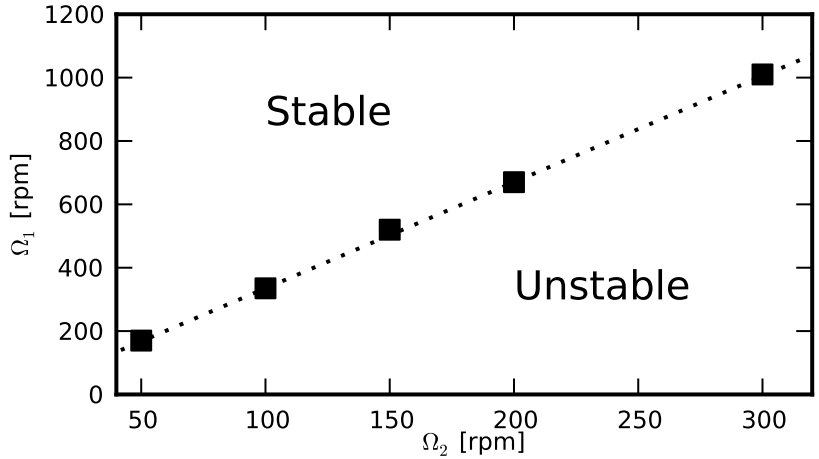
Advanced plotting, continued

```
In [59]: text(163, -180.,
  r'$\Omega_2$ [rpm]', fontsize = 6)

In [60]: text(100, 850., 'Stable',
  fontsize = 10)

In [61]: text(200, 300., 'Unstable',
  fontsize = 10)

In [62]:
```



Advanced plotting, continued

```
In [59]: text(163, -180.,
  r'$\Omega_2$ [rpm]', fontsize = 6)

In [60]: text(100, 850., 'Stable',
  fontsize = 10)

In [61]: text(200, 300., 'Unstable',
  fontsize = 10)

In [62]: savefig('critical_rossby.eps',
  dpi = 600)
```

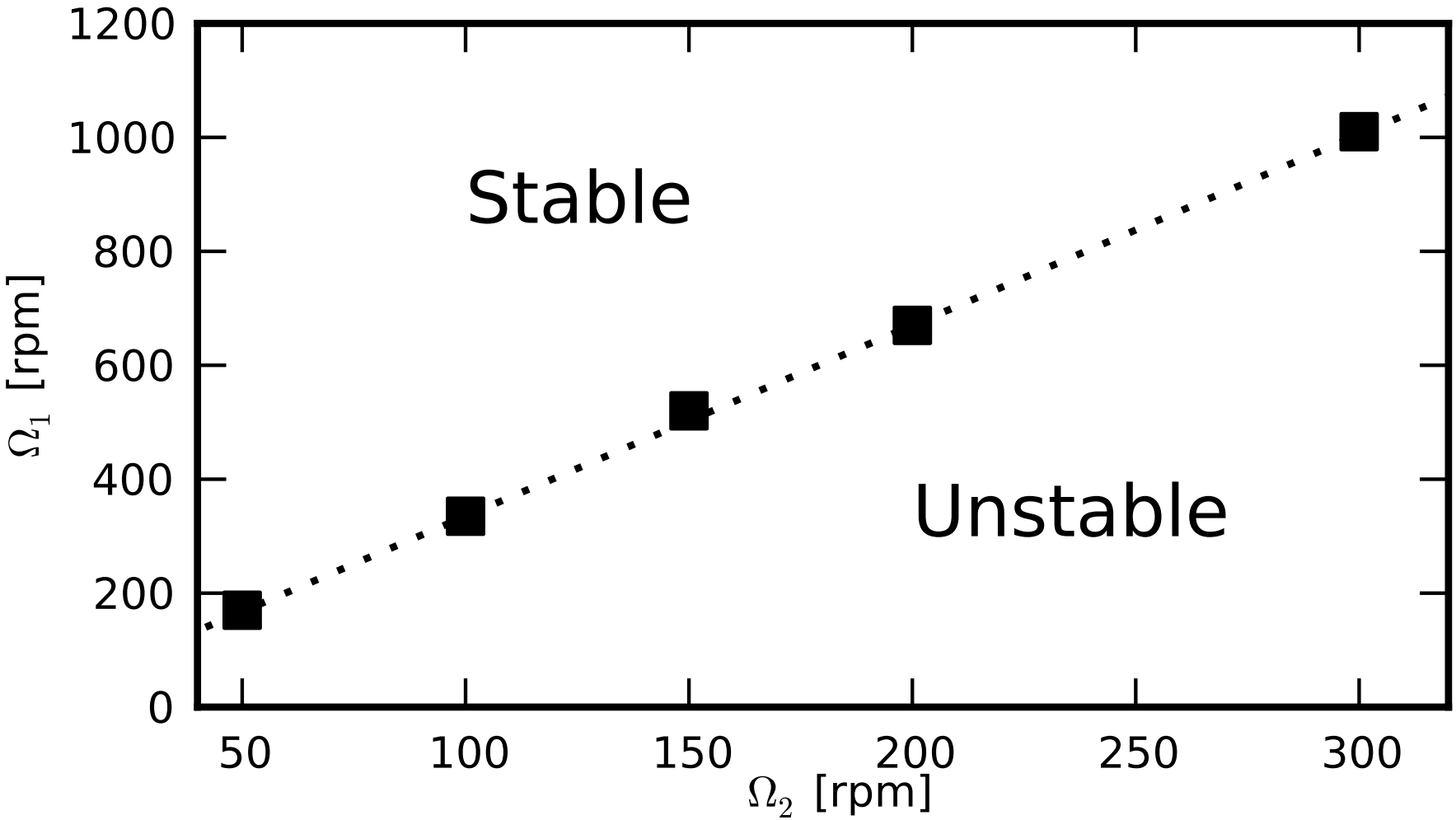


Use savefig to save your figures.

- Possible file types include EPS, PDF, PNG, and others.
- JPEG is not a valid type.
- Put this in your program which generates your figure for your paper!

SciNet compute • calcul CANADA

Final product



Another plotting example

```
In [63]:
```

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]:
```

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]:
```

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]:
```

Another plotting example

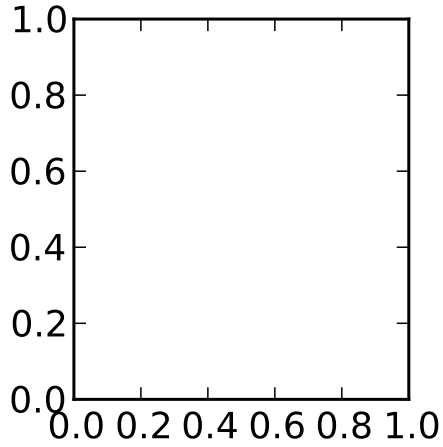
```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]:
```

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]:
```

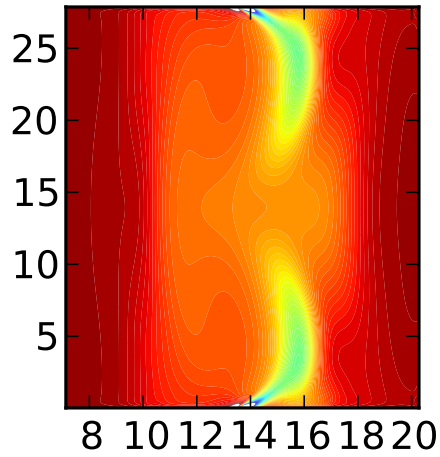
Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]:
```



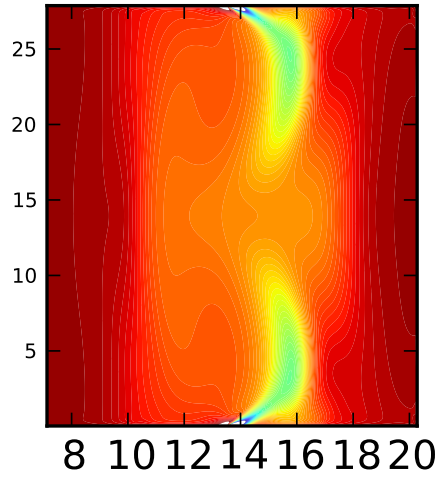
Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]: contourf(r, z, s, V)

In [70]:
```



Another plotting example

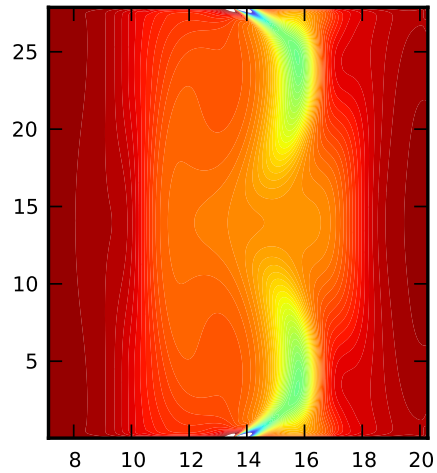
```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]: contourf(r, z, s, V)

In [70]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)
```




```
In [71]:
```

Another plotting example

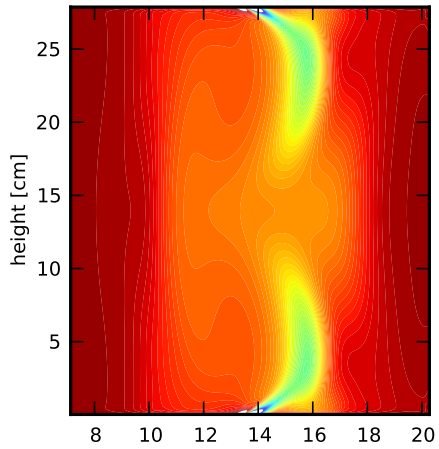
```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]: contourf(r, z, s, V)

In [70]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)
```




```
In [71]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [72]:
```

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]: contourf(r, z, s, V)

In [70]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)
```

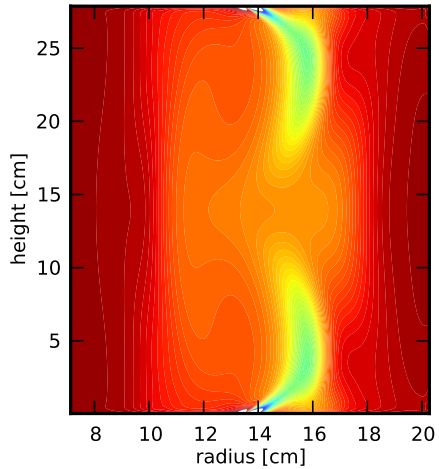



```
In [71]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [72]: ylabel('height [cm]',
  fontsize = 6,
  horizontalalignment='center')

In [73]:
```

compute • calcul CANADA

Another plotting example

```
In [63]: r,z,s,psi = calc_psi_pop2011()
In [64]: maxshear = 10.0; maxpsi = 42.0
In [65]: V = linspace(-maxshear, 0.0, 75)

In [66]: fig = figure(
  figsize = (3.375,2), dpi = 600)

In [67]: fig.subplots_adjust(top = 0.97,
  right = 0.98, left = 0.09,
  bottom = 0.12, wspace = 0.0,
  hspace = 0.0)

In [68]: a = fig.add_subplot(1,2,1)

In [69]: contourf(r, z, s, V)

In [70]: for t in
  a.yaxis.get_major_ticks():
  t.label.set_fontsize(6)
```

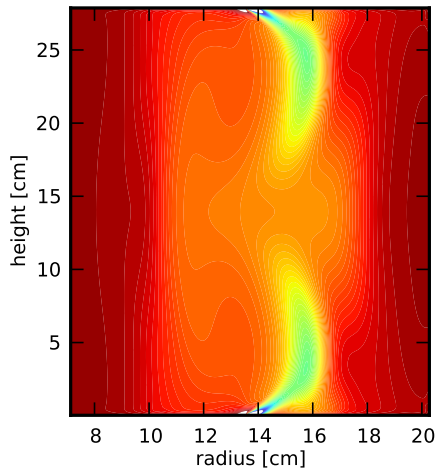



```
In [71]: for t in
  a.xaxis.get_major_ticks():
  t.label.set_fontsize(6)

In [72]: ylabel('height [cm]',
  fontsize = 6,
  horizontalalignment='center')

In [73]: xlabel('radius [cm]',
  fontsize = 6,
  verticalalignment='center')
```

compute • calcul CANADA

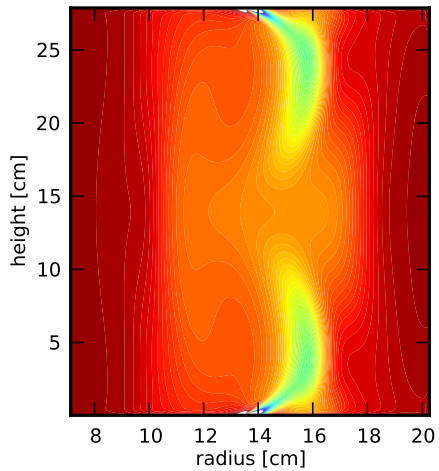
Another plotting example, continued

```
In [74]:
```



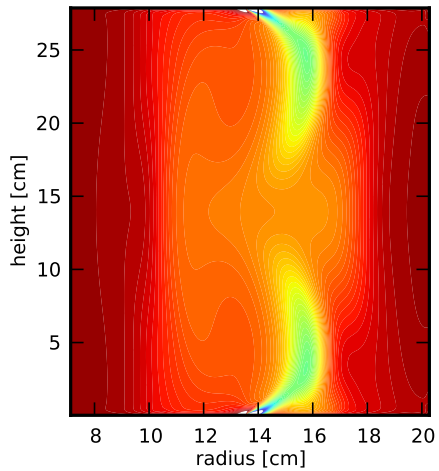
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]:
```



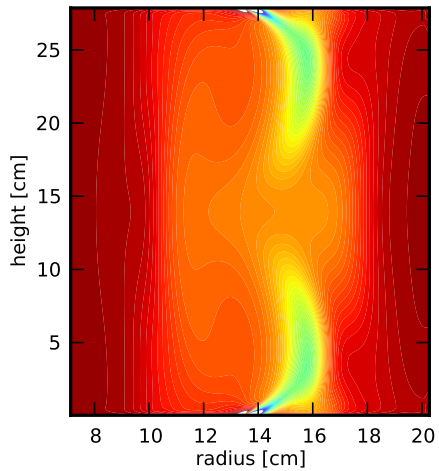
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]:
```



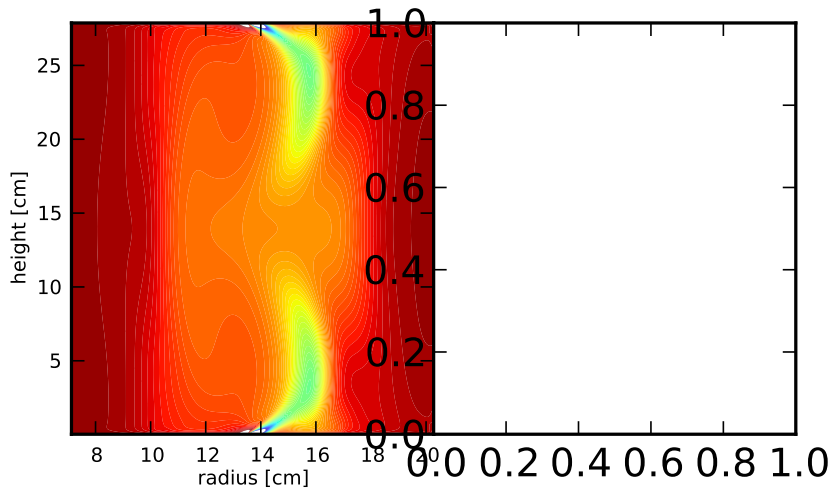
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]:
```



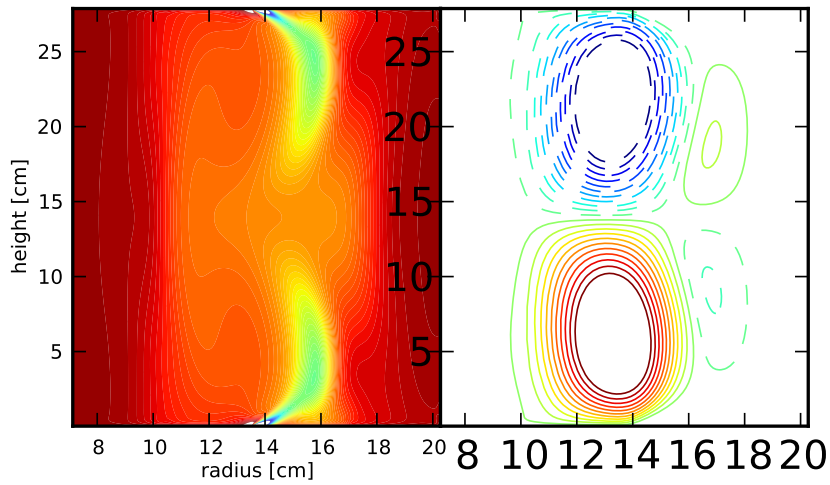
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]: a2 = fig.add_subplot(1,2,2)

In [79]:
```



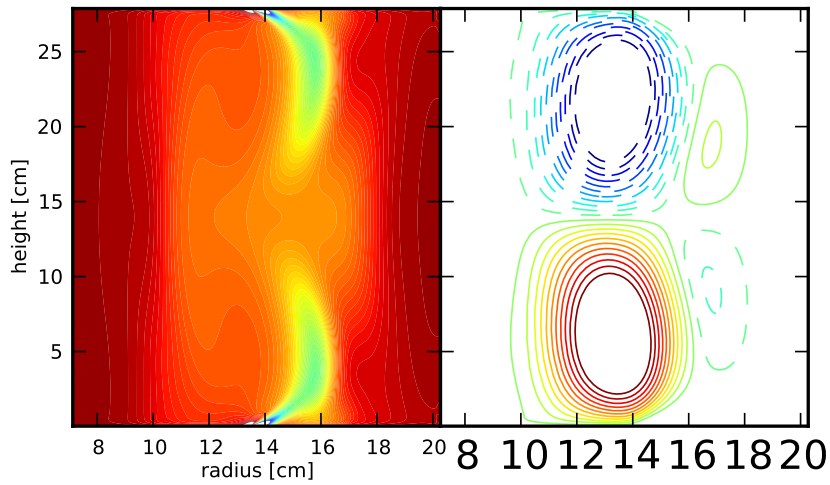
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]: a2 = fig.add_subplot(1,2,2)

In [79]: contour(r, z, psi, V2,
  linewidths = 0.5, linestyles = styles)

In [80]:
```



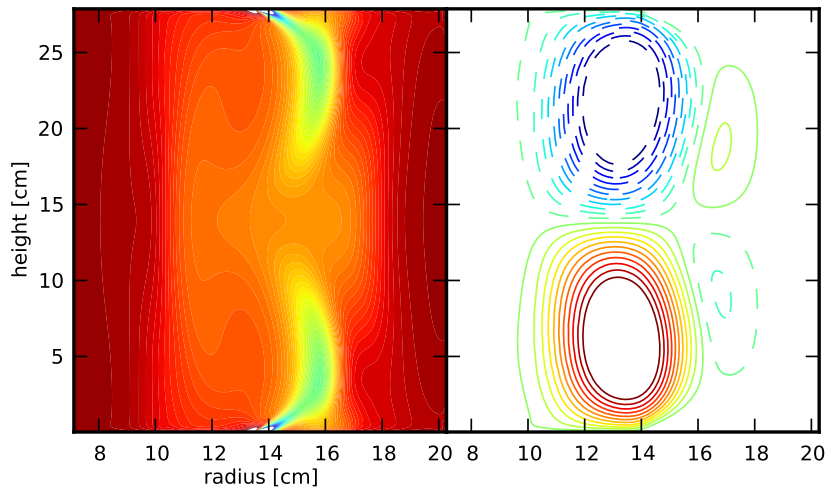
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]: a2 = fig.add_subplot(1,2,2)

In [79]: contour(r, z, psi, V2,
  linewidths = 0.5, linestyles = styles)

In [80]: a2.set_yticklabels([])

In [81]:
```



Another plotting example, continued

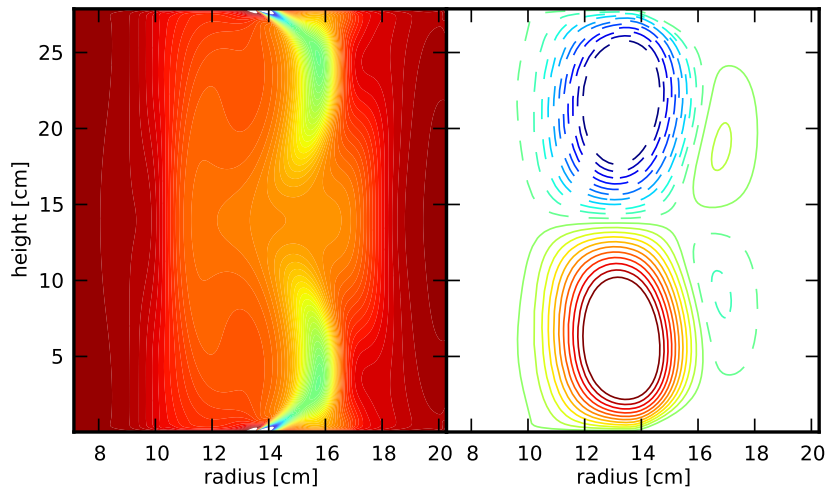
```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]: a2 = fig.add_subplot(1,2,2)

In [79]: contour(r, z, psi, V2,
  linewidths = 0.5, linestyles = styles)

In [80]: a2.set_yticklabels([])

In [81]: for t in
  a2.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```




```
In [82]:
```

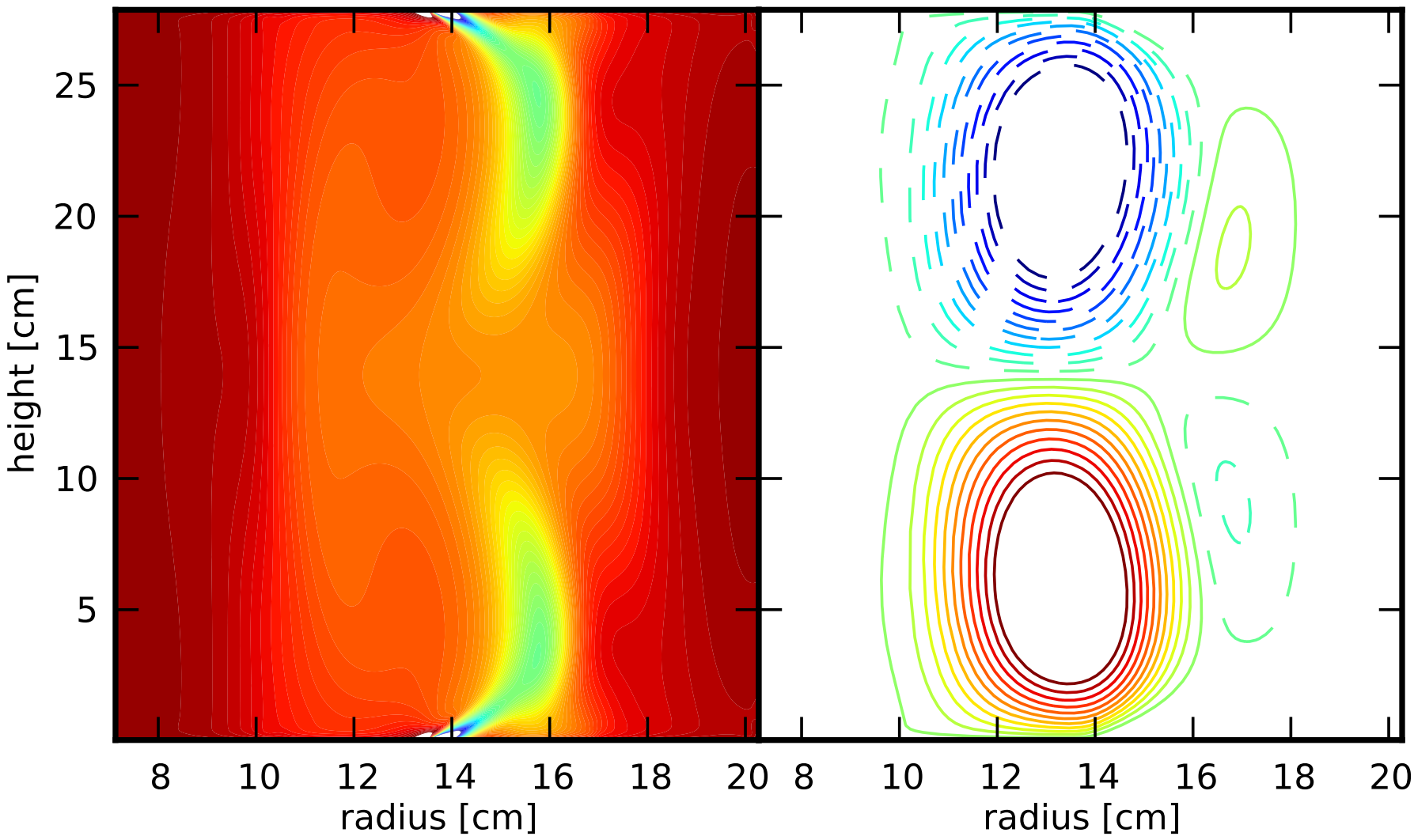
Another plotting example, continued

```
In [74]: V2 = linspace(-maxpsi,maxpsi,22)

In [75]: styles = []

In [76]: for i in range(11):
  styles.append('dashed')
In [77]: for i in range(11):
  styles.append('solid')

In [78]: a2 = fig.add_subplot(1,2,2)

In [79]: contour(r, z, psi, V2,
  linewidths = 0.5, linestyles = styles)

In [80]: a2.set_yticklabels([])

In [81]: for t in
  a2.xaxis.get_major_ticks():
  t.label.set_fontsize(6)
```




```
In [82]: xlabel('radius [cm]',
  fontsize = 6,
  verticalalignment='center')
```

Another final product



Animations

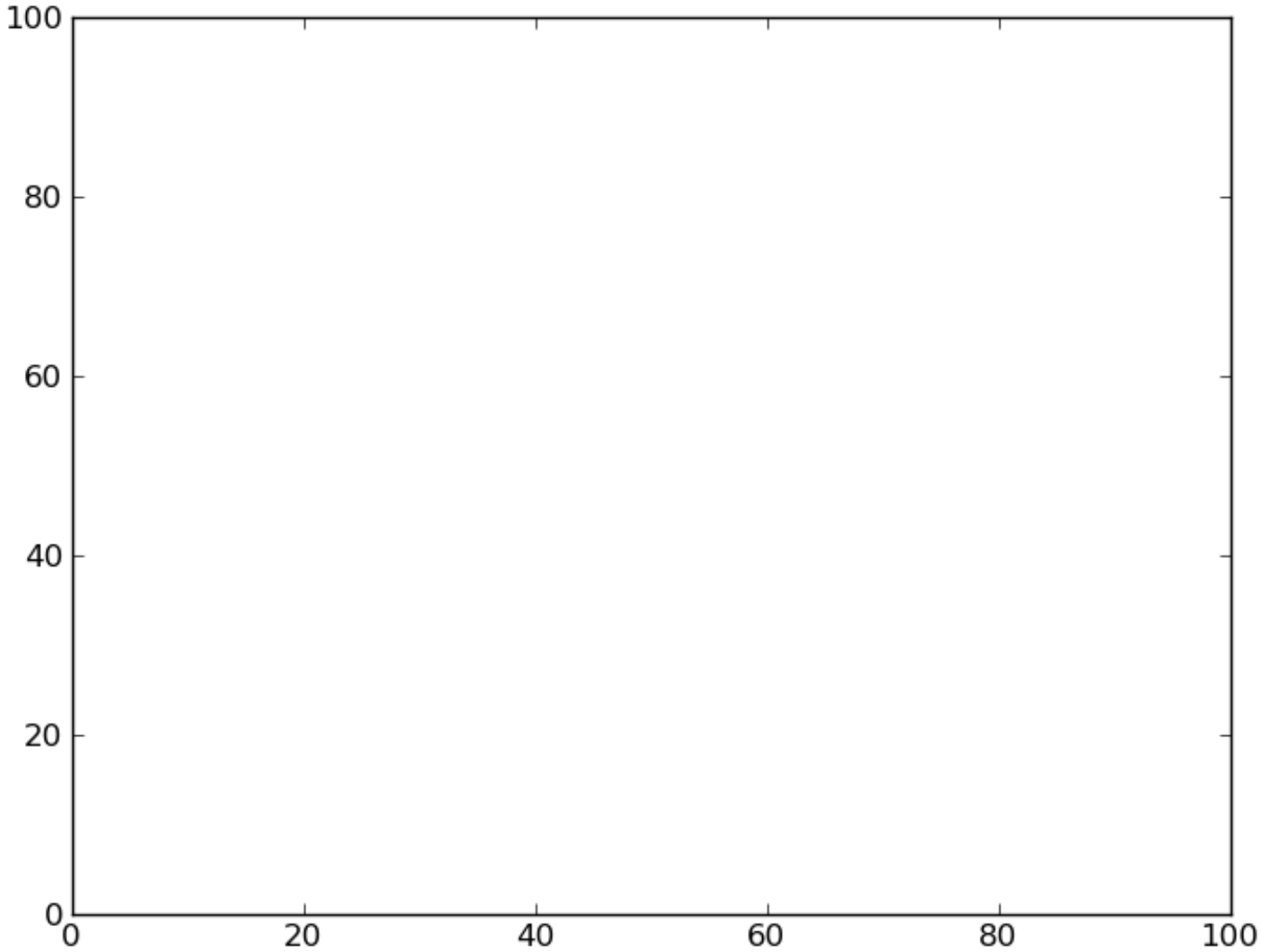
The best way to make animations in Python is to use the matplotlib.animation module.

```
# moviewriter.py
import numpy as np
import pylab as p
import matplotlib.animation as an

# Create an FFMPEG writer.
ffwriter = an.writers['ffmpeg']

# Metadata is good practice.
metadata = dict(title='My First
  Animation', artist='me')
```

Heavily stolen from matplotlib.org/examples/animation.

```
# Specify some features.
writer = ffwriter(fps = 15,
  metadata = metadata)

fig = p.figure()
# Open an empty plot.
l, = p.plot([], [], 'o')

p.xlim(-4, 4); p.ylim(-4, 4)
x0, y0 = 0.0, 0.0

with writer.saving(fig,
  'myanimation.mp4', 200)
  for i in range(200):
    x0 += np.random.randn()
    y0 += np.random.randn()

    # Specify the new x and y data.
    l.set_data(x0, y0)
    writer.grab_frame()
```

Our first animation



About animations

Some notes about animations in Python:

- Python itself does not create animations, it can only create images.
- Python streams the images into an external animation program.
- Two encoders are used, ffmpeg and mencoder. These must be installed on your computer separately for Python to access and use them.
- More advanced examples of animations can be found at `http://matplotlib.org/1.3.0/examples/animation`

SciNet compute • calcul CANADA

Homework 3

Questions for this week:

1. In the following two questions, use version control on your answers. Submit the output of 'hg log' for each question. We expect to see several commits.
2. Write a Python script which creates a professional-quality plot of your data. Use any data which is relevant to the work that you do. Please submit the code, the plot and the data which was used.
3. The NetCDF file, trajectory.nc, which is on the website, contains the positions and velocities for a set of particles at a number of points in time. Write a Python program which creates an animation of the data using just the positions at different times. Please submit both the code and your final animation.

SciNet compute • calcul CANADA