test-notebook

August 30, 2019

```
In [1]: from reaktoro import *

In [2]: db = Database('supcrt98.xml')

In [3]: editor = ChemicalEditor(db)
        editor.addAqueousPhaseWithElements('H O Na Cl C')
        editor.addGaseousPhase(['CO2(g)'])

Out[3]: <reaktoro.PyReaktoro.GaseousPhase at 0x7f4e05f61a40>

In [4]: # Step 4: Construct the chemical system
        system = ChemicalSystem(editor)
        print(system)

==========================================================================================
Aqueous                  Gaseous
------------------------------------------------------------------------------------------
CO(aq)                   CO2(g)
CO2(aq)
CO3--
Cl-
ClO-
ClO2-
ClO3-
ClO4-
H+
H2(aq)
H2O(l)
H2O2(aq)
HCO3-
HCl(aq)
HClO(aq)
HClO2(aq)
HO2-
Na+
NaCl(aq)
NaOH(aq)
O2(aq)
```

```
OH-
```

Index	Species	Element	Phase
0	CO(aq)	C	Aqueous
1	CO2(aq)	Cl	Gaseous
2	CO3--	H	
3	Cl-	Na	
4	ClO-	O	
5	ClO2-	Z	
6	ClO3-		
7	ClO4-		
8	H+		
9	H2(aq)		
10	H2O(l)		
11	H2O2(aq)		
12	HCO3-		
13	HCl(aq)		
14	HClO(aq)		
15	HClO2(aq)		
16	HO2-		
17	Na+		
18	NaCl(aq)		
19	NaOH(aq)		
20	O2(aq)		
21	OH-		
22	CO2(g)		

In [5]:
```
# Step 5: Define the chemical equilibrium problem
problem = EquilibriumProblem(system)
problem.setTemperature(60, 'celsius')
problem.setPressure(100, 'bar')
problem.add('H2O', 1.0, 'kg')
problem.add('NaCl', 1.0, 'mol')
problem.add('CO2', 10.0, 'mol')
```

Out[5]: <reaktoro.PyReaktoro.EquilibriumProblem at 0x7f4e046374c8>

In [6]:
```
# Step 6: Calculate the chemical equilibrium state
state = equilibrate(problem)
print(state)
```

Temperature [K]	Temperature [C]	Pressure [Pa]	Pressure [bar]
333.15	60	1e+07	100

Element	Amount [mol]	Aqueous [mol]	Gaseous [mol]
C	10	0.793515	9.20648
Cl	1	1	0
H	111.017	111.017	0
Na	1	1	0
O	75.5084	57.0955	18.413
Z	-1.32419e-16	-1.32419e-16	0

Species	Amount [mol]	Mole Fraction [mol/mol]	Activity Coefficient
CO(aq)	3.69283e-22	6.34245e-24	1
CO2(aq)	0.792322	0.0136082	1.22245
CO3--	5.69557e-10	9.78218e-12	0.154986
Cl-	0.922008	0.0158355	0.651298
ClO-	1.60807e-21	2.76188e-23	0.651298
ClO2-	2.33254e-22	4.00615e-24	0.651298
ClO3-	1.89895e-22	3.26146e-24	0.651298
ClO4-	1.39505e-22	2.39601e-24	0.641334
H+	0.00112022	1.92398e-05	0.634012
H2(aq)	4.31297e-22	7.40755e-24	1
H2O(l)	55.5072	0.95334	1.0167
H2O2(aq)	1.23781e-21	2.12595e-23	1
HCO3-	0.00119326	2.04942e-05	0.642544
HCl(aq)	7.30368e-05	1.25441e-06	1
HClO(aq)	3.89083e-20	6.68253e-22	1
HClO2(aq)	2.20932e-22	3.79452e-24	1
HO2-	4.0167e-22	6.89871e-24	0.651298
Na+	0.922081	0.0158368	0.642503
NaCl(aq)	0.077919	0.00133826	1
NaOH(aq)	5.19433e-11	8.92129e-13	1
O2(aq)	7.76627e-20	1.33386e-21	1
OH-	2.01562e-10	3.46184e-12	0.678583
CO2(g)	9.20648	1	0.657945

Phase	Amount [mol]	Stability	Stability Index [-]
Aqueous	58.224	stable	9.64327e-17
Gaseous	9.20648	stable	-4.71727e-22

Ionic Strength [molal]	pH	pE	Reduction Potential
0.923222	3.14859	11.3578	0.750799

```
In [7]: # Step 8: Print the amounts of some aqueous speciesk
        print('Amount of CO2(aq):', state.speciesAmount('CO2(aq)'))
        print('Amount of HCO3-  :', state.speciesAmount('HCO3-'))
        print('Amount of CO3--  :', state.speciesAmount('CO3--'))
        print('Amount of Na+    :', state.speciesAmount('Na+'))
```

```
Amount of CO2(aq): 0.7923219201161625
Amount of HCO3-  : 0.0011932550486751456
Amount of CO3--  : 5.695573946910143e-10
Amount of Na+    : 0.9220810121731402
```

```
In [8]: import sys
        from __future__ import print_function
        print('hi, stderr', file=sys.stderr)
```

```
hi, stderr
```

$$e^{i\pi} + 1 = 0$$

$$e^x = \sum_{i=0}^{\infty} \frac{1}{i!} x^i$$

```
In [9]: print('Amount of C in aqueous phase:', state.elementAmountInPhase('C', 'Aqueous'))
        print('Amount of C in gaseous phase:', state.elementAmountInPhase('C', 'Gaseous'))
```

```
Amount of C in aqueous phase: 0.7935151757343951
Amount of C in gaseous phase: 9.206484824265605
```

Code block:

```
print "Hello World"
def f(x):
    """a docstring"""
    return x**2
if (i=0; i<n; i++) {
  printf("hello %d\n", i);
  x += 4;
}
```

1 Heading 1

2 Heading 2

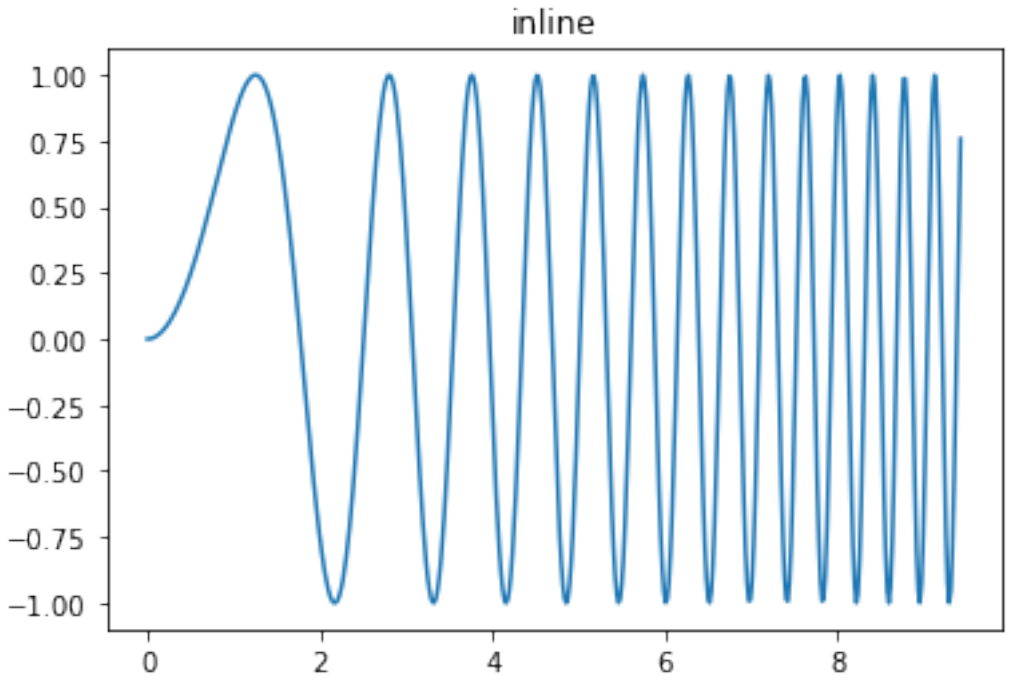
2.1 Heading 2.1

2.2 Heading 2.2

literal asterisks **literal asterisks**

In [10]:
```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 3*np.pi, 500)
```

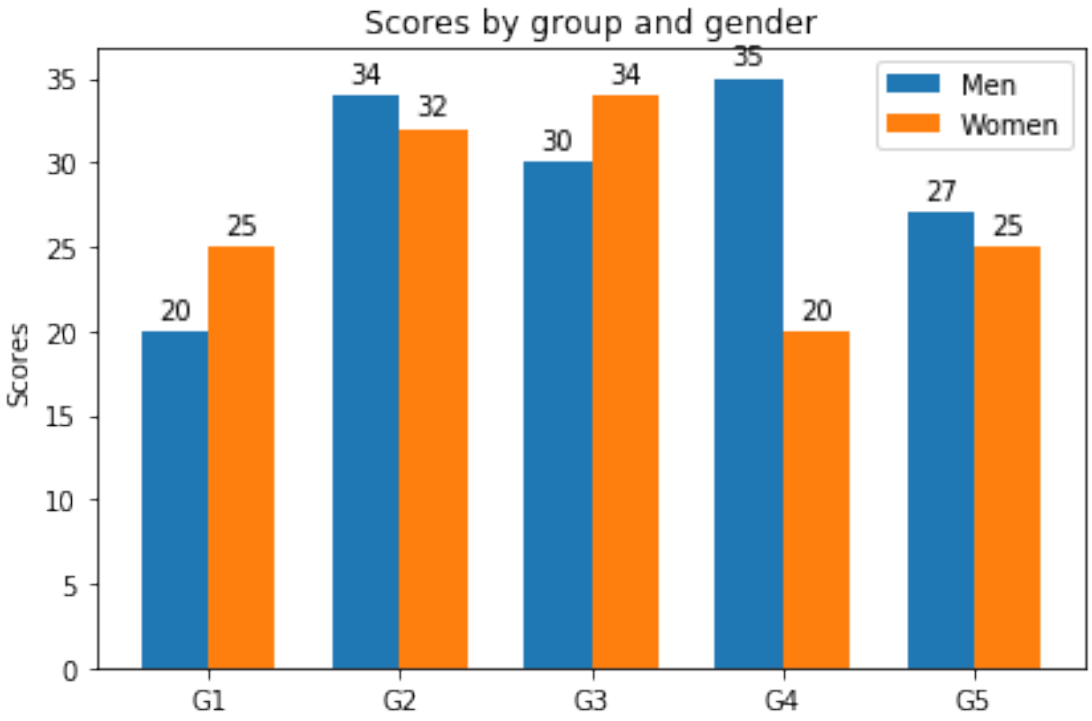
In [11]:
```
%matplotlib inline
plt.plot(x, np.sin(x**2))
plt.title('inline');
```



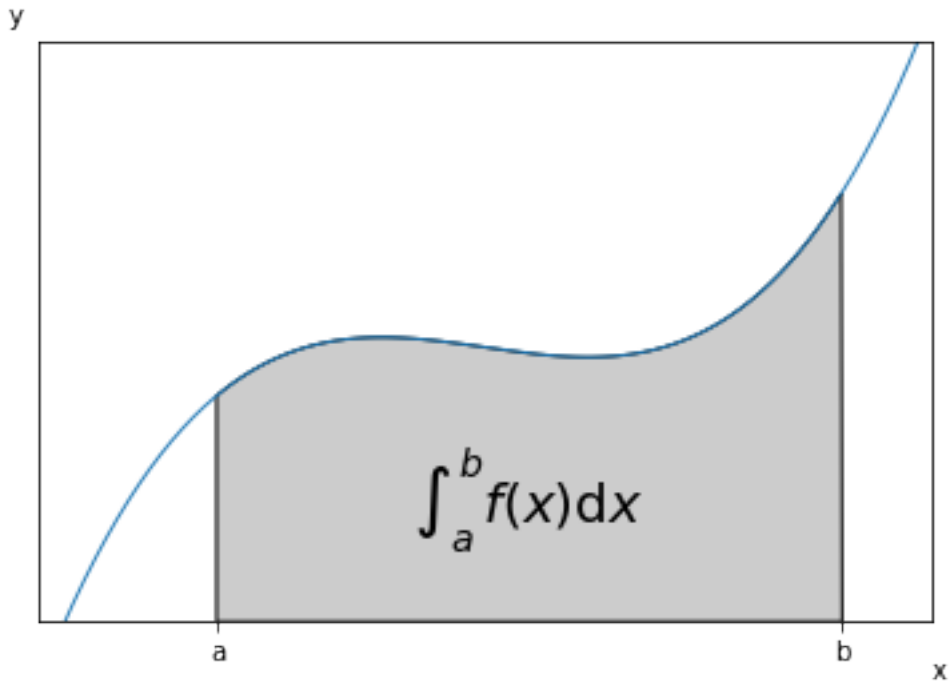
In [ ]:
```
#%load http://matplotlib.sourceforge.net/mpl_examples/pylab_examples/integral_demo.py
```

In [16]:
```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np


labels = ['G1', 'G2', 'G3', 'G4', 'G5']
men_means = [20, 34, 30, 35, 27]
women_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels))  # the label locations
width = 0.35  # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x - width/2, men_means, width, label='Men')
rects2 = ax.bar(x + width/2, women_means, width, label='Women')
```

```
# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Scores')
ax.set_title('Scores by group and gender')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()


def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')


autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()
```



In [17]:
```
#!/usr/bin/env python

# implement the example graphs/integral from pyx
from pylab import *
from matplotlib.patches import Polygon

def func(x):
    return (x-3)*(x-5)*(x-7)+85

ax = subplot(111)

a, b = 2, 9 # integral area
x = arange(0, 10, 0.01)
y = func(x)
plot(x, y, linewidth=1)

# make the shaded region
ix = arange(a, b, 0.01)
iy = func(ix)
verts = [(a,0)] + list(zip(ix,iy)) + [(b,0)]
poly = Polygon(verts, facecolor='0.8', edgecolor='k')
ax.add_patch(poly)

text(0.5 * (a + b), 30,
     r"$\int_a^b f(x)\mathrm{d}x$", horizontalalignment='center',
     fontsize=20)

axis([0,10, 0, 180])
figtext(0.9, 0.05, 'x')
figtext(0.1, 0.9, 'y')
ax.set_xticks((a,b))
ax.set_xticklabels(('a','b'))
ax.set_yticks([])
show()
```


y
$\int_a^b f(x)\mathrm{d}x$
a
b
x