SecondVirialCoeff

March 28, 2019

1 Homework 6 problem 1b –

1.1 Second virial coefficient

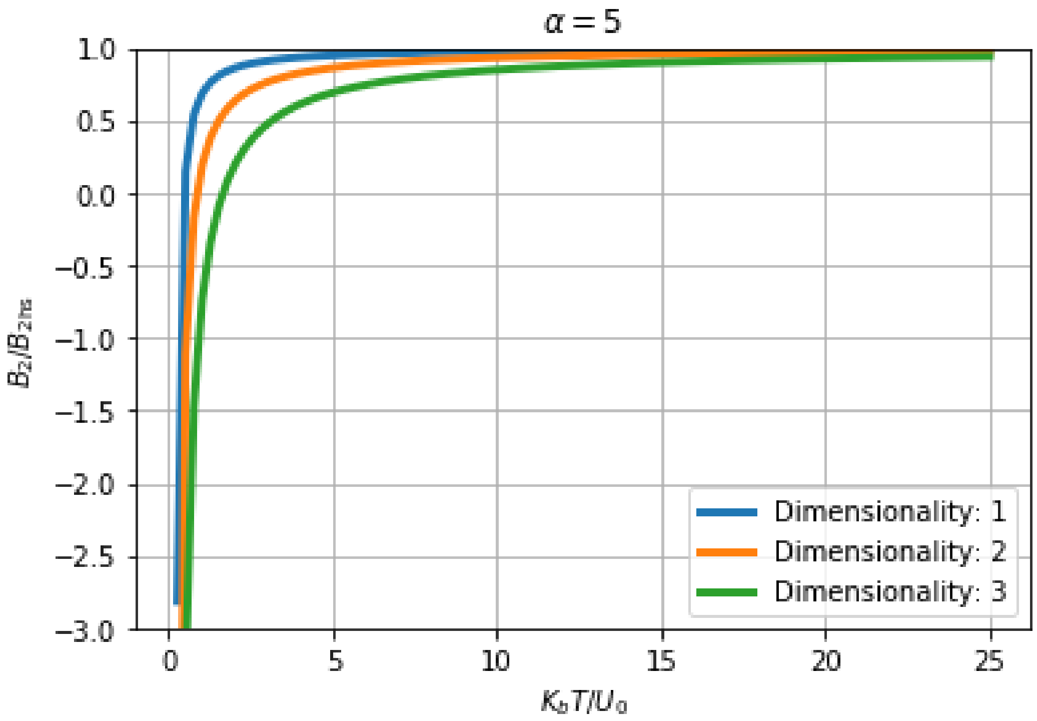
```
In [1]: from numpy import *
        from matplotlib.pyplot import *
        from scipy.integrate import quad #numerical quadrature
```

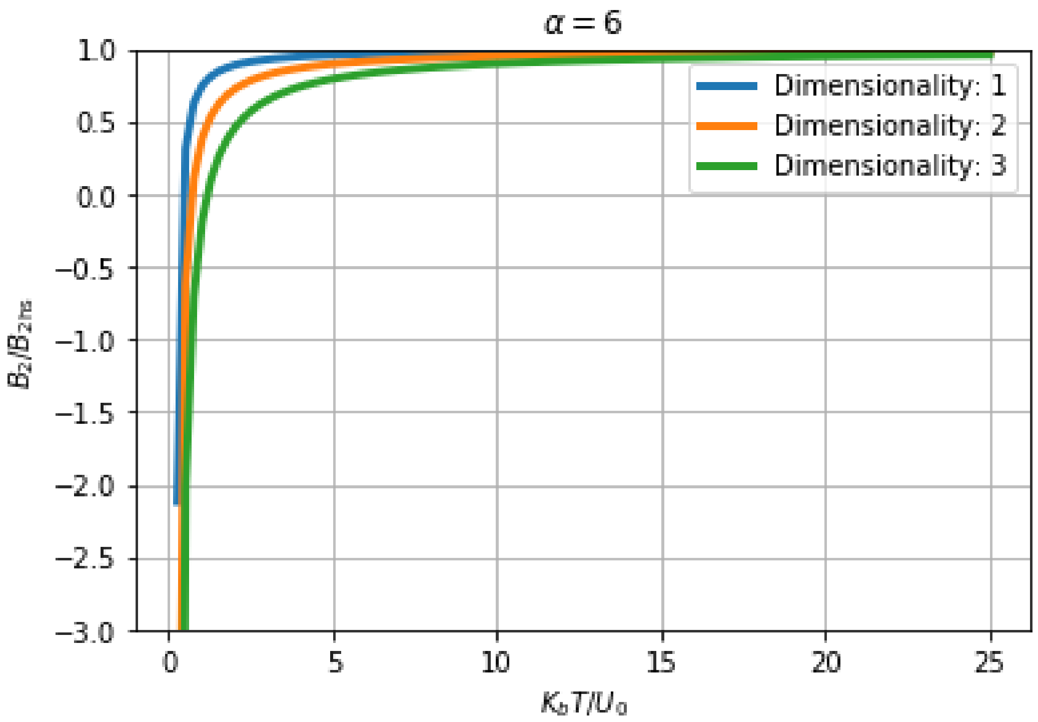
Plotting as a function of $x = \frac{K_B T}{U_0}$, and normalising with respect to the hard-sphere result, we get the following simplified expression for B_2:

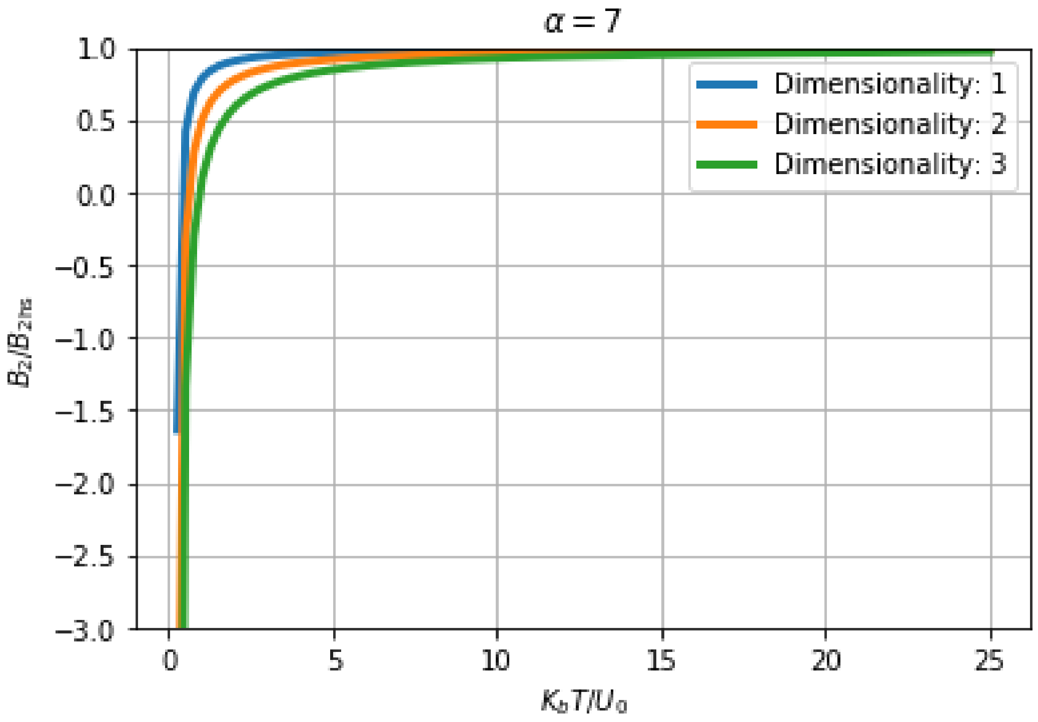
$$B_2(x) = d\left[\int_0^R drr^{d-1}(1-0) + \int_R^\infty drr^{d-1}\left(1-e^{1/x}\right)\right] \tag{1}$$

```
In [2]: alphas = array([5,6,7])
        dims = array([1, 2, 3])

        #radially symmetric pair potential
        def phi(r):
            if r > 1:
                return -(1/r)**alpha
            else:
                return inf

        #second virial coefficient
        def B2(x, d):
            beta = 1/x
            r1 = lambda r: r**(d-1)
            r2 = lambda r: r**(d-1)*(1-exp(-beta*phi(r)))
            return d *  ( quad(r1, 0, 1)[0] + quad(r2, 1, inf)[0] )

        def plotalpha(alpha):
            for d in dims:
                xvals = linspace(0,25,100)
                yvals = array([B2(x,d) for x in xvals])
                title(r'$\alpha = %s$' % alpha)
                plot(xvals, yvals, label='Dimensionality: %s' %d, lw=3)
                xlabel(r'$K_{b}T/U_{0}$')
                ylabel(r'$B_{2}/B_{2 \mathrm{hs}}$')
                grid()
                ylim(-3, 1)
                legend()
            show()

        for alpha in alphas:
            plotalpha(alpha)
```

```
/home/svein/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:13: RuntimeWarning: div
  del sys.path[0]
/home/svein/anaconda3/lib/python3.6/site-packages/scipy/integrate/quadpack.py:364: IntegrationW
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties.  If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges.  Perhaps a special-purpose integrator should be used.
  warnings.warn(msg, IntegrationWarning)
```




$\alpha = 6$
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
B_2/B_{2hs}
0
5
10
15
20
25
K_bT/U_0
Dimensionality: 1
Dimensionality: 2
Dimensionality: 3


$\alpha = 7$
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
B_2/B_{2hs}
0
5
10
15
20
25
K_bT/U_0
Dimensionality: 1
Dimensionality: 2
Dimensionality: 3