

Computer-based Exercises in Physical Chemistry

Raghunathan Ramakrishnan
ramakrishnan@tifrh.res.in

Tata Institute of Fundamental Research
TIFR Center for Interdisciplinary Sciences
Hyderabad, INDIA

26 December 2021



TIFR Centre **for**
Interdisciplinary
Studies

Why do we need

It's easy to plot and appreciate chem

Why should I learn about computers?

Often equations are difficult to solve manually.

Computer is a versatile equipment where one can perform several virtual experiments relevant for chemistry.

I want to have Python in my computer but I don't know how to install it. What to do?



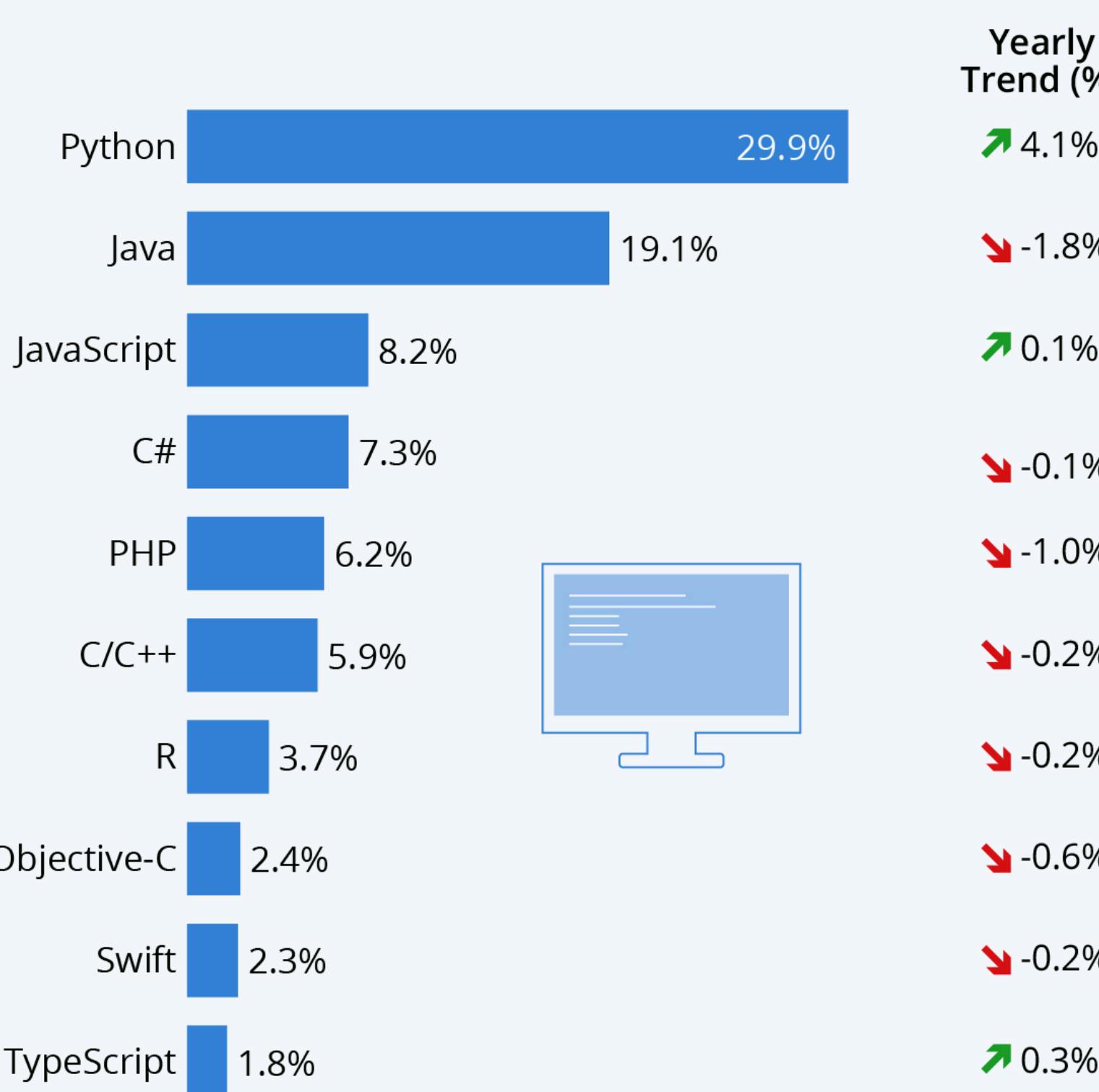
Don't be shy to ask around.

Take help from friends, teachers, or research scholars in your institute.

Choosing a programming language

Python Remains Most Popular Programming Language

Popularity of each programming language based on share of tutorial searches in Google



Yearly trend compares percent change from Feb 2019 to Feb 2020

Sources: GitHub, Google Trends

statista

Q: What's the best programming language to learn for science student with no previous programming experience

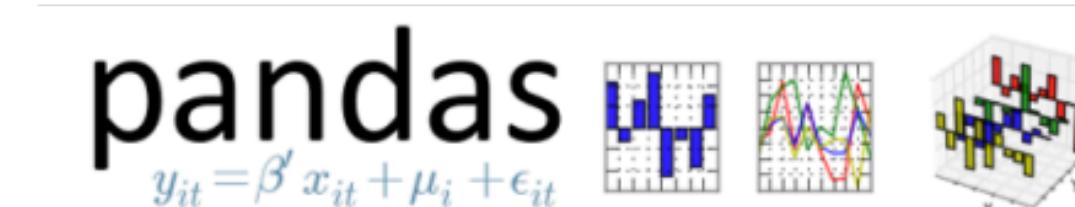
A: Python

Python is

- ❖ free
- ❖ easy to reference in the internet
- ❖ has a lot of libraries for visualisation, numerical methods, and data-analysis
- ❖ more libraries means less coding effort so that one can focus on the research problem at hand



IP[y]: IPython
Interactive Computing



Why should I learn about computers?

Plotting function

Simple Statistics

Solving simple problems

Solving Advanced problems

Mathematics and Numerical Methods

Plotting function

Simple Statistics

Solving simple problems

Solving Advanced problems

A warm up problem

1.0 atm of nitrosyl chloride is introduced into a reaction vessel. The compound dissociates into nitric oxide and chlorine according to the reaction



If the equilibrium constant of the reaction is known to be **2.18**, find the partial pressure of the gases at equilibrium

Answer

At equilibrium, $P_{\text{NOCl}} = 1 - 2x$, $P_{\text{NO}} = 2x$, and $P_{\text{Cl}_2} = x$.

Equilibrium-constant implies $\frac{P_{\text{NO}}^2 P_{\text{Cl}_2}}{P_{\text{NOCl}}^2} = \frac{(2x)^2 x}{(1 - 2x)^2} = K_{\text{eq.}} = 2.18$

The expression can be rearranged as a cubic equation $4x^3 - 8.72x^2 + 8.72x - 2.18 = 0$ which needs to be solved to determine the value of x (from which the partial pressures can be calculated)

Since $1 \geq P_{\text{NOCl}} \geq 0$ we know that $1 \geq 1 - 2x \geq 0$ or $x \geq 1/2$.

Cubic equation

We know how to solve a quadratic equation and find two solutions.

We are taught to rearrange the cubic equation into simple forms like (for example)

$(x - d)(ax^2 + bx + c) = 0$, in order to find the third solution.

Cubic equation

We know how to solve a quadratic equation and find two solutions.

We are taught to rearrange the cubic equation into simple forms like (for example)

$(x - d)(ax^2 + bx + c) = 0$, in order to find the third solution.

Now, let's see if a computer and Python can help us!

Graphical solution of the cubic equation

In [1]:

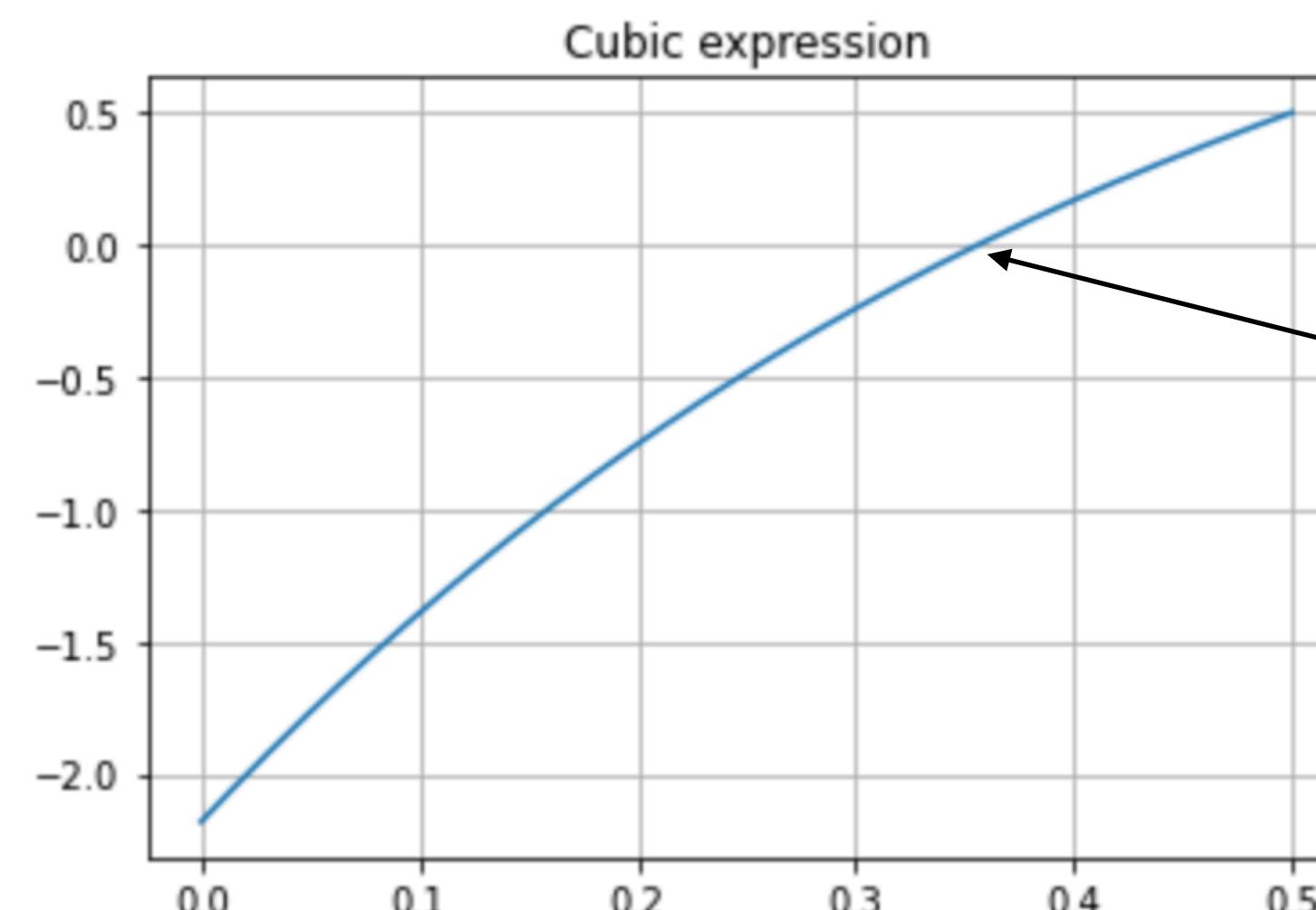
```
import numpy as np
import matplotlib.pyplot as plt

#--- x-values
x_min=0.0
x_max=0.5
x_grids=501
x=np.linspace(x_min, x_max, x_grids)

#--- f(x) values
f=4*x**3 - 8.72*x**2 + 8.72*x - 2.18

#--- plot
plt.plot(x,f)
plt.title('Cubic expression')
plt.grid()
plt.show()
```

x-range is fixed between 0 and 1/2 (using our previous knowledge of the problem)



The solution seems to be around $x = 0.35$

Numerical solution using secant method

```
In [2]: from scipy import optimize

def f(x):
    f=4*x**3 - 8.72*x**2 + 8.72*x - 2.18
    return f

xguess=0.1
solution=optimize.root_scalar(f, fprime=None, x1=xguess+0.01, method='secant', bracket=None, x0=xguess, \
                               options={'tol':1e-6, 'maxiter':100})
solution
```

```
Out[2]:      converged: True
              flag: 'converged'
      function_calls: 7
      iterations: 6
          root: 0.3560857818097863
```

- ❖ Secant method is a modified version of the Newton-Raphson method.
- ❖ Newton-Raphson requires that the derivative of the function is also known.
- ❖ In secant method, the derivative is approximated numerically using a finite-step (hence, finite-derivative)
- ❖ So, along with x_0 (the initial guess for the root), another point x_1 must also be specified.
- ❖ The derivative for the first iteration is estimated as $f'(x_0) \approx (f(x_1) - f(x_0))/(x_1 - x_0)$

The answer

At equilibrium, $P_{\text{NOCl}} = 1 - 2x$, $P_{\text{NO}} = 2x$, and $P_{\text{Cl}_2} = x$.

```
In [3]: x=solution.root  
print("The partial pressure of NOCl is ",1-2*x)  
print("The partial pressure of NO is ",2*x)  
print("The partial pressure of Cl2 is ",x)
```

```
The partial pressure of NOCl is  0.28782843638042743  
The partial pressure of NO is  0.7121715636195726  
The partial pressure of Cl2 is  0.3560857818097863
```

The IPython notebook can be downloaded from https://github.com/raghurama123/Comp_PhysChem_Basic

The answer

At equilibrium, $P_{\text{NOCl}} = 1 - 2x$, $P_{\text{NO}} = 2x$, and $P_{\text{Cl}_2} = x$.

```
In [3]: x=solution.root  
print("The partial pressure of NOCl is ",1-2*x)  
print("The partial pressure of NO is ",2*x)  
print("The partial pressure of Cl2 is ",x)
```

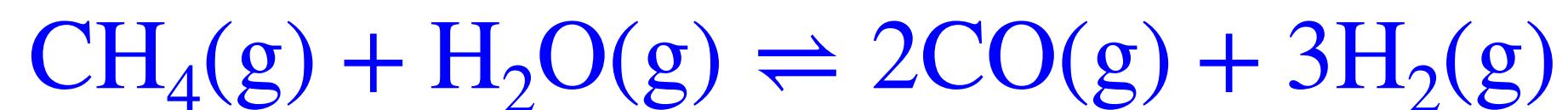
```
The partial pressure of NOCl is  0.28782843638042743  
The partial pressure of NO is  0.7121715636195726  
The partial pressure of Cl2 is  0.3560857818097863
```

Are you surprised that the sum of all partial pressures exceed the initial pressure of 1 atm?



Homework

Exercise 1: The following gas-phase reaction occurs at 300 K and has an equilibrium constant $K_{\text{eq.}} = 26$.



Initially 1.0 atm of methane and water (vapour) are introduced into the reaction vessel. Find the partial pressure of the species at equilibrium.

Answer: $P_{\text{CO}} = x = 0.615 \text{ atm}$. From this, you can find the values for other values.

Homework

Exercise 2: The following equation is encountered in the derivation of the Wien's displacement law. Find the solution of this equation graphically and by using the secant method.

$$e^{-x} + \frac{x}{5} = 1$$

Answer: $x = 4.965$

A problem in integration

Debye's theory of molar heat capacity (Debye- T^3 law) of a monoatomic crystal states

$$\bar{C}_V(T) = 9R \left(\frac{T}{\Theta_D} \right)^3 \int_0^{\Theta_D/T} \frac{x^4 e^x}{(e^x - 1)^2} dx$$

where R is the gas constant and Θ_D is the Debye temperature. Given that for copper, $\Theta_D = 309$ K, find its molar heat capacity at $T = 90$ K.

Experimentally measured value is $14.49 \text{ J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$

G. K. White and S. J. Collocott, *Heat Capacity of Reference Materials: Cu and W*
Journal of Physical and Chemical Reference Data 13, 1251 (1984)

<https://doi.org/10.1063/1.555728>

Numerical integration using quadrature

```
In [1]: import numpy as np
from scipy import integrate

R=8.314      # gas constant in J K^-1 mol^-1

Theta_D=309 # Debye Temperature of copper in K

T = 90      # given K at which we want molar heat capacity

def fn_I(x):
    fn_I= x**4 * np.exp(x) / (np.exp(x)-1)**2
    return fn_I

# For fine-tuning the integration settings see the scipy documentation
# https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.quadrature.html

Integral=integrate.quadrature(fn_I, 0.0, Theta_D/T)

print("The value of the integral is: ",Integral[0], " with a numerical error of ", Integral[1])

CV=9*R*(T/Theta_D)**3*Integral[0]

print("Heat capacity of Cu at T = 103 K is: ",CV, " J mol^-1 K^-1")
```

The value of the integral is: 7.9788851408858035 with a numerical error of 6.324787538147802e-08
Heat capacity of Cu at T = 103 K is: 14.751861725666032 J mol⁻¹ K⁻¹

Agrees well with the experimentally measured value: 14.49 J · mol⁻¹ · K⁻¹

Bonus: Plotting heat-capacities as a function of T

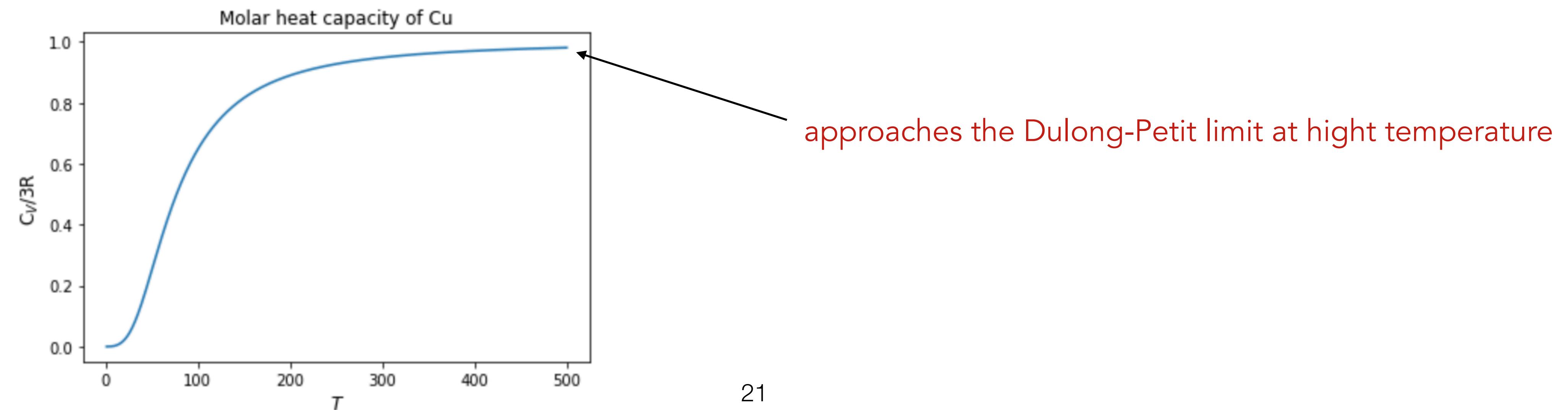
```
In [2]: import matplotlib.pyplot as plt

T_i=1
T_f=500
N_T=501
T=np.linspace(T_i, T_f, N_T)

Cv=np.zeros(501)

for i in range(501):
    Integral=integrate.quadrature(fn_I, 0.0, Theta_D/T[i])
    Cv[i]=9*R*(T[i]/Theta_D)**3*Integral[0]

plt.plot(T,Cv/(3*R))
plt.title('Molar heat capacity of Cu', fontsize=12)
plt.xlabel("$T$", fontsize=12)
plt.ylabel("C$_V$/3R", fontsize=12)
plt.show()
```



Homework

Exercise 3: The following integral is encountered in derivation of Stefan-Boltzmann's law. Determine its value using quadrature.

$$I = \int_0^{\infty} \frac{x^3}{e^x - 1} dx$$

Hint: Try different upper bounds like 1, 10, 100 and see how the integral converges.

Answer: $I = \frac{\pi^4}{15} = 6.494$

Energy of the simplest molecule, H₂⁺

One of the most interesting and useful applications of the variational method in quantum chemistry is to derive the total energy of the one-electron molecule H₂⁺ (Hydrogen molecule cation).

In this exercise, the H atoms are placed symmetrically at

$$\vec{r}_A = -\vec{R}/2 \text{ and } \vec{r}_B = +\vec{R}/2.$$

Each H-atom contains a **1s** orbital, described by the wave functions

$$\phi_A(\vec{r}) = \sqrt{\frac{\zeta^3}{\pi}} \exp(-\zeta |\vec{r} - \vec{r}_A|) \text{ and } \phi_B(\vec{r}) = \sqrt{\frac{\zeta^3}{\pi}} \exp(-\zeta |\vec{r} - \vec{r}_B|)$$

See: Quantum Chemistry, Donald A. McQuarrie (chapter 9)

Energy of the simplest molecule, H₂⁺

The wave function for the bonding molecular orbital (which is the ground state of the H₂⁺ molecule) turns out to be

$$\psi_{\text{bonding}} = \frac{1}{\sqrt{2}} [\phi_A(\vec{r}) + \phi_B(\vec{r})]$$

while the corresponding total energy of the molecule is the energy expectation value is a function of the exponent ζ and the inter-atomic distance $R = |\vec{r}_A - \vec{r}_B|$

$$E = \langle \psi | \hat{H} | \psi \rangle = f(\zeta, R)$$

According to variational principle, the ground state energy is given as the minimum energy (i.e. most negative energy) that can be obtained by varying the parameters ζ and R .

See: Quantum Chemistry, Donald A. McQuarrie (chapter 9)

Energy of the simplest molecule, H₂⁺

The final expression is given in textbooks as

$$E(\zeta, R) = \frac{J(\zeta, R) + K(\zeta, R)}{1 + S(\zeta, R)} + \frac{1}{R}$$

where the terms $J(\zeta, R)$, $K(\zeta, R)$, and $S(\zeta, R)$ are Coulomb, exchange and overlap integrals.

$S(\zeta, R) = \left(1 + \zeta R + \frac{1}{3}\zeta^2 R^2\right) \exp(-\zeta R)$, $J(\zeta, R) = \frac{1}{2}\zeta^2 - \zeta - \frac{1}{R} + \left(\frac{1}{R} + \zeta\right) \exp(-2\zeta R)$, and
 $K(\zeta, R) = -\frac{1}{2}\zeta^2 S(\zeta, R) - \zeta(1 + \zeta R)(2 - \zeta) \exp(-\zeta R)$. The term $\frac{1}{R}$ accounts for internuclear repulsion.

See: Quantum Chemistry, Donald A. McQuarrie (chapter 9)

Finding the minimum energy value graphically

```
In [1]: import numpy as np
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

def E(x,y):
    S=(1 + x*y + (1/3)*(x*y)**2) * np.exp(-x*y)
    J=(1/2)*x**2 - x -1/y + (1/y+x) * np.exp(-2*x*y)
    K=-(1/2) * S*x**2 - x * (1+x*y) * (2-x) * np.exp(-x*y)
    E=( J + K )/(1 + S) + 1/y
    return E

plt.rcParams["figure.figsize"] = (10,8)
fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
X, Y = np.mgrid[0.4:2.1:30j, 0.8:5:30j]

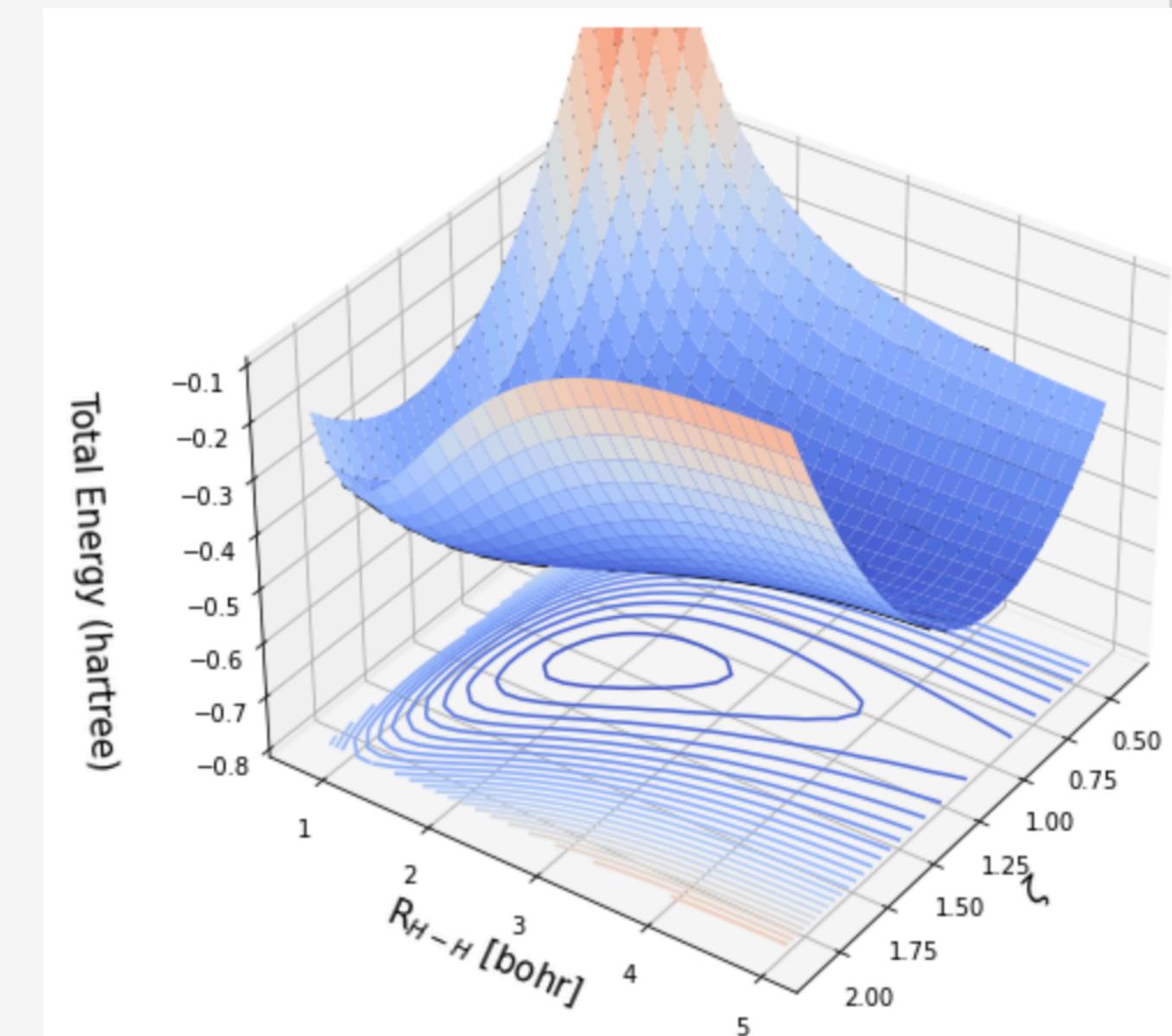
Z = E(X,Y)

ax.plot_surface(X, Y, Z, cmap="coolwarm", lw=1, rstride=1, cstride=1)
ax.contour(X, Y, Z, 40, cmap="coolwarm", linestyles="solid", offset=-0.8)
ax.contour(X, Y, Z, 40, colors="k", linestyles="solid")

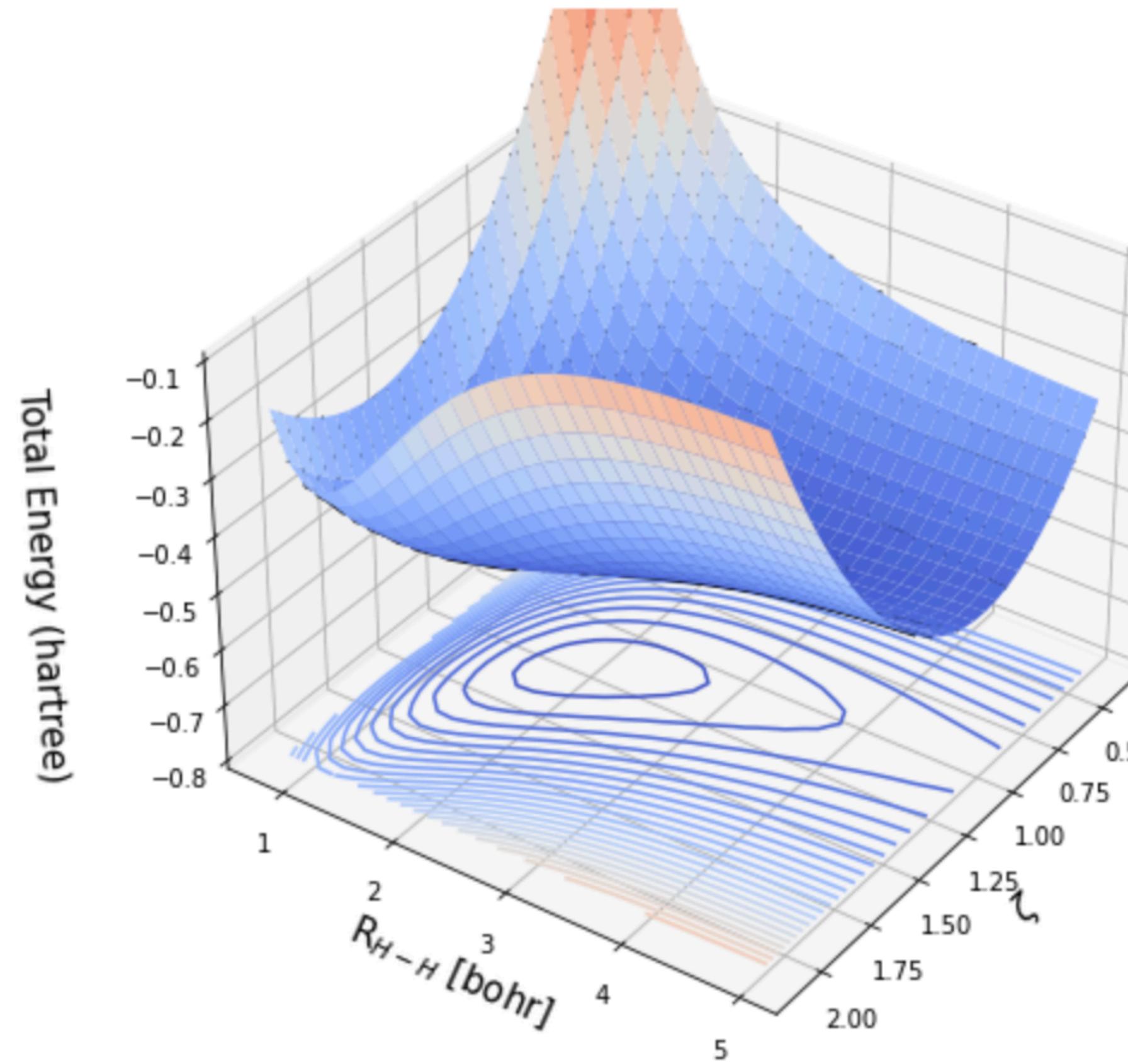
ax.set_xlabel('$\zeta$', fontsize=15)
ax.set_ylabel('R_{H-H} [bohr]', fontsize=15)
ax.set_zlabel('Total Energy (hartree)', fontsize=15, labelpad=20)

ax.set_zlim3d(-0.8,-0.1)
ax.view_init(35, 35)

plt.show()
```



Finding the minimum energy of H_2^+ graphically



```
In [2]: x_guess = 1.1 # zeta
y_guess = 2.0 # R in bohr
print('Eye ball estimation of the minimum energy is: ', E(x_guess,y_guess), ' hartree')
```

Eye ball estimation of the minimum energy is: -0.5754658938005228 hartree