Equilibrium Properties of the $H_2 + H + H^+ + e^-$ Reaction: MAE 727 Midterm Report

Kelvin Loh (20114534)

Room 3333, Department Of Aerospace Engineering

KAIST

kklloh@kaist.ac.kr
10/26/2011

Abstract

The equilibrium properties for the $H_2 + H + H^+ + e^-$ reaction are investigated at different high temperatures. The Boltzmann statistical interpretation was used to calculate the equilibrium constant, and various other properties such as the average internal energy of the atoms, and ions. Results show that even at high temperatures of up to 20000 K, the reaction still favors the formation of H_2 molecules (i.e. the number density of the H_2 molecule is dominant with respect to the other species).

1   Introduction

The primary reaction (1)

$$H_2 \rightleftharpoons H + H^+ + e^- \tag{1}$$

can be thought of to be a combination of two reactions. Mainly,

$$H_2 \rightleftharpoons H + H \tag{2}$$

and

$$H \rightleftharpoons H^+ + e^- \tag{3}$$

When both reactions (2) and (3) are combined, they form reaction (1). In order to calculate the equilibrium constant for reaction (1), K_{e1}, one can multiply the individual molar equilibrium constants for both reactions (2) and (3). This is because for a combination of reactions, the molar equilibrium constants are multiplicative.

$$K_{e1} = K_{e2} * K_{e3} = \frac{Q_H * Q_H^+ * Q_e}{N_A^2 * Q_{H_2}}$$
$$K_{e2} = \frac{[H][H]}{[H_2]} = \frac{Q_H^2}{N_A * Q_{H_2}} * exp(-D_0/kt) \tag{4}$$
$$K_{e2} = \frac{[H^+][e^-]}{[H]} = \frac{Q_H^+ * Q_e}{N_A * Q_H} * exp(-D_I/kt)$$

where Q are the individual partition functions of each species including the translational partition functions, and D_0, D_I are the dissociation and ionization energies respectively. Note that D_I is measured from the ground level of the atom, while D_0 is measured from the ground state of the molecule H_2 at the lowest vibrational mode.

As for the theory to calculate the number density ratios of the individual species from the partition functions, the book by Park [1] is a good resource.

2   General Description of the Code

A computer program (CSA_test) is used to obtain the individual partition functions of the molecule, atom, ion, and electrons of the reaction. The complete listings of the code are shown in the appendix.

The program starts with the input of all the relevant data such as the energy levels and states for the atom, and ion species. The input file "H_atom2.in" provided in the appendix is used for the program. The program then proceeds to calculate the first Bohr radius in order to obtain the cutoff radius either by density cutoff or Debye shielding length (whichever is the smallest). However, for now, the Debye shielding length calculation needs an input which is the number density of electrons N_e. Theoretically, one can obtain Ne for a particular pressure, however, this is not implemented in the program due to time constraints. Also the cutoff radius is not used in the calculation of the internal molecular partition functions.

After all the relevant inputs are given, the program then proceeds to calculate the partition functions of the atom, ion, and electron. The program then calls the vibrot subroutine which proceeds to calculate the partition function for the molecule based on a method by Vinokur, and Liu [2] and written by Prof. Park. A separate input file is needed which an excerpt is given "kklloh_H2_vibrot.inp" in the appendix. The subroutine also produces an output file "vibrot.out" which can be used for error checking.

The program then completes the calculation for the average electronic excitation energies of the atom, and ion (if given as an input), and then the individual reaction equilibrium constants. The results are then fed into a file named "CSAtest.out". The program then calls another subroutine called turnpt to obtain the potential function for the different molecule electronic states based on Jarmain's method. Again, this subroutine uses its own input file "turnptcoa.inp". The maximum vibrational modes which the turnpt subroutine requires, can be obtained from the "vibrot.out" file. The subroutine produces an output file "turnpt_kklloh.out" which contains all the coordinates for the energy level (in cm^{-1}), and the minimum and maximum radius (in Bohr radius) for that particular energy level. From there, post-processing of the results can be done using MATLAB or gnuplot or any other plotting programs. The results from this report are produced using MATLAB.

3   Results and Discussion

The numerical results of the computation are provided in an accompanying compressed file that came together with this report. Shown in this section are only the plots. Figure 1 shows the internal partition function for the atom and proton. The internal partition function for the H atom increases with temperature as expected and also is approximately constant at 2.0 until 15000 K. As for the proton, since it has no electronic structure, by default it has a value of 1.0.
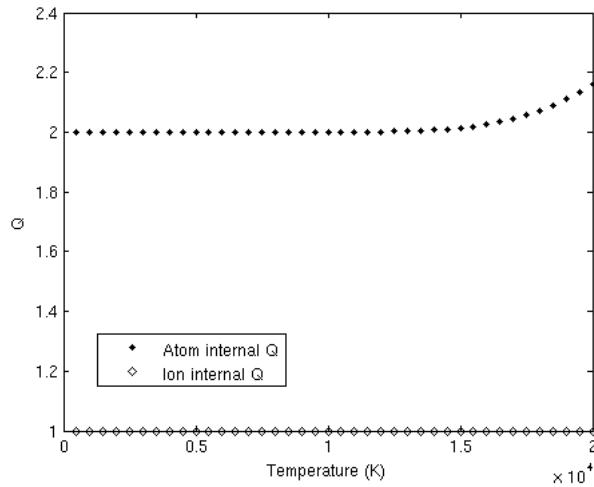

Figure 1: Internal Partition Function for Atoms and Ions

Figure 2 shows the translation partition function plot for both atom/ion (assuming mass of ion = mass of atom) and the molecule. It is shown here that the translation partition function for H_2 is always larger than either H of H^+.
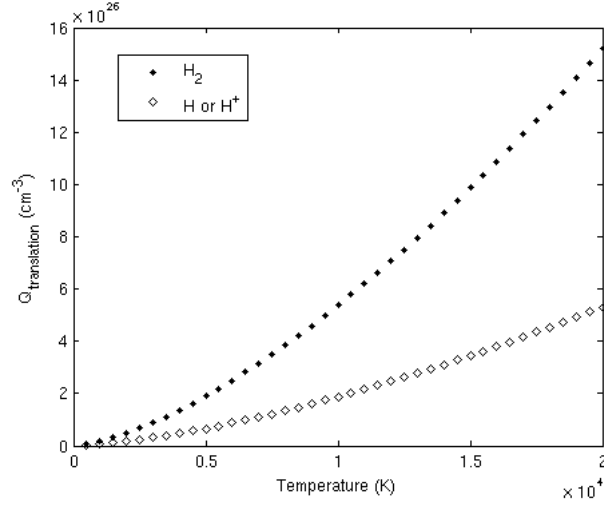

Figure 2: Translation Partition Function

Figure 3 shows the total partition functions of the individual species. It can be seen that the total partition function for the molecule H_2 is much larger than both atom and ion. One would then expect the number density of molecules to be larger than both atom and ion.
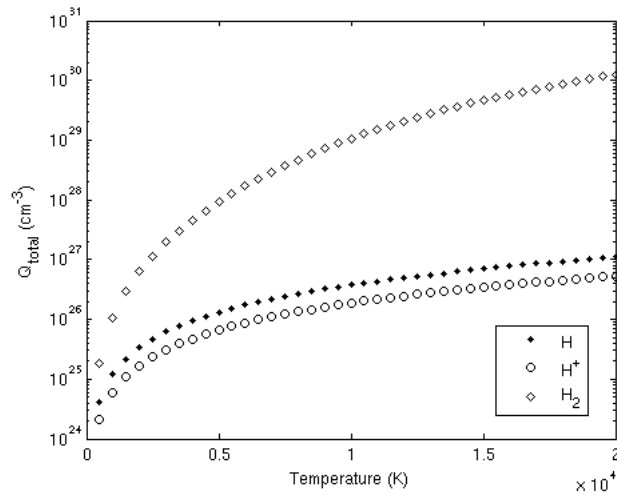

Figure 3: Total partition functions

Figure 4 shows the average electronic excitation energy and also the translational energy (for atom and ion) as calculated from the program. As expected, the internal energy for the atom increases with temperature.

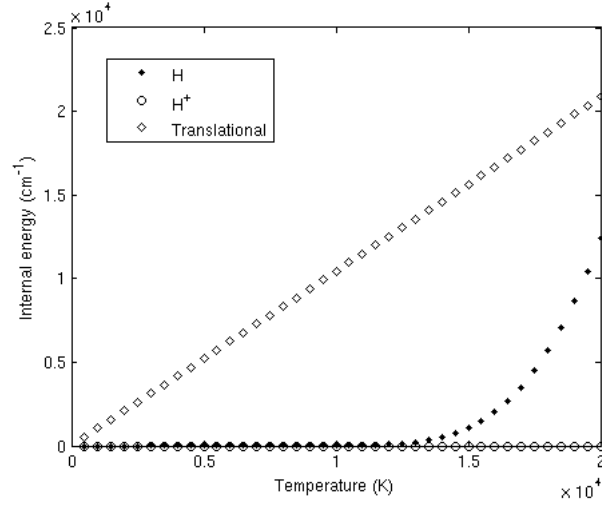Figure 4: Average internal energies

Figure 5 shows the equilibrium constants for all the reactions. It shows that the reaction favors the formation of the molecule H_2 even at the temperature of up to 20000 K. Although the numbers seem very small, one must remember that the K_{e1} value is normalized with N_A^2 due to the multiplication of K_{e2} and K_{e3}.
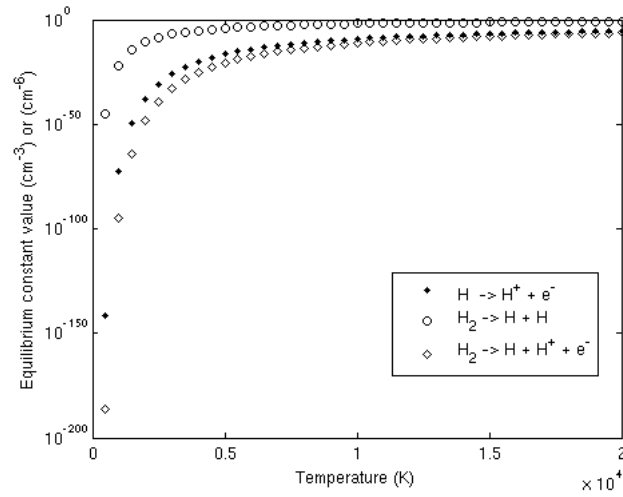

Figure 5: Equilibrium constants

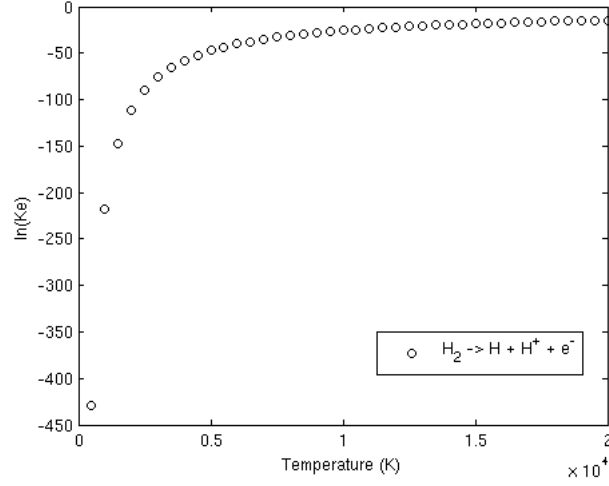A better form would be the log of K_{e1} which is shown in Figure 6.

Figure 6: $\ln(K_{e1})$

In order to visualize the potential function for the internal vibrations of the H_2 molecule, one can plot the coordinates given by the turnpt subroutine as shown in Figure 7.
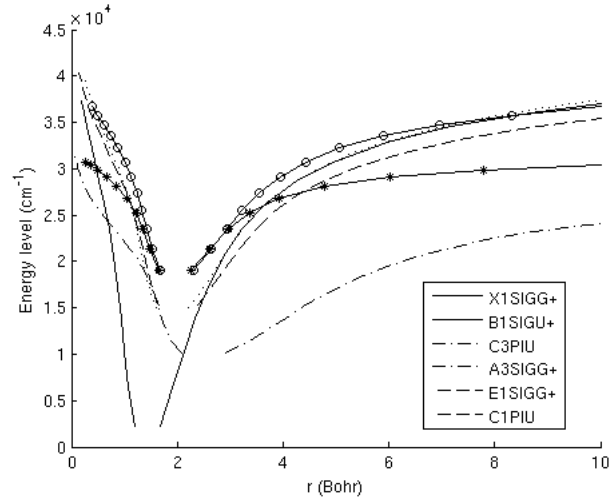

Figure 7: Potential function for H_2 using Jarmain's method

4   Summary and Future Work Discussions

In summary, the work done can provide insight as to the equilibrium composition for the decomposition and ionization of the hydrogen molecule, and atom. It shows that the H_2 molecule as being stable even at temperatures approximately 20000 K. However, one thing to note is that a lot of errors might be included due to the assumption that the molecule has a potential cutoff radius which is smaller than that calculated by the density cutoff or the Debye shielding length, and this one needs to be investigated more thoroughly. Given the time constraints it was not possible for the author to complete that investigation. One can actually feed in the values for the cutoff radius into the vibrot subroutine and then from there inversely calculate the maximum energy level (and hence the maximum vibrational level) from the potential functions (Hulburt-Hirchfelder or Lippincott). Another additional function would be to calculate the ionization level for comparison with the Saha equation.

Acknowledgments

References

[1] Park, *Nonequilibrium Hypersonic Aerothermodynamics, 1st Ed.* (John Wiley & Sons, USA, 1990).

[2] M. Vinokur, and Y. Liu "*Equilibrium Gas Flow Computations.I. Accurate and Efficient Calculation of Equilibrium Gas Properties*", presented at the 24th AIAA Thermophysics Conference, AIAA paper 89-1736, Buffalo, New York, June 12-14, 1989.

A   Code listing for CSA_test.f

```fortran
program CSA_test
integer i, ij, iT, Ni, Nn, NT, Nmax
integer, parameter :: n = 500
integer, parameter :: long = selected_real_kind(9,99)
```

```
        integer , dimension(n) :: x, s, xi = 0
        real(kind=long), dimension(n) :: J, E, QE, Ji, Ei, QEi = 0.
        real(kind=long) :: S_QE, S_J, boltz, g, D0, DI, Mo, PI, h, k = 0.
        real(kind=long) :: S_QEI, S_JI, Ne, Ntot, deb, dm, elec, a0 = 0.
        real(kind=long) :: C, Na, Me, Ke, Kei, Ked, QTe, QP, QA = 0.
        real(kind=long) :: eps_i, eps_a, eps_m, eps_t, S_QE2, S_QE1,
     &                      SI, SII = 0.
        real temp, mol_parti, mol_partf, dtemp
        open(51,file='CSAtest.out')
c        open(53,file='SAHA_stuff.out')
        open(52,file='H_atom2.in')
        PI = 4.d0*atan(1.d0)
        Me = 9.1093897d-31
        Na = 6.02214129d23
        elec = 1.60217733d-19
        k = 1.3806488d-23
        C = 299792458.d0
        h = 6.62606957d-34
c        boltzmann's constant 1/cm.K
        boltz = k/(100.d0*h*C)
        read (52,*) Ni, Nn, Mo, D0, Ntot !Ni = number of energy levels for at
        read (52,*) temp, dtemp, NT !temp = initial temperature, dtemp = tem
        write(51,21) 'T (K)', 'qA (cm-3)', 'qP (cm-3)',
     & 'qTe (cm-3)', 'qTA (cm-3)', 'QM_int (cm-3)',
     & 'QM (cm-3)', 'eps_A (cm-1)', 'eps_i (cm-1)', 'eps_t (cm-1)',
     & 'Kei (cm-3)', 'Ked (cm-3)',
     & 'Ke (cm-6)', 'Log(Ke)'
c        temperature input at eV to K
c        temp = temp*11604.505
        do i=1,Ni+1
          read(52,*) x(i), s(i), J(i), E(i)
        end do
        do i=1,Nn
          read(52,*) xi(i), Ji(i), Ei(i)
        end do
```

```
c          Calculate first Bohr radius
           a0 = 0.529177249d-8 !in cm
           Ne = 100. !Placeholder number for electron number density
           do iT = 1,NT
c          Calculate density cutoff radius
           dm = (1.d0/Ntot)**(1.d0/3.d0) !in cm
c          Calculate Debye shielding length
           deb = 0.69*sqrt(temp/Ne) !in cm
           if(dm .gt. deb) then
           dm = deb
           end if
           Nmax = floor(sqrt(dm/a0))
           ij = 0
c          Search for the corresponding energy level of index Nmax = s
           do i=1,Ni
             if(Nmax .lt. s(i)) then
             Nmax = x(i) - 1
             ij = 1
             end if
           end do
           if (ij .eq. 0) then
             Nmax = x(Ni)
           end if
           do i=1,Nmax
             QE(i) = (J(i)+1.d0)*exp(-E(i)/(boltz*temp))
           end do
           do i=1,Nn
             QEi(i) = (Ji(i)+1.d0)*exp(-Ei(i)/(boltz*temp))
           end do
c          Calculate the atom internal partition function
           do i=1,Nmax
           S_QE = S_QE + QE(i)
           end do
c           do i=2,4
c           S_QE2 = S_QE2 + QE(i)
```

```fortran
c          end do
c          do i=1,2
c          S_QE1 = S_QE1 + QE( i )
c          end do
c          Calculate ion internal partition function
           do i=1,Nn
           S_QEI = S_QEI +QEi( i )
           end do
c          To account for hydrogen ion internal partition function (proton)
           if (Nn == 0) then
           S_QEI = 1.0 d0
           end if
c          Obtain the ionization energy of the atom measured from the ground st
           DI = E(Nmax+1)
c          Atom and ion translational partition function (Assuming negligible m
           QT = (2.d0*PI*Mo*k*temp/(1000.d0*Na*h*h))**(1.5)*1.d-6 !multiplicatio
c           QT = 1.878 e20*sqrt (Mo*temp)*Mo*temp
           QTe = (2.d0*PI*Me*k*temp/(h*h))**(1.5)*1.d-6 !electron translation p
c          Electron total partition function (QEe*QTe) QEe = 2.0
           QTe = 2.d0*QTe
c          Atom total partition function
           QA = S_QE*QT
c          Ion total partition function (QTH*QEH)
           QP = QT*S_QEI
c          Calculate molecular partition function
           mol_parti = 0.0
           mol_partf = 0.0
           call vibrot (temp,mol_parti ,mol_partf)
c          Calculate internal energy of atoms
           do i=1,Nmax
              eps_a = eps_a + E( i )*(J( i )+1.d0)*exp(-E( i )/( boltz *temp))
           end do
           do i=1,Nn
              eps_i = eps_i + Ei( i )*(Ji( i )+1.d0)*exp(-Ei( i )/( boltz *temp))
           end do
```

```
        eps_t = (3.d0)*boltz*temp/(2.d0)
        eps_a = eps_a/S_QE! + (3.d0)*boltz*temp/(2.d0)
        eps_i = eps_i/S_QEI
c       Calculate equilibrium constant Ke
        Ke = exp(-(DI+D0)/(boltz*temp))*QP*QTe*QA/(mol_partf*Na*Na)
        Kei = exp(-(DI)/(boltz*temp))*QP*QTe/(QA*Na)
        Ked = exp(-(D0)/(boltz*temp))*QA*QA/(mol_partf*Na)
        write(51,22) temp, S_QE, S_QEI, QTe, QT, mol_parti,
     & mol_partf, eps_a, eps_i, eps_t, Kei, Ked, Ke, log(Ke)
c        write(*,'(1p4E15.6)') eps_t
c        write(53,20) temp, QP*QP*(QTe*QTe)*exp(-(DI+D0)/(boltz*temp))/
c        &              (mol_partf*Na*Na*Na)
c        write(*,*) Nmax
   20 format (2(1p4E10.3,3X))
   21 format (14(A,5X))
   22 format (14(1p4E15.6,3X))
        temp = temp + dtemp
        S_QE = 0.d0
        S_QEI = 0.d0
        S_QE1 = 0.d0
        S_QE2 = 0.d0
        end do
c       Obtain the coordinates for the potential function based on Jarmain's
        call turnpt
        end
```

B Input file H_atom2.in

```
66 0 1.0 36118.11 1.d16
500 500 40
1    1         1              0.0000000000
2    2         1         82258.9191133
3    2         3         82259.2850014
4    2         1         82258.9543992821
5    3         1         97492.211200
```

6	3	3	97492.319611
7	3	1	97492.221701
8	3	3	97492.319433
9	3	5	97492.355566
10	4	1	102823.8485825
11	4	3	102823.8943175
12	4	1	102823.8530211
13	4	3	102823.894250
14	4	5	102823.9094871
15	4	5	102823.90949
16	4	7	102823.917091
17	5	1	105291.62867
18	5	3	105291.65209
19	5	1	105291.63094
20	5	3	105291.651993
21	5	5	105291.659796
22	5	5	105291.65983494
23	5	7	105291.66370
24	5	7	105291.66373033
25	5	9	105291.66607200
26	6	1	106632.1485242
27	6	3	106632.1620756
28	6	1	106632.1498416
29	6	3	106632.1620536
30	6	5	106632.1665697
31	6	5	106632.16656761
32	6	7	106632.16882614
33	6	7	106632.16882188
34	6	9	106632.17017701
35	6	9	106632.17017434
36	6	11	106632.17107779
37	7	1	107440.43850
38	7	3	107440.44703
39	7	1	107440.43933
40	7	3	107440.447022

41	7	5	107440.449866
42	8	1	107965.0491583
43	8	3	107965.0548753
44	8	1	107965.0497145
45	8	3	107965.0548660
46	8	5	107965.05677174
47	9	1	108324.72016
48	9	3	108324.72418
49	9	1	108324.72055
50	9	3	108324.724162
51	9	5	108324.725501
52	10	1	108581.99051
53	10	3	108581.99344
54	10	1	108581.99080
55	10	3	108581.9934250
56	10	5	108581.9944009
57	11	1	108772.34135
58	11	3	108772.34355
59	11	1	108772.34157
60	11	3	108772.343536
61	11	5	108772.344270
62	12	1	108917.11870
63	12	3	108917.12039
64	12	1	108917.11886
65	12	3	108917.1203787
66	12	5	108917.12094342
67	0	3	109678.771732

Combined input file for H, and H+, 1st column = energy level index, 2nd column = n−th state, third column = degeneracy, fourth column = energy level (cm−1)

C   Excerpt for kklloh_H2_vibrot.inp

spnm	spwt(kg/m)	nelec
H2	0.002016	12

reduced mass factr

 0.5039   0.5

STATE NAME (SYMBOL)

DEGEN	TERM	WE	WEXE	WEYE	REQUIL	DZERO
	BE	ALPHA	DE	BETA		

X1SIGG+

1.	0.	4401.213	121.3360	0.000	0.7414	

36117.40

	60.85300	3.062	0.04710	-0.00274		

B1SIGU+

1.	91700.80	1358.090	20.8880	0.000	1.2928	

26675.60

	20.01540	1.184	0.01625	-0.002165		