Hydrogen Diffusion in Fe-Mn-Al alloys: A kinetic Monte Carlo study

HiWi Project
by
Ankit Izardar

completed at

Department of Computational Materials Design
Max-Planck-Institut Für Eisenforschung GmbH
Düsseldorf

Supervisor

Dr. Eunan McEniry

Düsseldorf, 28 September, 2017

Contents

Chapter 1

Introduction

Structural steels remain the most versatile and fundamental materials and are considered essential for the growth of any economy. Among these, steels with high Mn content(18-28 wt.%) offers a combination of high strength and excellent ductility which originates from their stable austenitic microstructure.The high Mn content ensures a stable fcc crystal structure (austenite phase) under normal conditions. The addition of Al has several advantages (refer [1]).

Hydrogen embrittlement is one of the most fundamental problems of the modern structural steels. During the production and service, hydrogen diffuses into the materials and the local accumulation of H around the defects such as vacancies, dislocations, etc., leads to the degradation of mechanical properties (ductility) which is commonly referred to as H embrittlement. It is known that alloying high Mn steels with Al leads to the improvement in the mechanical properties but on the other hand, it's effect on the hydrogen diffusivity is not clearly understood. Therefore, obtaining deep insight into the effect of alloying Al on the hydrogen diffusivity at the atomistic level becomes important.

In our study, we employ kinetic Monte Carlo (kMC) simulations for this purpose. kMC simulations provides access to timescales of the order of seconds and longer by coarse graining the time evolution to the discrete rare events and thus saving computation time and effort. All the input data to our kMC model, i.e. locations of hydrogen interstitial sites, onsite energies, diffusion paths and the corresponding diffusion barriers for H atoms are extracted from DFT calculations.

Chapter 2

Method and Model

2.1 kinetic Monte Carlo algorithm

We employ kinetic Monte Carlo (kMC) [2, 3] simulations to investigate the hydrogen diffusivity in Fe-Mn-Al alloys under varying alloy content and temperatures. In kMC method, the time evolution of a system is studied by means of stochastic computer simulations. Fig. 2.1 shows a flowchart of a lattice based kMC algorithm.

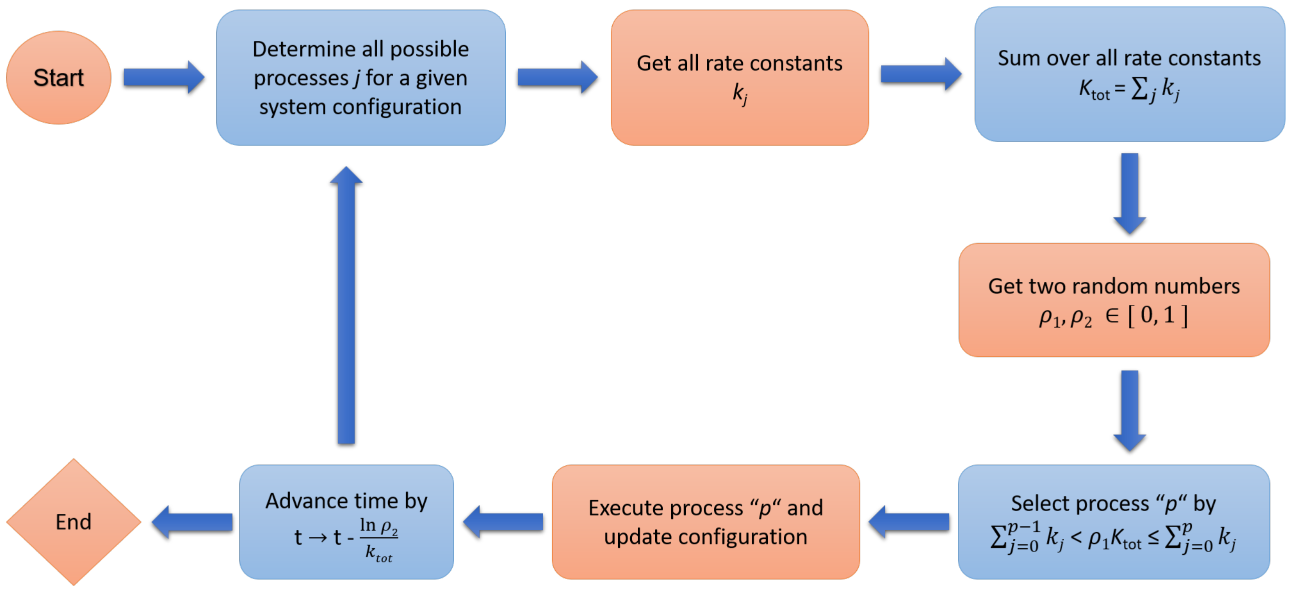

Start
Determine all possible processes j for a given system configuration
Get all rate constants k_j
Sum over all rate constants $K_{tot} = \sum_j k_j$
Get two random numbers $\rho_1, \rho_2 \in [0,1]$
Select process "p" by $\sum_{j=0}^{p-1} k_j < \rho_1 K_{tot} \leq \sum_{j=0}^{p} k_j$
Execute process "p" and update configuration
Advance time by $t \rightarrow t - \frac{\ln \rho_2}{k_{tot}}$
End

Figure 2.1: Lattice based kMC algorithm.

A rate catalog consisting a possible list of events is a critical input to the kMC method. These events are defined as diffusion jumps of H atoms and their rate constant k_j as defined by harmonic transition state theory can be written as

$$k_j = \nu_o exp(-\Delta E_j / k_B T)$$

where ν_o is the attempt frequency, ΔE_j is the energy barrier associated with a process j, k_B is the Boltzmann constant, and T is the temperature. The sum over all the possible rate constants, k_{tot} is evaluated as k$_{tot} = \Sigma_j k_j$.

In our model, we pre-calculate all the possible events and their associated barriers for all the interstitial sites. We then place a hydrogen atom on a randomly selected interstitial site. Out of all the possible events (jumps), a process p is selected such that it fulfills the following condition:

$$\sum_{j=0}^{p-1} k_j < \rho_1 k_{tot} \leq \sum_{j=0}^{p} k_j$$

where $\rho_1 \in [0,1]$ is a random number. Schematic Fig 2.2 illustrates this idea. Let us consider for each process j, we have a block of height k_j. On stacking each block on top of each other, we arrive at a stack of height k_{tot}. On choosing a random number between 0 and 1, we will point to one of the blocks. The selected process p will be then that block. The larger the rate constant, larger will be the height of that block and more will be the probability of it being chosen.

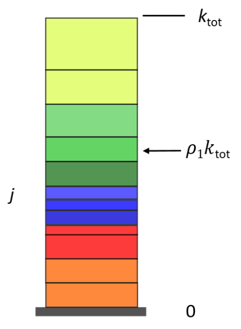


Figure 2.2: Stack of rate constant k_j with height k_{tot}.

After a process p is executed, time is advanced as follows:

$$t \to t - \frac{ln(\rho_2)}{k_{tot}}$$

Where ρ_2 is a random number is a second number between 0 and 1. This entire cycle repeats again till the maximum given number of steps are reached with each cycle propagating system to a new state i.e. a hydrogen jump from one O-site to another.

2.2 Model

Simulations were performed on fcc Fe with 10*10*10 supercell comprising of 4000 atoms with periodic boundary conditions in all three dimensions. Fig 2.3 shows a fcc structure. Positions shown by yellow colored balls can either be occupied by Fe, Mn or Al atoms while the pink colored balls represents the positions of the octahedral sites.

In our model, a lattice based kMC method is employed where the possible atomic positions are mapped onto a lattice with every second site representing an octahedral site (O-site). In fcc structures, each O-site is surrounded by 12 neighbouring O-sites and therefore, our system can evolve by hydrogen atom jumping from one O-site to one of its nearest 12 O-sites (see Fig. 2.4). Energy barriers are an important input in the kMC algorithm to calculate the rate constants. These were extracted from the DFT calculations (Eunan McEniry and Claas Hütter). The reference onsite energy for the octahedral sites surrounded by all i.e., 6 Fe atoms is taken as 0 eV. The activation energy for H jumping from one octahedral site to another under identical chemical environment (all Fe atoms) is 0.55 eV. The onsite energy of octahedral site is the function of number of Al and Mn atoms and scales linearly. On addition of one Mn atom,

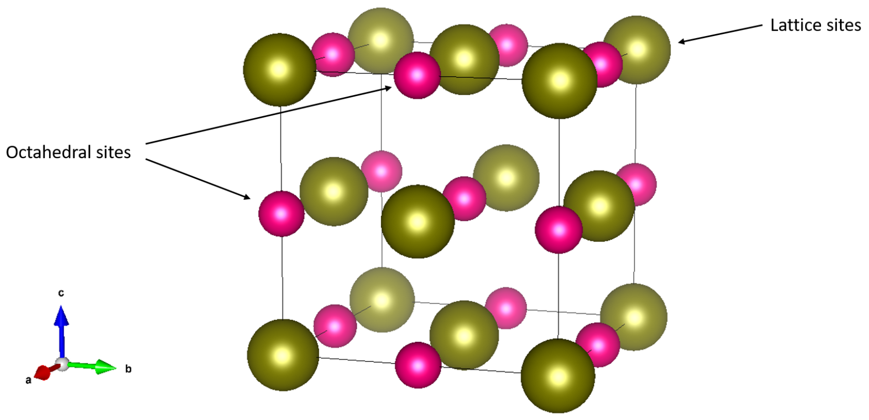


Figure 2.3: FCC structure with lattice and octahedral sites.

the onsite energy is lowered by 0.03 eV while the addition of one Al atom increases the onsite energy by 0.2 eV. Energy barrier is the function of number of Al and Mn atoms and also their positions in both octahedral sites and scales linearly. It is important to note here that while calculating onsite energies and barriers, volumetric effect and positions of atoms are not taken into account respectively.

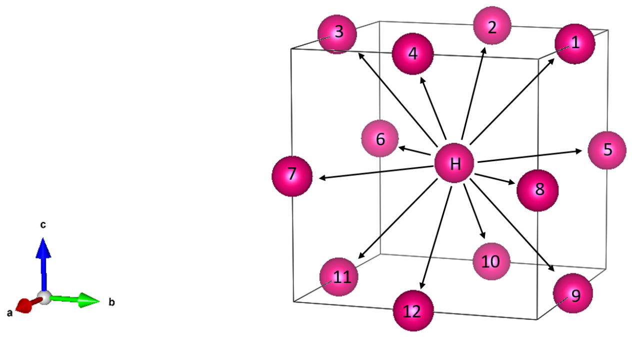


Figure 2.4: Possible jumps for a hydrogen atom.

Convergence tests revealed that 10k kMC steps per H atom gives well converged diffusivities. Diffusivity is calculated from the Einstein's diffusion relation i.e., by taking the square displacement of a hydrogen atom and then averaging it over 1000 H atoms as shown below:

$$D_i = \frac{[R^{final}_{x,y,z} - R^{initial}_{x,y,z}]^2}{6\Delta t}$$

$$D =< D_i >$$

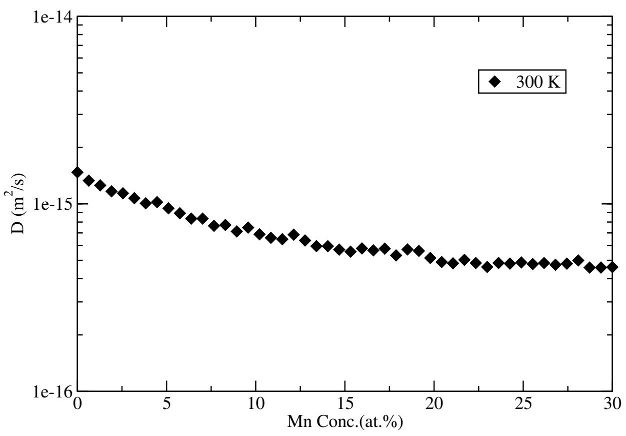
where D is the diffusivity for each hydrogen atom, $R^{initial}$ and R^{final} are the initial and final positions of hydrogen atoms respectively and Δt is the cumulative time. Diffusivity calculations in this work does not include the correlated diffusion i.e., H - H interactions.

Chapter 3

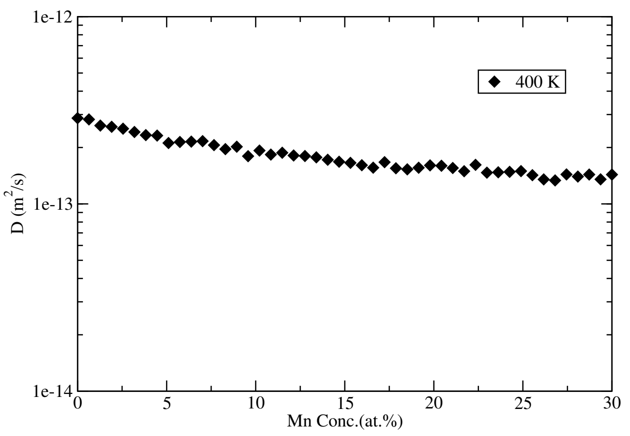
Results and Discussion

3.1 Effect of Mn

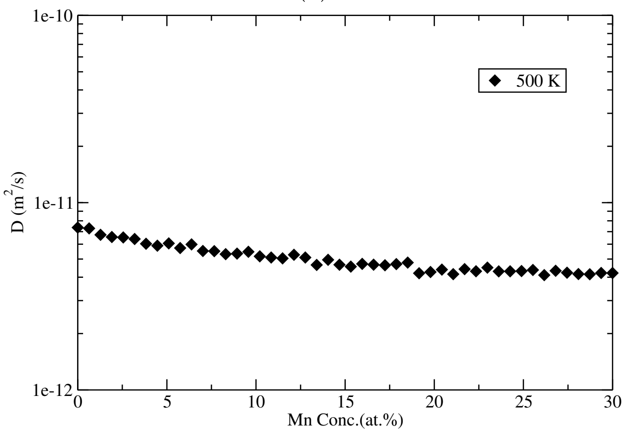
To study the effect of Mn on hydrogen diffusivity, Mn atoms were introduced randomly onto the pure iron matrix. In Fig. 3.1, hydrogen diffusivity is plotted as a function of Mn concentrations (0 at.% to 30 at.%) for a temperature range of 300 K - 600 K.



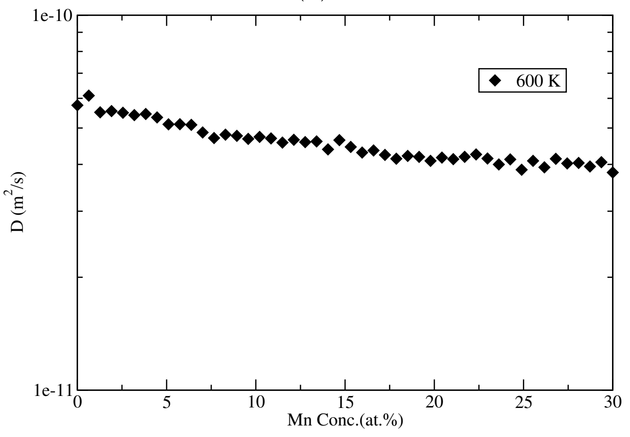
(a)



(b)



(c)



(d)

Figure 3.1: Hydrogen diffusivity as a function of Mn concentration. Results are shown for (a) 300 K, (b) 400 K, (c) 500 K, and (d) 600 K.

It is observed that decrease in hydrogen diffusivity is more at room temperature as compared to higher temperatures when Mn concentration is increased to 30 at.%. One can also notice that on increasing the Mn content from 0 to 10 at.% at room temperature, a significant decrease in H diffusivity is observed.

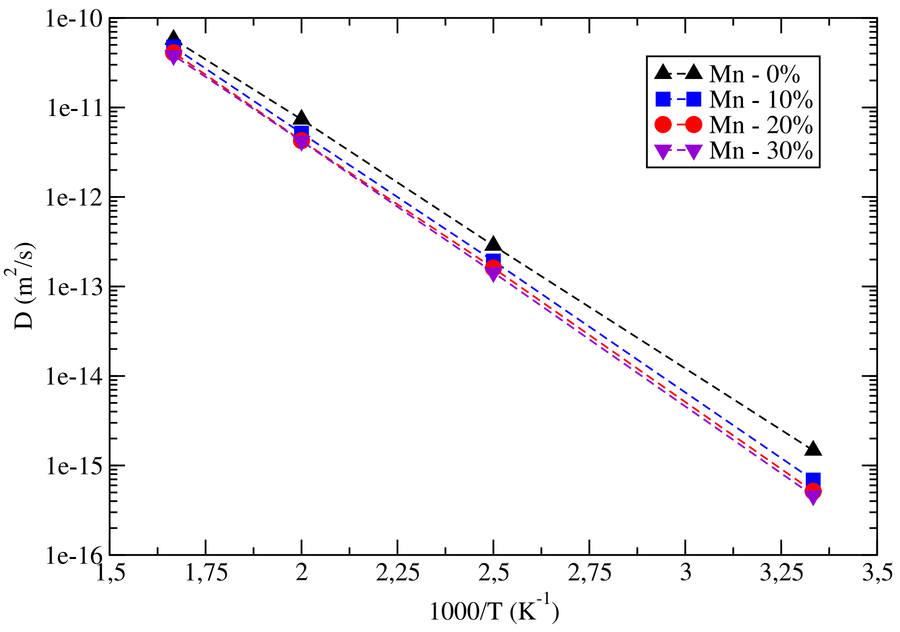


Figure 3.2: Logarithm of the hydrogen diffusivity as a function of the reverse temperature for four different Mn concentrations.

Fig. 3.2 shows the variation in hydrogen diffusivity as a function of inverse of the temperature for 0 at.% Mn, 10 at.% Mn, 20 at.% Mn and 30 at.% Mn. For a particular Mn concentration, hydrogen diffusivity increases by two order of magnitudes on increasing the temperature from 300 K to 400 K while it increases only by one order of magnitude on subsequent increase in temperatures. We can also observe that for lower temperature (300 K), hydrogen diffusivity decreases more when Mn content is increased from 0 at.% to 10 at.%. while this decrease is less on subsequent increase in Mn content.

3.2 Effect of Al

In order to investigate the effect of adding Al on hydrogen diffusivity in a Fe-Mn matrix, five concentrations of alloy content were studied. Fig. 3.3 shows a contour plot where hydrogen diffusivity is shown as a function of Al and Mn concentration at room temperature.

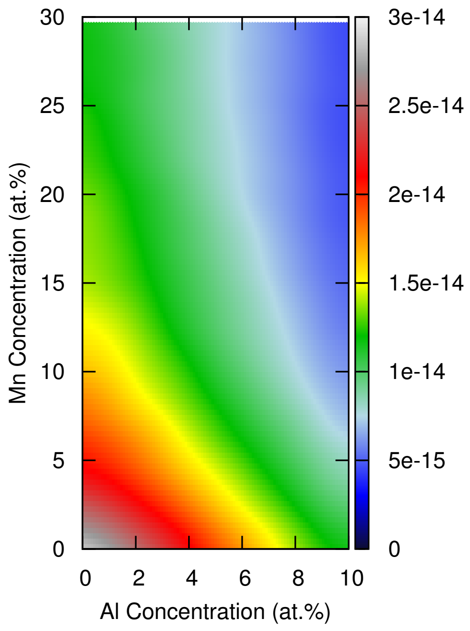


Figure 3.3: Hydrogen diffusion as a function of Al and Mn concentration.

With the increase in Al content, decrease in H diffusivity is observed. If we look at the yellow region of constant H diffusivity, we notice that for the same decrease in H diffusivity Al is more effective than Mn.

In Fig. 3.4, hydrogen diffusivities are plotted as a function of ratio of concentration of Al to the concentration of Al + Mn for four temperatures. From the plots in Fig. 3.4, it can be seen that with the increase in Al concentration, the diffusivities decreases and this decrease becomes more visible with increasing alloy content. According to a recent study by Hütter *et al.* [4], at higher concentrations of Al, there is an effective blocking of paths for hydrogen diffusion which is caused by the high energy barriers. Also, at higher Al concentration, there is a phase transformation from austenite (fcc) to ferrite (bcc) as Al is a ferrite stabilizer which leads to the breakdown of our model as the parameters are no longer valid in this regime.

In Fig. 3.5, hydrogen diffusivity is shown as a function of inverse of temperature for 3 different alloy contents. Black triangles represents the hypothetical pure iron system, red squares represents the system with 70 at.% Fe and 30 at.% Mn which is what we generally find in high Mn steels and violet dots represents system with 70 at.% Fe, 20 at.% Mn and 10 at.% Al. It can be seen that at higher temperature, decrease in hydrogen diffusivity is not much significant when 30 at.% Mn is added to the pure Fe system while this decrease is more visible on adding 20 at.% Mn with 10 at.% Al.

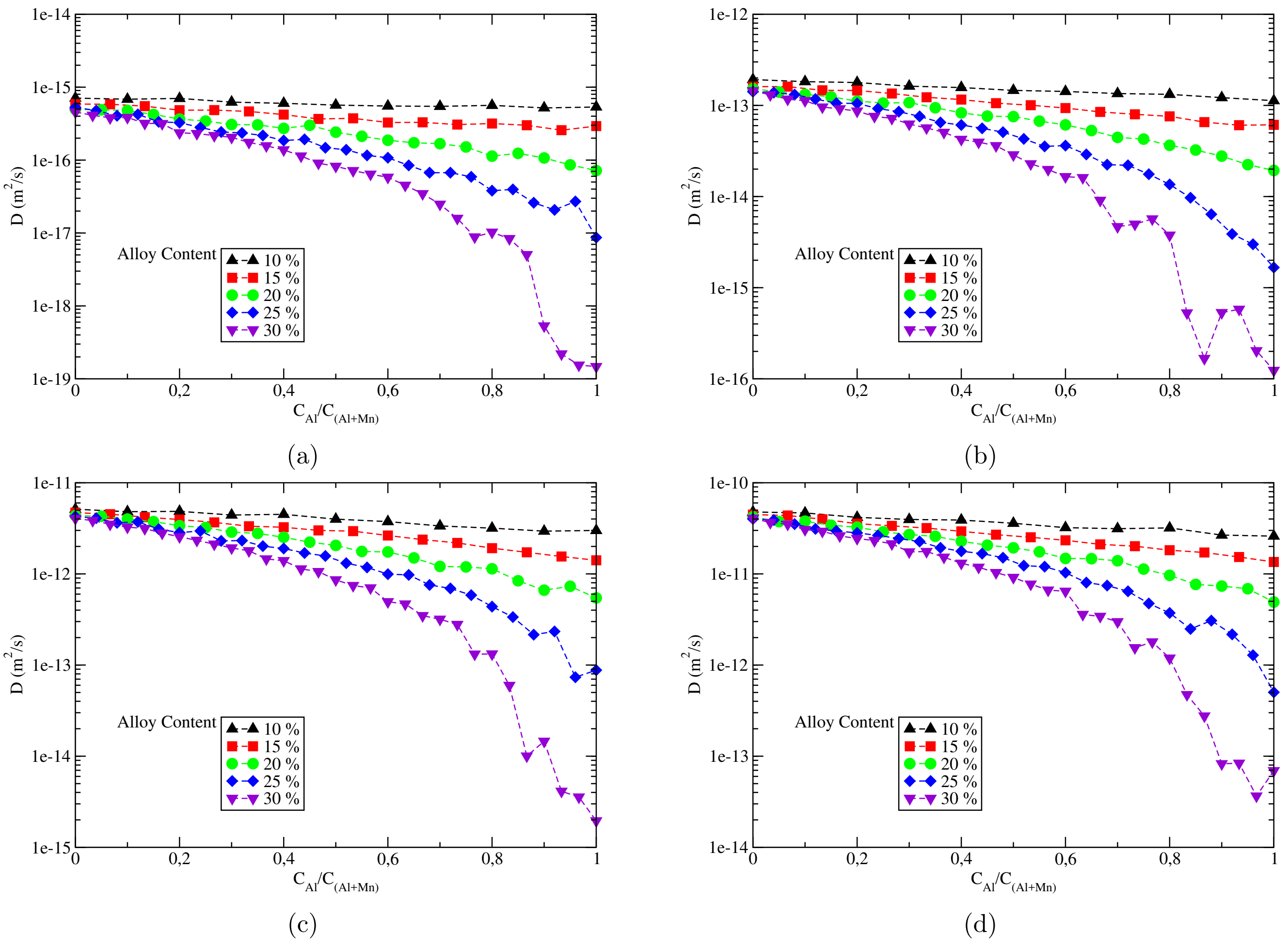


Figure 3.4: Hydrogen diffusivity as a function of the ratio of concentration of Al to the concentration of Al + Mn. Results are shown for (a) 300 K, (b) 400 K, (c) 500 K, and (d) 600 K.

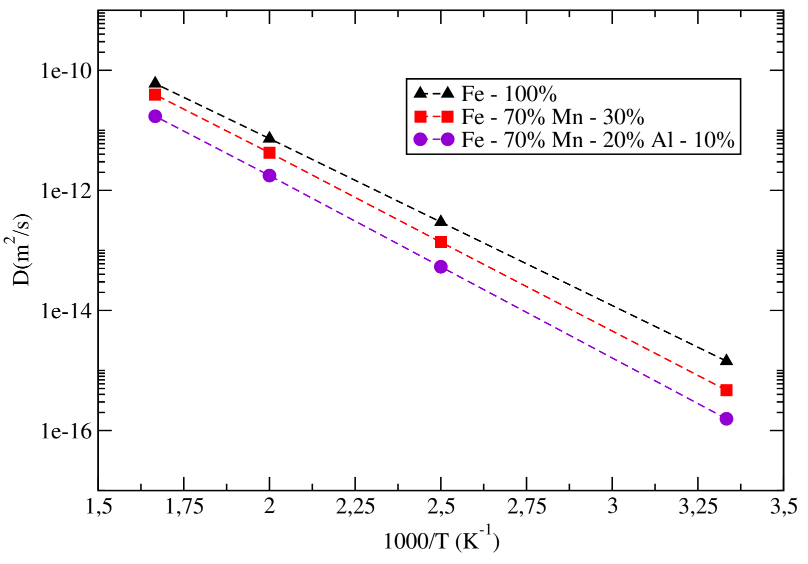


Figure 3.5: Logarithm of the hydrogen diffusivity as a function of the reverse temperature for three different concentrations.

Chapter 4

Conclusions

In this study, we have taken a step further to better understand the effect of alloying Fe-Mn system with Al on the hydrogen diffusion by using kinetic Monte Carlo simulations. Energy barriers for hydrogen diffusion were obtained from DFT calculations. We observe that the H diffusivity decreases with the addition of Mn. This decrease is significant upto 10 at.% of Mn at room temperature. In case of Al, we see the same trend, addition of Al leads to decrease in H diffusivity and is found to be more effective than Mn. At higher Al content, our model breaks down due to the change of structure from fcc to bcc and the parameters are no longer valid to obtain the correct H diffusivity. We also conclude that Al is much more efficient than Mn in decreasing the H diffusivity.

Chapter 5

Appendix

5.1 kMC code

Code is written in python and is divided into two parts. Utilities, where all the parameters and functions are written and the KMC, where the kMC algorithm is implemented. The CPU time for setting up the current model and for 10k kMC steps is about 0.8 seconds and 600 seconds respectively.

5.1.1 utilities

```
import numpy as np
import numpy.random as random

# system parameters
L =10      # number of unit cells
number_of_H = 1000 # Number of hydrogen atoms
max_steps = 10000 # maximum kmc steps per atom.
T = 250      # Temperature(T)
E_H_Fe = 0.0      # Reference Energy (All Fe)
dE_H_FeFe = 0.55     # Activation Energy for jump from all Fe to all Fe Configuration
dE_H_Al = 0.2      # Energy difference between 5Fe1Al and 6Fe configuration
dE_H_Mn = -0.03    # Energy difference between 5Fe1Mn and 6Fe configuration
nu = 1.0E13             # Attempt frequency
kB = 8.617E-5  # Boltzmann Constant




Coordination_NN = {}        # To store the Coordination of first nearest neighbours

E_Conf = {} # To store the Configuration energies
events = {}         # To store the events

LatticeF = []


# To calculate number of Al and Mn atoms

def NeighCount(Coordination):

N_Al = 0
N_Mn = 0
```

```
     for element in Coordination:

       if(element == "Al"):

         N_Al = N_Al + 1
elif(element == "Mn"):

N_Mn = N_Mn + 1

return N_Al,N_Mn


# To calculate the barrier


def Calc_barrier(n, nprime):
        Delta_E = E_Conf[n] - E_Conf[nprime]
        if (Delta_E<0):
            return -Delta_E + dE_H_FeFe  + NeighCount(Coordination_NN[n])[0]*dE_H_Al + \
NeighCount(Coordination_NN[nprime])[0]*dE_H_Al
        else:
            return  dE_H_FeFe + NeighCount(Coordination_NN[n])[0]*dE_H_Al + \
NeighCount(Coordination_NN[nprime])[0]*dE_H_Al


# To calculate the Coordinations, Energies, events, and barriers


def Calc_Coordination(Al_Concen, Mn_Concen):
    TotSites = (4 * L**3)
    LatticeF = np.chararray((TotSites),itemsize = 2)  # 4 atoms per unit cell in the FCC lattice
    Al_number = int((Al_Concen*(TotSites))/100.0)
    Mn_number = int((Mn_Concen*(TotSites))/100.0)
    LatticeF[:] = "Fe"   #  Fe = Iron
    LatticeF[0:Al_number] = "Al"  # Al = Aluminium
    LatticeF[Al_number:Al_number + Mn_number] = "Mn"  # Mn = Manganese
    np.random.shuffle(LatticeF) # random redistribution of Al and Mn

#    print 'Al_concentration', Al_Concen
#    print 'Mn_concentration', Mn_Concen
    print 'System size', 4  * L**3
    print 'Number of Al atoms', Al_number
    print 'Number of Mn atoms', Mn_number

    Lat_site = {}; Lat_Index = {}; Inter_site = {}; Inter_Index = {}

    barrier = {}; event_index = {}      # To store the barriers



    n = 0

    for x in range (L):
      for y in range (L):
        for z in range (L):
          # 4 lattice sites first
          Lat_site[n] = [2 * x, 2 * y, 2 * z]
          Lat_site[n+1] = [2 * x+ 1, 2 * y + 1, 2 * z]
          Lat_site[n+2] = [2 * x+1, 2 * y , 2 * z + 1]
          Lat_site[n+3] = [2 * x, 2 * y +1 , 2 * z + 1]

          Lat_Index[2 * x, 2 * y, 2 * z] = n
          Lat_Index[2 * x + 1, 2 * y+1, 2 * z] = n+1
          Lat_Index[2 * x +1, 2 * y, 2 * z+1] = n+2
          Lat_Index[2 * x, 2 * y+1, 2 * z+1] = n+3

          # Octahedral sites next

          Inter_site[n] = [2 * x+1, 2 * y, 2 * z]
          Inter_site[n+1] = [2 * x, 2 * y + 1, 2 * z]
          Inter_site[n+2] = [2 * x, 2 * y , 2 * z + 1]
          Inter_site[n+3] = [2 * x+1, 2 * y +1 , 2 * z + 1]

          Inter_Index[2 * x+1, 2 * y, 2 * z] = n
          Inter_Index[2 * x, 2 * y+1, 2 * z] = n+1
          Inter_Index[2 * x, 2 * y, 2 * z+1] = n+2
          Inter_Index[2 * x+1, 2 * y+1, 2 * z+1] = n+3

          n += 4

        # Setting up configuration
    for n in range(TotSites):
        x,y,z = Inter_site[n]


        Coordination_NN[n] = [LatticeF[Lat_Index[np.mod(x+1, 2 * L), y, z]],\
 LatticeF[Lat_Index[np.mod (x-1, 2* L), y, z]], LatticeF[Lat_Index [x, np.mod (y+1, 2 * L), z]],
LatticeF[Lat_Index [x, np.mod (y-1, 2 * L), z]], LatticeF[Lat_Index[x, y, np.mod (z+1, 2 * L)]],
LatticeF[Lat_Index[x,y, np.mod (z-1, 2 * L)]]]

# Calculating the Onsite energy

        E_Conf[n] = E_H_Fe + NeighCount(Coordination_NN[n])[0] * dE_H_Al + \
NeighCount(Coordination_NN[n])[1] * dE_H_Mn



    for n in range (TotSites):
        x, y, z = Inter_site[n]
        events[n] = [[1,1,0], [1, -1, 0], [-1, 1, 0], [-1, -1, 0], [1, 0, 1], \
[1, 0, -1], [-1, 0, 1], [-1, 0, -1], [0, 1, 1], [0, 1, -1], [0,-1, 1], [0,-1, -1]]

        barrier_list = []
        event_list = []
        for event in (events[n]):
             x_prime, y_prime, z_prime =  np.mod (x + event[0], 2 * L), \
np.mod (y + event[1], 2 * L), np.mod (z + event[2], 2 * L)

             barrier_list += [Calc_barrier (n, Inter_Index[ x_prime, y_prime, z_prime]) ]
```

```
            event_list += [ Inter_Index [x_prime, y_prime, z_prime]]

        barrier[n] = barrier_list
        event_index[n] = event_list
    return TotSites, Inter_site, events, event_index,  barrier
```

5.1.2 kMC

```
import numpy as np
import numpy.random as random
import copy
import numpy.linalg as lg
import matplotlib.pyplot as plt
from utilities import *


Aluminium_Concentration = np.linspace (0.0,  10.0, 11)    # List of Al Concentration
Manganese_Concentration = np.linspace (0.0,  30.0, 11)    # List of Mn Concentration

print 'List of concentrations for Al', Aluminium_Concentration
print 'List of concentrations for Mn', Manganese_Concentration


f = open ('D_vs_Al_Mn_concentration_250K.dat', 'w')
print >> f, '# Al conc (at.%) Mn conc (at.%)-- D (m^2/s)'
f.close()

for Al_Concen in Aluminium_Concentration:
for Mn_Concen in Manganese_Concentration:


      N_sites, positions,  events, event_list, barrier = Calc_Coordination(Al_Concen, Mn_Concen)
# combine events, event_list and barrier later
      print "############################################"
      print ' Al Concentration(%) = ', Al_Concen
      print ' Mn Concentration(%) = ', Mn_Concen
      print "--------------------------------------------"

      temp = 0.0
      temp_average = 0.0


      for i in range (number_of_H):

  cumulative_Delta_t = 0

  steps = 0

  site = random.randint (N_sites)
  r_0 = copy.copy(positions [site])
  r_real = copy.copy (r_0)

  while (steps<max_steps): # Keeps on simulating till given steps
    R = []
    for j in range(len(events[site])):
       Barriers = barrier[site][j]
       R += [np.exp(-Barriers/(kB*T))]

    R_total = sum(R)   # Sum of all rates


    rho_1 = random.random_sample()
    cum_R = 0.0

    for k in range(len(events[site])):

        cum_R += R[k]

        if ((rho_1 * R_total) < cum_R):
#
            for coord in range (3):
                r_real[coord] += events[site][k][coord]
            site = event_list[site][k]

            break
    rho_2 = random.random_sample ()

    Delta_t = -np.log (rho_2)/ (R_total * nu)
    cumulative_Delta_t += Delta_t           # Increment time
    steps += 1


  D_i = ((r_real[0]-r_0[0])**2 + (r_real[1]-r_0[1])**2 + (r_real[2]-r_0[2])**2)*(2.5e-10)**2 \
/ (2*cumulative_Delta_t*3)     #  Diffusion Constant for each Jump
  print "#",i+1,"Diffusivity(m2/s) = ", D_i

  temp += D_i *cumulative_Delta_t
  temp_average += D_i


      f = open ('D_vs_Al_Mn_concentration_250K.dat', 'a')
      print >> f, Al_Concen, Mn_Concen, temp_average/number_of_H
      print Al_Concen, Mn_Concen, temp_average/number_of_H
      f.close()
```

Bibliography

[1] Motomichi Koyama, Eiji Akiyama, Young-Kook Lee, Dierk Raabe, and Kaneaki Tsuzaki. Overview of hydrogen embrittlement in high-mn steels. *International Journal of Hydrogen Energy*, 42(17):12706 – 12723, 2017.

[2] A.B. Bortz, M.H. Kalos, and J.L. Lebowitz. A new algorithm for monte carlo simulation of ising spin systems. *Journal of Computational Physics*, 17(1):10 – 18, 1975.

[3] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403 – 434, 1976.

[4] C. Hüter S. Dang X. Zhang, A. Glensk and R. Spatschek. Effects of aluminum on hydrogen solubility and diffusion in deformed fe-mn alloys,. *Advances in Materials Science and Engineering*, 2016:9 pages, 2016. Article ID 4287186.

Acknowledgment

I would like to thank Dr.Eunan McEniry and Dr.Tilmann Hickel for their support and giving me the opportunity to work in Computational Materials Design research group at MPIE. This HiWi project gave me an excellent chance to hone my critical thinking and programming skills. Eunan is an excellent supervisor and I appreciate all his contributions of time and ideas for making this experience very productive and stimulating.