

Application of MatCalc in the atomistic / cluster dynamics modeling of cementite nucleation in Fe-Cr-C alloy

JAROSLAV ŽENÍŠEK

Motivation

The precipitates can significantly influence the properties of materials. The nucleation of precipitates is described by CNT. However, for some conditions like for instance highly supersaturated alloys, this concept may give wrong predictions (unrealistically small critical radius of the nucleus). Therefore, we decided to develop an atomistic model for simulation of nucleation and to compare its results with the theoretical predictions for nucleation. The atomistic modeling can give us more detailed information about the system behavior during nucleation and its results can help to improve the theoretical treatment of nucleation.

Basic system description

We deal with simulations of cementite (M_3C) nucleation. Currently, our system consists of three components: Fe, Cr, C. In the simulations, the system is represented as a cubic BCC box whose lattice positions are occupied with substitution elements (Fe, Cr). The interstitial elements are not simulated atomistically, we only assume their presence in the box. We consider that one single cluster can evolve in the centre of the simulation box. The size of the cluster is defined by the number of C atoms in the cluster, n_c . The number of substitution positions in the cluster at any time is $3n_c$ (one of these positions can be eventually vacant).

The evolution of the system is due to diffusion of both substitution and interstitial elements. The growth and the size fluctuation of the cluster are determined by the diffusion of C atoms and their attachments (detachments) to (from) the cluster. Substitution atoms can enter or leave the cluster by the vacancy diffusion mechanism which causes the evolution of the chemical composition of the cluster.

The exchange frequencies between atoms of type i and vacancy are computed from the chemical diffusivities of component i . The attachment rate of the carbon atoms depends on the carbon diffusivity in the matrix; the detachment rate of carbon from the cluster depends on the difference between Gibbs free energies of the system before and after such a detachment. There are different diffusivities in the matrix and in the cluster which also depend on the chemical composition of the matrix and of the cluster.

Use of MatCalc I – calculation of diffusivities and Gibbs energies

Before a particular simulation is executed, we must define the maximum allowed size of the precipitate and then calculate the diffusivities and Gibbs energies for all possible cluster sizes and chemical compositions which may (at least theoretically) occur during simulation. All our simulations represent nucleation at given constant temperature. To summarize, for given temperature and chemical composition of the alloy, the following quantities are calculated:

- 1) chemical diffusivity of Fe in the matrix, $D_{Fe}^M(\vec{n})$
- 2) chemical diffusivity of Cr in the matrix, $D_{Cr}^M(\vec{n})$
- 3) chemical diffusivity of C in the matrix, $D_C^M(\vec{n})$
- 4) Gibbs energy of the system, $G(\vec{n})$

The vector \vec{n} defines the composition of the cluster, its coordinates are the number of vacancies, number of C, number of Fe and number of Cr atoms in the cluster. The diffusivity data for precipitates (clusters) are not available in the database so the diffusivities of Fe, Cr in the cluster, $D_{Fe}^P(\vec{n}), D_{Cr}^P(\vec{n})$, are estimated as $D_{Fe}^P(\vec{n}) = D_{Fe}^M(\vec{n}) / 40$ and $D_{Cr}^P(\vec{n}) = D_{Cr}^M(\vec{n}) / 40$.

All the quantities described in the points (1)-(4) above are calculated with the use of a small program which accesses a MatCalc library. In the program, firstly the functions for the system setting up are called, and then we use two loops. The diffusivities and the Gibbs energies are computed for each system consisting of cluster with nC carbon and nCr chromium atoms and the surrounding matrix, where nC (nCr) are varied in the first (second) loop. It is necessary to note that the calculated Gibbs energies do not account for the interface energy between the cluster and the matrix. The interface energy is a product of the cluster surface and the specific interfacial energy, γ , which is a free parameter in our simulations. The interfacial cluster energy is computed during the simulation and added to the Gibbs energy obtained by MatCalc.

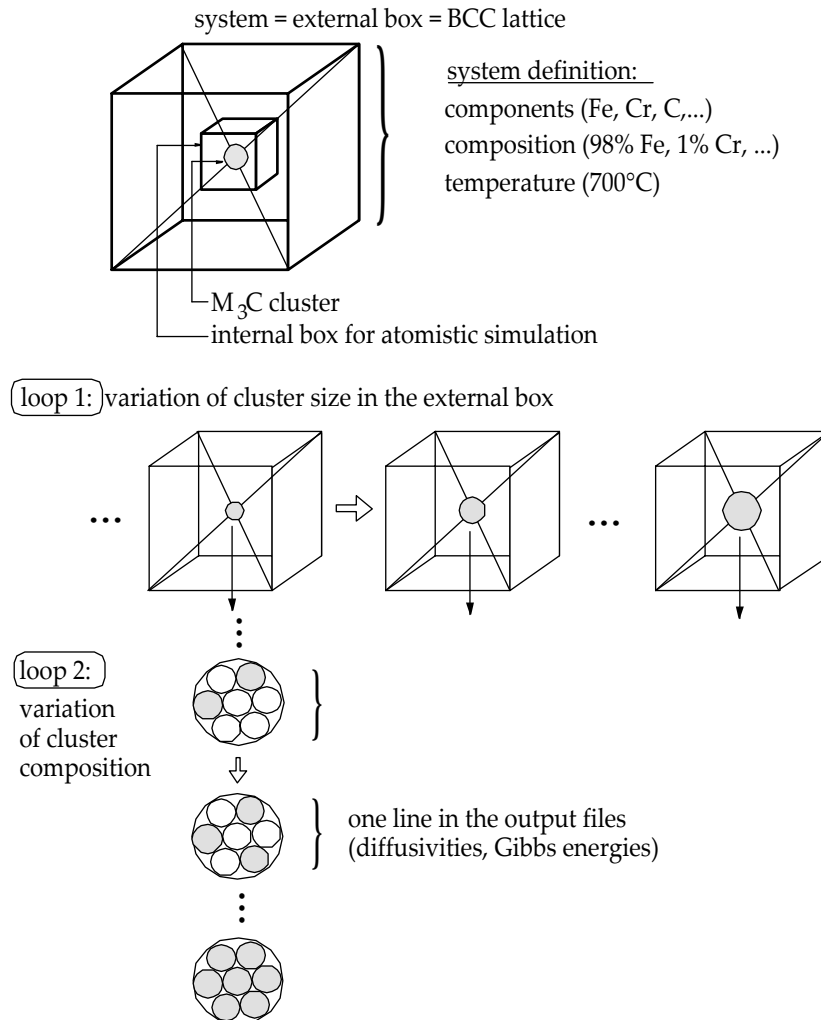


Fig.1: A schematic of the system set up and its variation for the computation of the quantities described in the points (1)-(4) above. During precipitation, the growth and changes in the cluster also depend on the surrounding matrix. The volume of the surrounding matrix should be chosen with respect to the realistic values. On the other hand, it is more convenient to perform the atomistically based simulations of the cluster evolution in a relatively small box

in order to minimize the computer memory requirement. Therefore, the external box is used to calculate the input data for the atomistic simulations which are performed in the smaller (internal) box.

Use of MatCalc II – calculation of the matrix composition after nucleation of precipitates

In the previous simulations, the initial composition of the matrix corresponds to the state in which no cluster exists in the matrix. As soon as first clusters emerge in the matrix, the composition of the matrix may change. In the earlier stages of precipitation, mostly precipitates with para-equilibrium composition may nucleate, and the matrix composition may change in such a way that further nucleation is only possible for precipitates with ortho-equilibrium composition. The most interesting part is the transition region between the nucleation of precipitates with the para- and ortho-equilibrium composition. Therefore, now we want to examine the (further) evolution of clusters in a matrix, the composition of which has been changed due to the nucleation of precipitates with para-equilibrium composition.

Basically, we need to know the chemical composition of the matrix after the nucleation of precipitates with para-equilibrium composition. This is calculated by MatCalc, the details of such calculation can be found in the MatCalc tutorial (<http://www.matcalc.tugraz.at/> > Online help > Examples > Precipitation simulation). The new chemical composition of the matrix is then used as an input parameter for the calculations of diffusivities and Gibbs energies, as described in the previous section.

Program code for the calculation of diffusivities and Gibbs energies

Here, I present my use of MatCalc functions in a small program for calculation of the diffusivities and Gibbs energies. In the following, only the central part of the code is shown.

```
//initialize MatCalc kernel ...
MCC_InitializeKernel(NULL);

//open database
MCC_OpenDatabaseFile(path, DBOC_EQUILIBRIUM, FALSE/*StartAsThread*);

//define elements, phases ...
MCC_SelectDBElement(MCC_GetDBElementIndex("VA") , TRUE);
MCC_SelectDBElement(MCC_GetDBElementIndex("C"), TRUE);
MCC_SelectDBElement(MCC_GetDBElementIndex("FE"), TRUE);
MCC_SelectDBElement(MCC_GetDBElementIndex("CR"), TRUE);

MCC_SelectDBPhase(MCC_GetDBPhaseIndex("BCC_A2"), TRUE);
MCC_SelectDBPhase(MCC_GetDBPhaseIndex("cementite") , TRUE);

//read ...
MCC_ReadDatabaseFile(DBOC_EQUILIBRIUM, TRUE, FALSE);
MCC_SetReferenceElement(MCC_GetElementIndex("FE"));

//read diffusion data
MCC_OpenDatabaseFile(path2, DBOC_DIFFUSION, FALSE);
MCC_ReadDatabaseFile(DBOC_DIFFUSION, TRUE, FALSE);

//some initial equilibrium
MCC_SetTotalAmountOfElement(MCC_GetElementIndex("C"),
0.001, FALSE, SAE_ADJUST_REF_ELEMENT);
```

```

MCC_SetTotalAmountOfElement(MCC_GetElementIndex("CR"),
                             0.1, FALSE, SAE_ADJUST_REF_ELEMENT);
MCC_SetAllStartValues(FALSE);
MCC_CalcEquilibrium(TRUE, EO_QUIET);

//set composition ...

MCC_SetTotalAmountOfElement(MCC_GetElementIndex("C"),
                             (C_TOTAL / (N_S+C_TOTAL-1)), FALSE/, SAE_ADJUST_REF_ELEMENT);

MCC_SetTotalAmountOfElement(MCC_GetElementIndex("CR"),
                             (CR_TOTAL / (N_S+C_TOTAL-1)), FALSE, SAE_ADJUST_REF_ELEMENT);

// the atom fractions of C and Cr
// N_S is the number of all substitution lattice positions in the external box
// -1 is for one vacancy in the external box

MCC_SetTemperature((TEMP+273.16) /*in K*/, TRUE);

MC_HANDLE ph_handle_cem = MCC_GetPhaseHandleByName("cementite");
MC_HANDLE ph_handle_bcc = MCC_GetPhaseHandleByName("bcc_a2");
MCC_SetPhaseFlags(ph_handle_cem, FLAG_PHASEISFIXED, TRUE); //fix phase amount

int element_index = MCC_GetElementIndex("Cr");

//loop for the variation of the cluster size

for (n_C_cem=1; n_C_cem<=CEM_C_MAX; n_C_cem++)
{
    if(VAC_IN_CEM==0){
        atoms_in_cem=(4*n_C_cem);
        Cr_cemMAX=(3*n_C_cem);
    }
    else{
        atoms_in_cem=(4*n_C_cem-1);
        Cr_cemMAX=(3*n_C_cem-1);
    }

    // different max values of cluster-constituting atoms for clusters containing zero (first case) or one
    // vacancy given by value of parameter VAC_IN_CEM which is read from the input file

    n_C_matrix=(C_TOTAL-n_C_cem);
    F_cem=(atoms_in_cem / (N_S+n_C_cem+n_C_matrix-1));
    MCC_SetPhaseAmount(ph_handle_cem, F_cem);

    //loop for the variation of the chemical composition of the cluster

    for (n_Cr_cem=0; n_Cr_cem<=(Cr_cemMAX); n_Cr_cem++)
    {
        if(VAC_IN_CEM==0)
            x_Cr_cem=(n_Cr_cem / (4*n_C_cem));
        else
            x_Cr_cem=0.75*(n_Cr_cem/(3*n_C_cem-1));

        // we cannot set the x_Cr_cem value to exact zero or one because
        // it is used in some functions as the argument of a logarithm

        if (x_Cr_cem <= 0.0)
            x_Cr_cem = 1e-10;
        if (x_Cr_cem >= 1.0)

```

```

        x_Cr_cem = 1.0-1e-10;

//set composition in terms of u-fraction
MCC_SetElementStatusInPhase(ph_handle_cem, element_index, FET_FIXED_MF,
                             x_Cr_cem);

// calc equilibrium ...
MCC_CalcEquilibrium(TRUE, EO_QUIET);

// extraction of Gibbs energy ...
Gibbs=MCC_GetMolarGibbsEnergy();

// recalculation of the Gibbs energy according to the number of all atoms in the
// external box using Avogadro number N_AVOG
Gibbs_reduced=Gibbs*( (N_S+C_TOTAL-1) / N_AVOG);

// extraction of diffusivities...
MCC_GetCalcVariable("D$bcc_a2$c", &DC_chem, TRUE, FALSE);
MCC_GetCalcVariable("D$bcc_a2$Cr", &DCr_chem, TRUE, FALSE);
MCC_GetCalcVariable("D$bcc_a2$Fe", &DFe_chem, TRUE, FALSE);

// chemical diffusion coefficient of C in matrix
// chemical diffusion coefficient of Cr in matrix
// chemical diffusion coefficient of Fe in matrix

char buffer[100];

// example of writing one output file with Gibbs energies, similarly for diffusivities

ofstream myfile (name, ios::out | ios::app);

if(myfile.is_open())
{
    myfile << rank << "\t";
    myfile << n_C_cem << "\t";
    myfile << n_Cr_cem << "\t";
    sprintf(buffer, "%5.10e", (Gibbs_reduced-Gibbs_BCC));
    myfile << buffer << "\t";
    myfile << endl;
    if(n_Cr_cem==Cr_cemMAX)
        myfile << endl;
    myfile.close();
}

rank++;
}
}

```

Examples of output files

Values in the following examples are calculated for a box containing 221184000 substitutional lattice positions with composition Fe-1at%Cr-0.1at%C at 660°C. The meaning of the columns is the following (from left): rank, number of carbon atoms in cluster, number of chromium atoms in cluster, desired quantity, which is the chemical diffusion coefficient of carbon in the matrix in the first example. The blocks with constant number of carbon atoms are separated by one empty line.

0	1	0	2,7570805295e-011
1	1	1	2,7570805777e-011
2	1	2	2,7570806259e-011
3	1	3	2,7570806741e-011
4	2	0	2,7570804653e-011
5	2	1	2,7570805135e-011
6	2	2	2,7570805617e-011
7	2	3	2,7570806099e-011
8	2	4	2,7570806581e-011
9	2	5	2,7570807063e-011
10	2	6	2,7570807545e-011
11	3	0	2,7570804011e-011
12	3	1	2,7570804493e-011
13	3	2	2,7570804975e-011
14	3	3	2,7570805457e-011
15	3	4	2,7570805939e-011
16	3	5	2,7570806421e-011
17	3	6	2,7570806903e-011
18	3	7	2,7570807385e-011
19	3	8	2,7570807867e-011
20	3	9	2,7570808348e-011

Example 1: Chemical diffusion of carbon (fourth column) as a function of cluster size given by number of carbon atoms (second column) and chemical composition given by number of chromium atoms (third column).

0	1	0	8,1360997485e-019
1	1	1	8,1360992722e-019
2	1	2	8,1360987959e-019
3	2	0	8,1360997674e-019
4	2	1	8,1360993229e-019
5	2	2	8,1360988784e-019
6	2	3	8,1360984339e-019
7	2	4	8,1360979894e-019
8	2	5	8,1360975449e-019
9	3	0	8,1360997864e-019
10	3	1	8,1360993498e-019
11	3	2	8,1360989132e-019
12	3	3	8,1360984767e-019
13	3	4	8,1360980401e-019
14	3	5	8,1360976035e-019
15	3	6	8,1360971669e-019
16	3	7	8,1360967304e-019
17	3	8	8,1360962938e-019

Example 2: Chemical diffusion of chromium (fourth column) as a function of cluster size given by number of carbon atoms (second column) and chemical composition given by number of chromium atoms (third column). In this case, the cluster contains one vacancy.

The second example shows the chemical diffusion coefficient of chromium in the fourth column. One can see that the each block (between two empty lines) has one line less than in the previous example. This is due to one vacant lattice site in the cluster which reduces the

number of chromium atoms which can be accommodated by the cluster of given size. All diffusion coefficients and Gibbs energies calculated for both situations: matrix+cluster without vacancy and matrix +cluster with vacancy. This example also shows that the diffusivity of chromium as a substitution element is some orders of magnitude lower than the diffusivity of interstitial carbon atoms (example 1).

Simulations of nucleation

We use the computed quantities to simulate the nucleation of cementite particles. In this part, we present two examples of cluster evolution at 660°C and $\gamma = 0.08 \text{ Jm}^{-2}$ in an alloy with composition Fe-1at%Cr-0.1at%C.

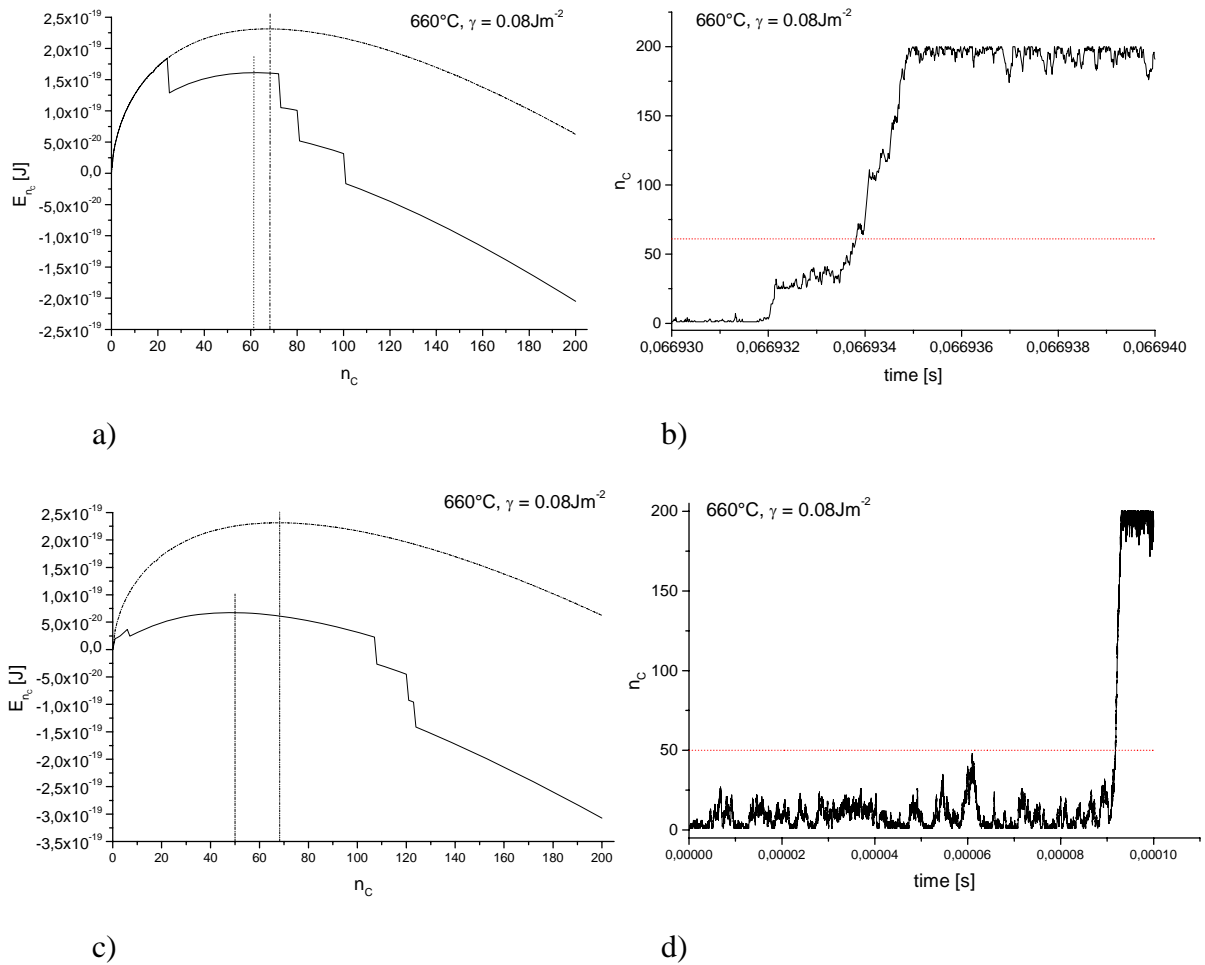


Fig. 2: Each row belongs to one simulation. The total energy profiles are shown on the left-hand side and the evolutions of the cluster sizes are given on the right-hand side. The size is expressed by the number of carbon atoms in the cluster, n_c . In the plots b) and d), the horizontal dotted lines mark the critical cluster sizes. The dash-dotted (smooth) curves in the plots a) and c) are the total cluster energy profiles for Cr-free vicinities of the cluster centers. The full lines in the plots a) and c) are the total cluster energy profiles corresponding to the presence of some Cr atoms in the vicinities of the cluster centres. The dotted and dash-dot-dotted vertical lines in the plots a) and c) denote the critical cluster sizes for the Cr-free and for the actual chemical compositions of the vicinities of the nucleation centers, respectively. In these simulations, the maximum cluster size was set as $n_c^{\max} = 200$.

Let us assume that a vicinity of the cluster center is a sphere and that the size of the sphere is determined by the number of carbon atoms in it, n_c . The types and numbers of substitution atoms in a sphere of size n_c are always exactly known, so one can calculate the corresponding energy required to build such a spheric cluster. Moreover, one can plot an energy-profile for clusters of sizes ranging from zero to given maximum size n_c^{\max} . This is done in the plots a) and c) of figure 2. The actual profiles marked with full line correspond to different distributions of Cr atoms in the vicinity of the cluster center. The hops in the curves indicate the positions of Cr atoms. Additionally, energy profiles for Cr-free vicinity of the cluster center are marked by dash-dot line (smooth curves without hops). The vertical lines indicate the critical cluster sizes corresponding to the maximum in the total energy profiles.

In the plots b) and c) of figure 2, the nucleations for two different distributions of chromium atoms in the vicinities of the cluster centers are shown. The critical cluster sizes are marked by the horizontal lines. One can clearly observe fluctuations of the cluster size up to the moment at which the critical size is overcome. Then, the clusters start to grow. An important output of these simulations is the nucleation time, i.e. the time necessary to overcome the critical size. In the first case shown in the plots a) and b), it is about 0.07 s but in the second case it is about 1×10^{-4} s, which is more than two orders of magnitude shorter time than in the first example. Therefore, it turns out that the nucleation can be strongly affected by the chemical composition in the vicinities of the nucleation centers.