Kinetic Precipitation Model

User's Guide and Reference Manual

Alp H. Gencer

Version: 97.221 (August 09, 1997)

Contents

Chapter 1

Theory

Nucleation and growth processes play a critical role in a large range of materials processing systems. Classical modeling approaches divide such processes into two discrete steps, with nucleation and growth being modeled using fundamentally different assumptions, each valid only under idealized conditions. Thus, although these approaches are very useful for understanding qualitative behavior, they are unsuitable in many cases for the development of quantitative models, particularly under complex annealing conditions (e.g., multi-step anneals). Noting the power of modern computers to solve complex systems of coupled differential equations, we have developed a unified approach to modeling of nucleation and growth processes which extends nucleation theory to include the behavior of supercritical as well as sub-critical aggregates.

This manual describes briefly the theory that's underlying the Kinetic Precipitation Model (KPM) and the specific implementation. User's that are only interested in using the model can confine themselves to Chapter 2, although a short look at this chapter should help to understand the parameters.

1.1 Full Kinetic Precipitation Model

The major challenge in modeling the evolution of precipitates and extended defects is the fact that different sized defects have very different properties. The Full Kinetic Precipitation Model (FKPM) [3] treats precipitates of different sizes as independent species (f_n) and accounts for their kinetics by considering the attachment and emission of solute atoms.

The driving force for precipitation is the minimization of the free energy of the system, where the free energy of a size n extended defect is given by:

$$\Delta G_n = -nkT \ln \frac{C_A}{C^{\mathrm{ss}}} + \Delta G_n^{\mathrm{exc}} \tag{1.1}$$

Here, C^{ss} is the solid solubility and $\Delta G_n^{\mathrm{exc}}$ is the excess surface and strain energy of a size n extended defect.

The main reaction in the system is the attachment and emission of solute atoms to and from precipitates. If I_n denotes the net growth rate from size n to $n+1$ we may write the following equation:

$$I_n = \begin{cases} D\lambda_n \left(C_A f_n - C_n^* f_{n+1}\right) & \text{for } n \geq 2 \\ D\lambda_1 \left(C_A^2 - C_1^* f_2\right) & \text{for } n = 1 \end{cases} \tag{1.2}$$

Note that I_1 is different from other terms, because it represents the rate for formation of the defects by reaction of two interstitials.

The growth rate of precipitates is written in the form $D\lambda_n$, where λ_n incorporates effects of both diffusion to the precipitate/silicon interface and the reaction at the interface. λ_n is calculated based on solving the steady-state diffusion equation in the neighborhood of a precipitate, taking its shape into account. C_n^* represents the interstitial concentration at which a size n precipitate would gain no energy by growing from size n to size $n+1$:

$$C_n^* = C^{\text{ss}} \exp\left(\frac{\Delta G_{n+1}^{\text{exc}} - \Delta G_n^{\text{exc}}}{kT}\right) \tag{1.3}$$

The evolution of the size distribution f_n is given by the difference between the net rate at which defects grow from size $n-1$ to n (I_{n-1}) and the net rate of growth from size n to $n+1$ (I_n). Since the fundamental growth process is the incorporation of a solute atom, the total change in C_A includes a term from each growth reaction, giving a sum over I_n:

$$\frac{\partial f_n}{\partial t} = I_{n-1} - I_n \tag{1.4}$$

$$\frac{\partial C_A}{\partial t} = -2I_1 - \sum_{n=2}^{\infty} I_n \tag{1.5}$$

1.2 Reduced Kinetic Precipitation Model

The Full Kinetic Precipitation Model adds an extra dimension, namely precipitate size, to the problem being solved. To minimize the computational budget, Clejan and Dunham [1] have developed a more efficient version of this model, based on the same principles. Instead of calculating all the f_n, one needs to calculate only the lowest three *moments* of the distribution ($m_i = \sum_{n=2}^{\infty} n^i f_n$, where $i = 0, 1, 2$) and make a closure assumption. In particular, the closure assumption used is that the distribution is the one that minimizes the free energy, given the moments. The resulting system, known as the Reduced Kinetic Precipitation Model (RKPM), has the following set of continuity equations:

$$\begin{aligned} \frac{\partial m_0}{\partial t} &= I_1 \\ \frac{\partial m_1}{\partial t} &= 2I_1 + \sum_{n=2}^{\infty} I_n \\ \frac{\partial m_2}{\partial t} &= 4I_1 + \sum_{n=2}^{\infty} I_n(2n+1) \\ \frac{\partial C_A}{\partial t} &= D\nabla^2 C_A - 2I_1 - \sum_{n=2}^{\infty} I_n \end{aligned} \tag{1.6}$$

Note that the sums over the I_n can all be written in terms of sums over f_n and nf_n, and hence can be calculated from the three moments.[1] However, these calculations require the solution of a non-linear equation system at every time step and each grid point. To make the simulation computationally efficient, the sums are pre-tabulated for a range of m_i values and interpolation from these values is used during the simulation. The resulting system is thus:

$$
\begin{aligned}
\frac{\partial m_0}{\partial t} &= D\left[\lambda_1 C_A^2 - m_0 C^{\mathrm{ss}}\gamma_0\right] \\
\frac{\partial m_1}{\partial t} &= D\left[2\lambda_1 C_A^2 + m_0 C_A \gamma_1 - m_0 C^{\mathrm{ss}}\gamma_2\right] \\
\frac{\partial m_2}{\partial t} &= D\left[4\lambda_1 C_A^2 + m_0 C_A \gamma_3 - m_0 C^{\mathrm{ss}}\gamma_4\right] \\
\frac{\partial C_A}{\partial t} &= -\frac{\partial m_1}{\partial t}
\end{aligned}
\tag{1.7}
$$

with

$$
\begin{aligned}
\gamma_0 &= \lambda_1 \hat{C}_1^* \hat{f}_2 \\
\gamma_1 &= \sum_{n=2}^{\infty} \lambda_n \hat{f}_n \\
\gamma_2 &= \lambda_1 \hat{C}_1^* \hat{f}_2 + \sum_{n=2}^{\infty} \lambda_{n-1} \hat{C}_{n-1}^* \hat{f}_n \\
\gamma_3 &= \sum_{n=2}^{\infty} (2n+1)\lambda_n \hat{f}_n \\
\gamma_4 &= \lambda_1 \hat{C}_1^* \hat{f}_2 + \sum_{n=2}^{\infty} (2n-1)\lambda_{n-1} \hat{C}_{n-1}^* \hat{f}_n
\end{aligned}
\tag{1.8}
$$

1.3 Algorithm of the Table Generator

At the heart of the implementation of the Reduced Kinetic Precipitation Model lies the look-up table, that is used to get the values of γ_i during the simulation. Since the γ_i and the moments are related by means of strongly non-linear sets of functions, a table look-up and interpolation method is used to calculate γ_i.

The distribution of the defects over sizes is given by the function derived from the closure assumption:

$$f_n = K \exp\left(-\frac{\Delta G_n^{\mathrm{exc}}}{kT} + z_1 n + z_2 n^2\right) \tag{1.9}$$

with K, z_1 and z_2 are chosen given the moments. Obviously, K is just a normalization constant, hence we can normalize the distribution by dividing by m_0. We denote normalized values with $\hat{f}_n$:

$$\sum_{n=2}^{\infty} \hat{f}_n = 1$$

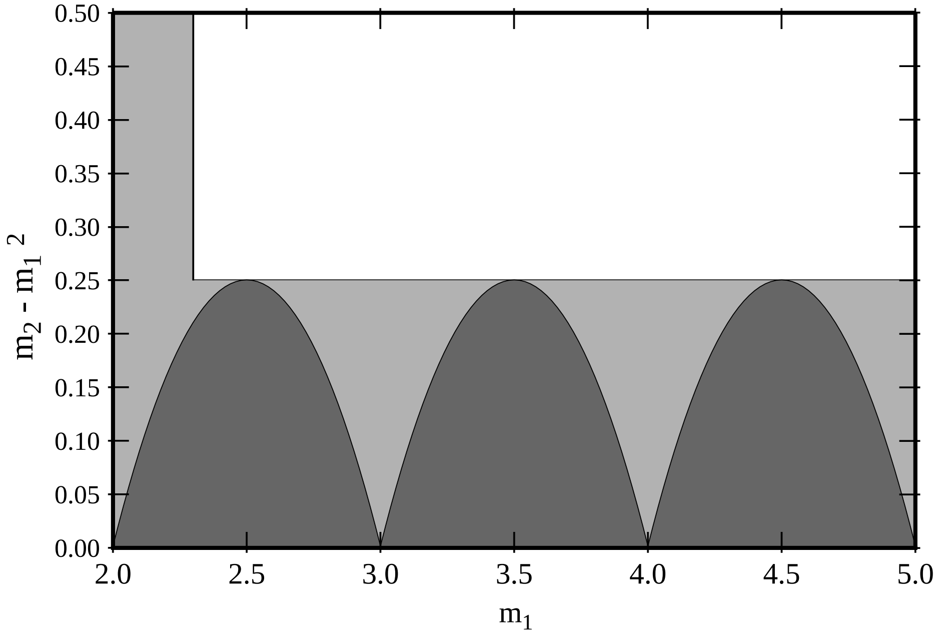

Figure 1.1: Black regions denote the physically impossible values of $\hat{m}_2$ and grey regions denote portions of the $\hat{m}_1$,$\hat{m}_2$ plane where "narrow" is called. The pattern repeats itself continuously to the right.

$$\begin{aligned} \sum_{n=2}^{\infty} n\hat{f}_n &= \hat{m}_1 \\ \sum_{n=2}^{\infty} n^2\hat{f}_n &= \hat{m}_2 \end{aligned} \tag{1.10}$$

The shape of the distribution, which is determined by z_1 and z_2 depends only on $\hat{m}_1$ and $\hat{m}_2$. To generate the look-up table, we find z_1,z_2 values corresponding to various $\hat{m}_1$,$\hat{m}_2$ values, and calculate γ_i from that.

1.3.1 Layout of the grid

There are, however, some issues related to the limits of the table and the grid spacing. The first obvious limit is that $\hat{m}_1 >= 2$. Also, for any statistical distribution $\hat{m}_2 >= \hat{m}_1^2$.

There is, however, one more interesting point due to the discreteness of the precipitate sizes: The relationship $\hat{m}_2 = \hat{m}_1^2$ for any distribution is true only if there is a single size present, namely $\hat{m}_1$. What if $\hat{m}_1$ is non-integer? Then, obviously the above equality cannot hold, since there will be at least two distinct sizes present: the two integer numbers that bracket $\hat{m}_1$. So, if we plot the minimum $\hat{m}_2 - \hat{m}_1^2$ that is theoretically possible versus the size, we get something like Fig. 1.1.

Thus, we use a minimum value of $\hat{m}_2 - \hat{m}_1^2$ equal to 0.25 for the look-up table, and use a routine called `narrow` for $\hat{m}_2 - \hat{m}_1^2$ values less than 0.25. The routine `narrow` assumes no specific distribution function as given in (1.9), but assumes that precipitates are limited to only three sizes: the two bracketing $\hat{m}_1$, and one smaller or larger than these (whichever is closer to $\hat{m}_1$). Then, it solves three equations (1.11) for the three unknown f_n and calculates γ_i from that.

If the average precipitate size is less than some number, it is assumed that only precipitates of size 2, 3 and 4 are present, and again narrow is used.

The remaining portion of $\hat{m}_1$,$\hat{m}_2$ plane is grided and a look-up table is generated. To make the griding equally spaced, and to correctly set the origin of the table, the following transformed variables are used as the base of the grid:

$$\tilde{m}_1 = \log(\hat{m}_1 - (m_1^{\min} - 1)) \tag{1.11}$$

$$\tilde{m}_2 = \log\left(\frac{\hat{m}_2 - m_2^{\min}}{\hat{m}_1^2}\right) \tag{1.12}$$

This ensures that the lower left-hand corner of the grid is at (0,0). The grid is equally spaced with the fineness the user specifies.

1.3.2 Filling the table

The table is "adaptive", i.e. a grid point is generated only when it is needed. The lookup routine calculates first $\hat{m}_1$,$\hat{m}_2$ values and determines from that whether a table interpolation is needed or narrow routine will be called. The interpolation routine calculates $\tilde{m}_1$,$\tilde{m}_2$ values and finds into which box they fall. Then, the four grid points surrounding that box are requested from the table. If they had been previously generated, the γ_i values are returned immediately, otherwise the table generator tries to generate these points.

Since $\tilde{m}_1$,$\tilde{m}_2$ and z_1,z_2 are related through highly non-linear functions, a Newton-Raphson multi-dimensional root finder is used to find the z_1,z_2 values corresponding the requested point. Since Newton-Raphson may become unstable and not converge, we need a close starting point, like the next grid point to the one requested. So, z_1,z_2 values for that point are asked, and if that point hasn't been previously generated, first that point is generated. This search process continues until a generated point is found. The table always starts with one point, the lower left-hand corner point, which is filled during initialization.

Upon exit, the generated portion of the table is written to disk, with the user-specified file name. So, successive calls to a table with the same parameters causes the table to be read from the disk, expanded if necessary, and then to be rewritten to disk.

1.3.3 Re-discretization

The sums that appear in the expressions for γ_i (Eq. 1.8) can easily be calculated by direct summation over the range of sizes of interest. However, for systems where the sizes can get very large, such as dislocation loops in silicon, doing the sums from, say from size 2 to 1,000,000 becomes very cumbersome. We may gain in speed if we realize that for larger sizes the distribution will not vary considerably from one size to the neighboring one. Thus, we will only need some representative value of f_n for a range of n values. We assume that the distribution is continuous for large sizes and then re-discretize it.

The n_i values that we will consider in the sum are found by the following procedure: We start with $n_0 = 2$ and $dn_0 = 1$, and find $n_{i+1} = n_i + dn_i$, where we multiply dn_i by a constant factor α after each step, and go up to a user-specified maximum size $n_{\max}$. We make sure that n_i is always an integer. Then, each n_i gets a weight in the sums depending on the range it represents, $w_i = \frac{1}{2}(n_{i+1} - n_{i-1})$. The sums then are performed with this weight.

Chapter 2

Using KPM

2.1 KPM commands

2.1.1 The `defect` command

The `defect` command creates a new defect/precipitate and returns it. It's syntax is as follows:

```
defect {energy script} {lambda script} |maxsize| |step|
```

The `{energy script}` is a script returning the excess energy function. Any valid Tcl script can be used. When the script is invoked the Tcl variable n is set to the current precipitate size. The script should return the excess energy ΔG_n^{exc}, divided by kT as a function of the precipitate size.

`|maxsize|` and `|step|` show rediscretization parameters (see section 1.3.3), which may be omitted. If they are omitted, `|maxsize|` defaults to 1,000,000 and `|step|` defaults to 1.1.

The `{lambda script}` is a script returning the kinetic forward rate, again as a function of the precipitate size, n. For customary defect sizes, Tcl procedures have been defined. All procedures are called with a single argument, the precipitate size, and return λ_n as determined by the geometry. Some preset Tcl variables can be used to determine the characteristics of the functions. In particular, `KPM_capture` should be set to the capture radius of the precipitate (defaults to 2.71, the value for silicon) and `KPM_lambda` should be set to the interface reaction distance (D/k)at the precipitate/matrix interface. The default value is 2.71. Here is a list of procedures for calculating lambda:

- `KPM_Spherical`: For spherical precipitates. `KPM_rho` is used to determine the density of solute atoms in the precipitate (defaults to $2.5 \times 10^{22}\text{cm}^{-3}$)

- `KPM_Loop`: For dislocation loops (disc shaped). `KPM_loop_sigma` is used to determine the areal density of interstitial atoms in the dislocation loop (defaults to $1.57 \times 10^{15}\text{cm}^{-2}$)

- `KPM_311`: For $\{311\}$ defects (elongated planar defects). `KPM_311_sigma` is used to determine the areal density of interstitial atoms in $\{311\}$ defects (defaults to 10^{14}cm^{-2})

and `KPM_311_wmax` is used to determine the maximum width {311} defects can attain (defaults to 90)

2.1.2 The `table` command

The `table` command creates a new RKPM lookup table and returns it. Thus, the `table` command is only necessary when RKPM (and not FKPM) is used. It's syntax is as follows:

```
table "defect" "filename" |base1| |base2|
```

Here, `"defect"` is the output of a `defect` command and `"filename"` is the name of the file the calculated table will be written to. Try to use a consistent naming scheme, so that tables can be reused. `|base1|` and `|base2|` are the grid spacings of the table in the $\tilde{m}_1$ and $\tilde{m}_2$ directions, respectively. If omitted, they both default to 0.05.

2.1.3 Support commands

The `moments` command: This command is used (in DOPDEES) to initialize a moment variable. The syntax is:

```
moments |num| |m1hat| |m0|
```

Here, `|num|` indicates the number of moments to be initialized (3 for RKPM), `|m1hat|` is the average size, and `|m0|` is the zeroth moment. This commands returns a list of values (of length `|num|`) where each element has the value $m_0 \hat{m}_1^i$ with $0 \leq i <$ `|num|`.

The `KPM-init` command: This command is used to reinitialize the KPM library.

2.1.4 Using UNITS

Included with KPM distribution is a package called UNITS, which has simple commands for resolving unit discrepancies. Please read the documentation for this package, which can be found in the directory `KPM/UNITS`.

IMPORTANT: One thing to be careful is that if the user changes the unit system, the KPM library has to be re-initialized using the `KPM-init` command.

2.2 KPM Operators and Functions

2.2.1 RKPM operators in DOPDEES

Within DOPDEES, there are 3 operators that call the RKPM routines. Each of the 3 routines are used for different systems:

- `op rkpm-dop #CA# #M# <CA> <DA> |Css| "table"`

  `rkpm-dop` is the basic RKPM operator and is used for systems as described in Chapter 1. `#CA#` is the name of the solute atom, and `<CA>` is a function returning its active value. #M# is a vectorial field (with 3 components) denoting the 3 moments. <DA> and |Css| are the diffusivity and the solid solubility of the solute, respectively. The last argument, "table" is the output of a table command.

- op rkpm-iv #CI# #CV# #M# <CI> <CV> <DI> <DV> |Css| "table"

  rkpm-iv is a modification of the rkpm-dop operator for systems where the precipitate size can grow by incorporating one atom or shrink by incorporating another one. Extended defects in silicon, which can grow by incorporating interstitials or shrink by incorporating vacancies, are an example. The equations governing this system are slightly different.

- op rkpm-bic #CB# #M# <CB> <DB> <theta> |Css| "table"

  rkpm-bic is for systems where two atoms form a precipitate, as in boron interstitial clusters ($\mathrm{B}_n\mathrm{I}_m$) or arsenic vacancy clusters ($\mathrm{As}_n\mathrm{I}_m$). Kinetics and energetics of such systems are slightly different.

Specifying any RKPM operator once updates the residuals of all components. More than one RPM operator may be specified, with different parameters and fields, so that two processes may be going on at the same time (e.g., interstitial agglomeration and dopant precipitation).

In addition, there is a function that is written for systems where the rkpm-bic operator will be used:

- func kpm-theta <CI0> |CI0*| |kTalpha| |gamma|

  This function returns the <theta> parameter needed for the rkpm-bic operator. <CI0> is a function returning concentration of neutral interstitials, |CI0*| is their equilibrium number, |kTalpha| is kT/α in the model. The function returns:

$$\left(\frac{C_{\mathrm{I}^0}}{C^*_{\mathrm{I}^0}}\right)^{-\gamma}\exp\left[-\frac{kT}{4\alpha}\ln^2\left(\frac{C_{\mathrm{I}^0}}{C^*_{\mathrm{I}^0}}\right)\right]$$

2.2.2 RKPM functions in Alamode

Due to structural differences between Alamode and DOPDEES, RKPM has been implemented in Alamode as functions, rather than operators. They have same names and similar syntax as their DOPDEES counterparts. Since Alamode doesn't support field names that are vectors, all three components of the moment vector must be specified explicitly.

Since the function can return a single value, a last argument comp has been added to each function. The function will return only the residual of the component indicated by comp, which then can be added to comp's equation using rhsFunction. The moments have component numbers equal to their indices, and the last component(s) is(are) the solute concentration(s). It should be obvious that the function has to be used 4(5) times — once for each component — to obtain a complete set.

- `rkpm-dop`:

  Parameters: M_0 M_1 M_2 C_A D_A C_{ss} `table comp`

  C_A D_A are fields or functions.

- `rkpm-iv`:

  Parameters: M_0 M_1 M_2 C_I C_V D_I D_V C_{ss} `table comp`

  C_I C_V D_I D_V are fields or functions.

- `rkpm-bic`:

  Parameters: M_0 M_1 M_2 C_B D_B Θ C_{ss} `table comp`

  C_B D_B Θ are fields or functions.

For all functions following arguments are common:

- M_0 M_1 M_2 are fields indicating the three moments.
- C_{ss} is a variable holding the solid solubility.
- `table` is the output of a `table` command.
- `comp` is a number indicating the component number.

Again, there is the `kpm-theta` function, which returns the Θ argument for `rkpm-bic`:

- `kpm-theta`:

  Parameters: C_{I^0} $C^*_{I^0}$ $kT\alpha$ γ

  C_{I^0} is a field or function.

  $C^*_{I^0}$ $kT\alpha$ γ are variables.

2.2.3 FKPM operator in DOPDEES

DOPDEES also provides a `fkpm` operator as the implementation of Full Kinetic Precipitation Model. FKPM uses many fields at every grid point and is thus much slower than RKPM, but it is much more stable and doesn't have the closure assumption of RKPM. The syntax is:

- op fkpm #F# #M# <D> |Css| defect

  Here, #F# is a field with as many components as in the model, #M# is a field having two components for handling large sizes, <D> is the diffusivity of the solute atom, |Css| is the solid solubility and `defect` is the output of a `defect` command (when using `fkpm` there is no need for a `table` command).

Bibliography

[1] I. Clejan and S.T. Dunham. "A reduced moment based model for precipitaion kinetics and application to dopant deactivation in silicon". *J. Appl. Phys.* **78**(12), 7327 (1995).

[2] I. Clejan and S.T. Dunham. "Interactions of point defects with boron and arsenic precipitates". In **Process Physics and Modeling in Semiconductor Technology**, Electrochemical Soc., 398 (1996).

[3] S.T. Dunham. "Modeling of the kinetics of dopant precipitation in silicon". *J. Electrochem. Soc.* **142**, 2823 (1995).

[4] A.H. Gencer and S.T. Dunham. "A predictive model for transient enhanced diffusion based on evolution of {311} defects". *J. Appl. Phys.* **81**(2), 631 (1997).