

Chapter II. Finite element model

Introduction

The finite element method (FEM) is a very useful numerical tool that allows one to effectively integrate differential equations. The physical problems that can be solved by FEM can in general case be discrete or continuum in nature [Com94]. As far as diffusion is concerned, solving continuum problems is relevant in the present work. Consequently, the general definition for the finite element problem is given in this chapter. As diffusion along and across GBs has its own specific features, the integrals that were evaluated to find the physical property at each node of the finite element mesh are also given. Different numerical problems related to FEM and accompanying simulation of GB diffusion by FEM are explained along with discussion of the main results in coming chapters.

Particularly, FEM was chosen as effective tool for numerical integration in such problems that are characterized by complicated geometries. One can find a broad literature on the application of FEM to different mechanical problems. There even exist several serious books on using FEM to integrate conduction type equations [Com94], [Lew96], [Hun02] as

for heat conduction. Moreover, FEM was recently used to describe atoms and diatomic molecules [Kop98], to investigate electronic structure of quantum dots [Qu03] and to make it possible to couple atomistic and continuum approaches [Cur03]. So a broad range of physical problems that can be solved by FEM show its importance and uniqueness.

2.1 General aspects with respect to continuum problems

Two procedures are always associated with FEM, namely discretization and integration. The former is directly related to the transformation of differential equation into a system of algebraic equations which then has to be solved numerically [Com94]. The latter applies to the fact, that in FEM each term of the differential equation is written in integral form, and each integral has to be solved at each point of the mesh. The mesh is a common principle of all numerical integrations. Mostly all numerical difficulties that arise during the calculation are associated with the mesh, especially in transient physical problems. Importantly, in FEM the unknown function (field) is approximated by interpolation at the points of mesh.

The general concept to solve a physical problem by using FEM constitutes of the following:

- understanding the physical phenomenon
- writing down the governing partial differential equation subject to the corresponding boundary conditions
- obtaining the equation in FEM formulation which should be solved, i.e. with the projection polynomials and in integral form
- finding a qualitative and reasonable mesh for the physical problem
- running FEM program

In the case of instationary problems the time interval is also responsible for the accuracy and stability of the obtained results. So, in principle, one has to keep in mind this aspect too [Tho97].

2.2 Fisher's system expressed in the form suitable for finite element calculation

Fisher's system reads

$$\begin{cases} \frac{\partial C_g(x, y, t)}{\partial t} - D_g \operatorname{divgrad}[C_g(x, y, t)] = 0 \\ \frac{\partial C_{gb}(y, t)}{\partial t} - D_{gb} \operatorname{divgrad}[C_{gb}(y, t)] - \frac{2D_g}{\delta} \operatorname{grad}_x[C_g(x, y, t)] = 0 \end{cases} \quad (2.1)$$

The weak formulation [Flu92a] requires to first project the differential equation on the projection polynomials based on Galerkin's projective method. Accordingly, the equation for grain diffusion can be written as [Flu92a]

$$\iint_{\Gamma} \alpha_i \frac{\partial C_g(x, y, t)}{\partial t} dx dy + \iint_{\Gamma} \alpha_i \left(-D_g \operatorname{divgrad}[C_g(x, y, t)] \right) dx dy = 0, \quad (2.2)$$

where α_i is the projection polynomial. The second integral in Eq. (2.2) may also be generalized as

$$\iint_{\Gamma} \alpha_i \frac{\partial C_g(x, y, t)}{\partial t} dx dy + \iint_{\Gamma} \alpha_i \operatorname{div} \left\{ -D_g \operatorname{grad}[C_g(x, y, t)] \right\} dx dy = 0. \quad (2.3)$$

By using the mathematical transformation [Flu92a]

$$\begin{aligned} \alpha_i \operatorname{div} \left\{ -D_g \operatorname{grad}[C_g(x, y, t)] \right\} &= \operatorname{div} \left\{ -\alpha_i D_g \operatorname{grad}[C_g(x, y, t)] \right\} \\ &+ D_g \operatorname{grad}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] \end{aligned} \quad (2.4)$$

Eq. (2.3) becomes

$$\begin{aligned} \iint_{\Gamma} \alpha_i \frac{\partial C_g(x, y, t)}{\partial t} dx dy + \iint_{\Gamma} \operatorname{div} \left\{ -\alpha_i D_g \operatorname{grad}[C_g(x, y, t)] \right\} dx dy \\ + \iint_{\Gamma} D_g \operatorname{grad}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] = 0. \end{aligned} \quad (2.5)$$

According to Gauss-Ostrogradsky's theorem [Flu92a], [Kor68] it follows that

$$\iint_{\Gamma} \operatorname{div} \left\{ -\alpha_i D_g \vec{\operatorname{grad}}[C_g(x, y, t)] \right\} dx dy = - \int_{d\Gamma} \alpha_i D_g \vec{\operatorname{grad}}[C_g(x, y, t)] \vec{n} dl. \quad (2.6)$$

Substituting Eq. (2.6) into Eq. (2.5) gives

$$\begin{aligned} & \iint_{\Gamma} \alpha_i \frac{\partial C_g(x, y, t)}{\partial t} dx dy + \int_{d\Gamma} \alpha_i \left| D_g \vec{\operatorname{grad}}[C_g(x, y, t)] \right| \vec{n} dl \\ & + \iint_{\Gamma} D_g \operatorname{grad}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] dx dy = 0 \end{aligned} \quad (2.7)$$

The equation for GB diffusion in Eq. (2.1) is the boundary condition itself and is taken at $x = \pm \delta/2$ (fig. 1.2). Recalling this boundary condition and using the transformations given by Eqs. (2.4) and (2.6), the final form of Fisher's system used in finite element program is

$$\begin{aligned} & \iint_{\Gamma} \alpha_i \frac{\partial C_g(x, y, t)}{\partial t} dx dy + \frac{\delta}{2} \int_{d\Gamma} \alpha_i D_{gb} \vec{\operatorname{grad}}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] dy + \frac{\delta}{2} \int_{d\Gamma} \alpha_i \frac{\partial C_{gb}(y, t)}{\partial t} dy \\ & + \iint_{\Gamma} D_g \operatorname{grad}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] dx dy = 0 \end{aligned} \quad (2.8)$$

According to standard finite element procedure this final expression is used in the matrix form. In the expression the second and fourth integrals represent the conduction parts, whereas the other two integrals are responsible for the time variation. It should be noted that surface conditions (Eq. 1.3a) that exist in the GB diffusion problem are taken into account in Eq. (2.8). Importantly, segregation problems require the use of the same equation in which the second and the third integral are multiplied by the segregation factor s according to Eq. (1.6c).

2.2.1 Finite element method formulation of space charge layer problem

The space charge layer problem is complex due to additional equations and boundary conditions. Fisher's system was extended in order to take into account the space charge effects. According to the model which will be discussed in chapter V the system of partial differential equations for the space charge problem reads

$$\left\{ \begin{array}{l} \frac{\partial C_g(x, y, t)}{\partial t} - D_g \operatorname{divgrad}[C_g(x, y, t)] = 0 \\ \frac{\partial C_{scl}(x, y, t)}{\partial t} - D_g \operatorname{divgrad}[C_{scl}(x, y, t)] = 0 \\ \frac{\partial C_{gb}(x, y, t)}{\partial t} - D_{gb} \operatorname{divgrad}[C_{gb}(y, t)] - \frac{2D_{scl}}{\delta} \vec{\operatorname{grad}}_x [C_{scl}(x, y, t)] = 0 \end{array} \right. , \quad (2.9)$$

where C_{scl} and D_{scl} are the concentration and diffusivity within the SCL. While the continuity condition at the interface between the GB (or the GB core) and space charge layer (SCL) is reflected in the last equation (as it is supposed by Fisher's system), additional boundary conditions should be written for the interface between the SCL and bulk (fig. 5.1). Supposing equal fluxes and concentration at that interface gives

$$C_{scl}(x, y, t) = C_g(x, y, t) \quad \text{at} \quad \mathbf{x} = \pm \left(\frac{\delta}{2} + \delta_{scl} \right)$$

and (2.10)

$$D_{scl} \frac{\partial C_{scl}(x, y, t)}{\partial y} = D_g \frac{\partial C_g(x, y, t)}{\partial y} \quad \text{at} \quad \mathbf{x} = \pm \left(\frac{\delta}{2} + \delta_{scl} \right).$$

The advantage of using the finite element program is that these boundary conditions together with the boundary condition of zero flux at the bottom of geometry used are automatically included. One needs to slightly modify the equation derived for the more general problem, namely Eq. (2.8). This leads to the following expression:

$$\begin{aligned} & \iint_{\Gamma} \alpha_i \frac{\partial C(x, y, t)}{\partial t} dx dy + \frac{\delta}{2} \int_{dr} \alpha_i D_{gb} \vec{\operatorname{grad}}[C_g(x, y, t)] \operatorname{grad}[\alpha_i] dy + \frac{\delta}{2} \int_{dr} \alpha_i \frac{\partial C_{gb}(y, t)}{\partial t} dy \\ & + \iint_{\Gamma} D \operatorname{grad}[C(x, y, t)] \operatorname{grad}[\alpha_i] dx dy = 0 \end{aligned} \quad (2.11)$$

In the last expression C and D can either correspond to the grain or SCL. This means that the same equation is applied to both regions which have finite dimensions. The difference between these regions (grain and SCL) does not lie only in different diffusivities. Applying Eq. (2.11) for the bulk (grain), D_{gb} is not used or equal to zero. This is achieved in the finite element program, at least in a particular case of FLUX-EXPERT used in the current work, by

creating the regions with different properties when generating the mesh for the problem [Sima95]. The finite element calculations are relevant for chapter IV and chapter V. In the following chapter III we numerically evaluate Whipple's solution by using the program MatLab.

2.3 Finite element calculation by using FLUX-EXPERT

FLUX-EXPERT is based on the Expert-System technique [Mas84] and, consequently, is organized in the way that one has to use different modules [Flu00] in order to obtain the solution. Each module represents a program (or a package) for passing a separate step in solution (fig. 2.1). Usually one has to start with definition of the geometrical model and of the mesh. This is done in the program 'SIMAIL' [Sima95], [Sima97] which only creates the mesh, and the result is saved in the file with the extension 'DAT'. The shape of the elements is also determined in this program. In the present study on GB diffusion the triangular elements were only used.

The program FLUX-EXPERT has several libraries of equations for solving particular physical problems. For example, the diffusion equation, called DIFFUS_2D.EQU [Simu02], is usually supplied once the program is installed. However, this equation exists for solving 2D stationary diffusion problems and simple boundary conditions of Neumann type [Flu95a], [Flu99]. Fisher's system represents Fick's second law subject to a specific boundary condition which represents diffusion along a GB and takes into account a leakage of atoms from a GB into a grain. Thus, to simulate GB diffusion, one needs first to create an equation. This can be done in the module 'Generator' [Flu92b]. The 'Generator' is an important feature of FLUX-EXPERT which suggests much more possibilities for the user [Bau85], [Szt92]. This module makes the use of FLUX-EXPERT very flexible since different equations can be met in real research. When describing the equation, the computation algorithms which can be important for this problem are also defined in 'Generator'. To solve Fisher's system, the transient algorithm is necessary. Consequently, the algorithm called 'RLTDMC' was used. This algorithm is non-symmetric, transient, linear, "ill (poorly)-conditioned", i.e. the Gauss method is used for solving the system of equations [Flu92c]. The operator $\partial/\partial t$ is replaced by the corresponding finite difference scheme [Flu92d]. Importantly, the final system containing the integrals and time differences is implicit in nature. The latter avoids the numerical instabilities associated with finite difference schemes when written in explicit form [For60]. However, before describing the equation all the physical properties should be defined. The module

‘GNOYAU’ allows one to define the physical properties, also called identifiers (for example, D_{gb}), and different operators [Flu00] such as the integrand operators, the operators of differential weights of integrals, the algorithm operators. These operators can then be used in the module ‘Generator’ for a particular equation. In the case of Fisher’s system Eq. (2.8) was incorporated and used along all calculations by FLUX-EXPERT. Another feature of FLUX-EXPERT is that all the physical properties are collected in the data base which can be modified by ‘Generator’, i.e. new necessary physical properties can be added. Therefore, D_{gb}

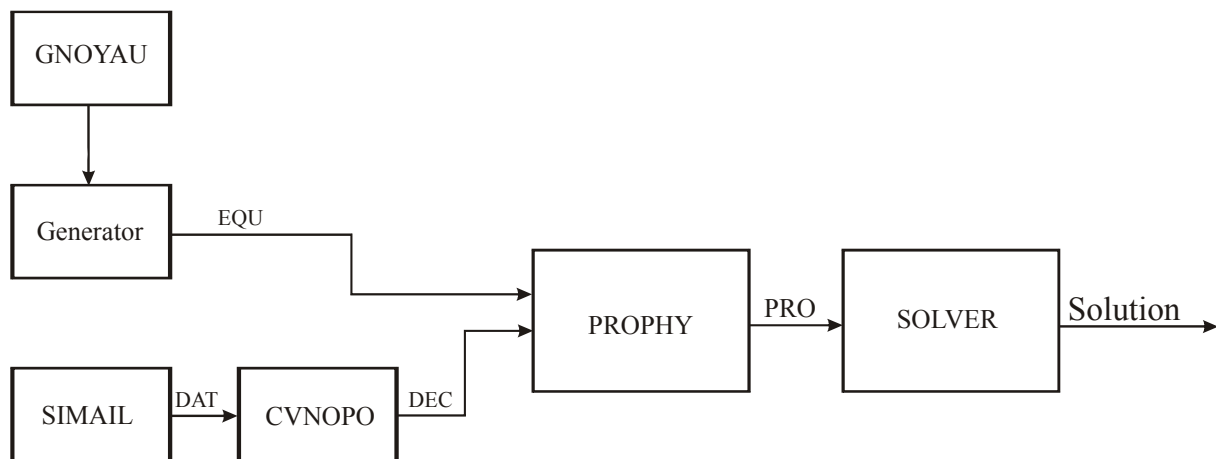


Fig. 2.1 A scheme of steps necessary for obtaining the final solution by using FLUX-EXPERT. Each step represents a module of the program.

was described in order to use it in the equation while D_g and D_{scl} were represented by a diffusion coefficient DC supported by the program [Simu02].

Since the equation and the mesh are defined, these can be used in the module ‘Prophy’ which creates the problem file for integration. Thus, this file contains information on the mesh, the physical properties involved and the equation. The extension of the file produced by ‘Prophy’ is ‘PRO’, and this is usually the biggest file among all the files which are used for solving a physical problem. Finally, the program ‘Solver’ is used to integrate the differential equations and the result of this integration is saved in PRO-file.