

April, 2004

User's Manual *for*

The logo for MultiDiFlux is displayed within a dark red rectangular box with a thin white border. The text 'MultiDiFlux' is written in a stylized, multi-colored font: 'Multi' is yellow, 'Di' is green, and 'Flux' is orange. A small white copyright symbol (©) is located to the right of the text.

MultiDiFlux ©

Version 1.2

*A computational environment
For the analysis of
Multi-component Diffusion
for
Interdiffusion Fluxes and Interdiffusion Coefficients
And
Diffusion Paths*



M. A. Dayananda
Purdue University
dayanand@ecn.purdue.edu



L. R. Ram-Mohan
QSA
lrram@wpi.edu

This software is based upon the work supported by the National Science Foundation under grant No. 0304777

This material is based upon work supported by the National Science Foundation under Grant No. 0304777.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

	Page
I Introduction	4
II Preparation of an Input File for use in <i>MultiDiFlux</i>	6
III Running the program <i>MultiDiFlux</i>	12
IV The output files	13
V Methodology	14
<i>A Data Smoothing</i>	14
<i>B Determining the Matano plane</i>	15
<i>C Obtaining the flux and the local diffusion coefficients ..</i>	15
<i>L Solving the diffusion equation</i>	15
VI Sample Input Files	16
VII Output Plots for a5 vs. a12	21

I. Introduction

MultiDiFlux provides a computational environment for the analysis of experimentally determined data (C_i as a function of x) for concentration profiles of single-phase and multi-phase diffusion couples in isothermal multicomponent systems. Interdiffusion in an n -component system is phenomenologically described by the equation

$$\frac{\partial C_i(x, t)}{\partial t} = \frac{\partial}{\partial x} \sum_{j=1}^{n-1} \tilde{D}_{ij}^n \frac{\partial C_i(x, t)}{\partial x} \quad (i, j=1, 2, \dots, n)$$

In general, the interdiffusion coefficients \tilde{D}_{ij}^n are not constants; rather, they are nonlinear functions of the composition variables, C_i , in the diffusion zone. If $\sum_j C_j = \rho$ (density), there exist only $(n-1)$ independent concentration variables, $(C_1, C_2, \dots, C_{n-1})$. Similarly, on a frame of reference employed for the determination of interdiffusion fluxes, such as laboratory-fixed frame, volume-fixed frame or mass-fixed frame, there exist only $n-1$ independent interdiffusion fluxes that need to be determined. Hence, one requires a knowledge of $(n-1)^2$ interdiffusion coefficients as functions of composition in a n -component system. For a constant molar volume system, the interdiffusion fluxes based on a laboratory frame of reference are identified by \tilde{J}_i and $\sum_{i=1}^n \tilde{J}_i = 0$.

The analysis of the experimental data on concentration profiles of diffusion couples in multicomponent systems becomes complicated by the concentration dependence of the diffusion coefficients and the nonlinearity in the diffusion equation. We are thus challenged in our attempts to **predict** the diffusion path experimentally adopted by a multi-component diffusion couple assembled with two terminal alloys of known, initial compositions.

For a ternary system with three components, we need 4 interdiffusion coefficients. For the determination of these \tilde{D}_{ij}^3 ($i = 1, 2$) coefficients as functions of composition, it is necessary to perform isothermal experiments with at least two diffusion couples in order to obtain two diffusion paths (composition paths) that intersect on a ternary isotherm. We can then set up four independent equations and solve for the four interdiffusion coefficients at the common composition of the two couples. Clearly, a very large number of experiments will have to be performed in order to fully map out the interdiffusion coefficients \tilde{D}_{ij}^3 over the entire C_1 - C_2 plane for a ternary system. Such knowledge of the functional form of the interdiffusion coefficients is essential to be able to predict the diffusion paths for ternary diffusion couples.

Dayananda¹ has shown that the concentration profiles of isothermal, solid-solid diffusion couples can be analyzed for interdiffusion fluxes of all components directly from their concentration profiles without a prior knowledge of the interdiffusion coefficients. In addition, based on a new analysis, Dayananda and Sohn² have also shown that it is possible to determine ternary interdiffusion coefficients over selected composition ranges along the diffusion path of a ternary diffusion couple. This new approach to the determination of interdiffusion coefficients from a single diffusion couple is an important advancement in the methodology of multi-component diffusion analysis and circumvents the need for multiple diffusion experiments with intersecting diffusion paths.

The **MultiDiFlux** program implements the Dayananda technique (See References 1 and 2) for obtaining the interdiffusion fluxes of all components over the entire diffusion zone as well as determining the interdiffusion coefficients over selected composition ranges along the entire diffusion path of a diffusion couple. The interdiffusion coefficients, which are determined for the concentration ranges selected within the diffusion zone, represent average diffusion coefficients over those ranges. With the diffusion coefficients thus determined at a selected temperature, we can predict composition profiles for a given diffusion couple at the selected temperature. Thus, we have the means of comparing predicted concentration profiles with the input experimental curves. Once a sufficiently large data-bank is set up for the ternary diffusion coefficients over a wide concentration range, we would be able to use such a “look-up table” of coefficients to predict the diffusion path for any diffusion couple assembled with terminal alloys of selected compositions.

This version (1.2) of the **MultiDiFlux** software is capable of analyzing mainly single-phase diffusion problems containing several components. However, the program will be extended to cover the analysis of multiphase systems.

¹ M.A. Dayananda: Metall. Trans. A, 14A, 1851-1858 (1983).

² M.A. Dayananda and Y.H. Sohn: Metallurgical and Materials Transactions **30A**, (1999) p535.

II. Preparation of an Input File for Use in *MultiDiFlux*

In the input file the user needs to provide all the needed parameters for the program. All lines that begin with the “#” sign in the file are ignored by the input parser. This allows the user to add his own explanations into the input file that would be of use to him. ***No blank lines are to be used in the input file.*** Each entry in the input file is given on a new line. The various entries in the input file are shown in a sample file reproduced in Section VI and explained in the following paragraphs.

The first two entries are shown below:

#1. Output control: (Select program output level: 0/1)

1

#2. Number of Components

3

1. In the first entry, the **output level** is selected. The output level determines how much information will be displayed in the command window as the program is running. The user should select either 0 or 1 as the output level.
2. Next, we specify the **number of components** in the diffusion couple to be analyzed. For example, for a binary diffusion couple we will indicate the number of components as **2**; in a ternary diffusion couple, the number of components would be set at **3**.

The next three entries are shown below:

#3. Experimental Data File Name for Concentration(C) vs Distance(x) plot

Example: a5_a12.dat

Example: alloyA_alloyB_long.dat

Example: alloyA_alloyB.dat

a5_a12.dat

#4. Index for the Dependent Component

3

#5. Single phase or Multi phase

Switch (Single/Multi) (0/1)

0

3. In this entry, we enter the **name of the experimental data file** that contains the experimental data for the concentration profiles of the individual components. The extension of the data file must be “.dat”. The user should open the data file a5_a12.dat to examine the proper way to format a data file. Data from a diffusion couple is input into a data file as follows: the distance coordinate (x) followed by the concentration values (C_1 , C_2 , C_n) of the individual components at that coordinate. Note that on each line for a given coordinate value of x , we list the values of $C_i(x)$ in the sequence

identified as 1 through n with no commas for separating the data. The unit of the distance coordinate should be μm and the unit of the concentration values should be atomic fraction. Comments can be added to the data files by using the “#” symbol at the very beginning of a comment line.³

4. We next specify which one of the components is considered as the dependent concentration variable in our analysis. This is done by labeling or numbering the individual components by 1 through n and identifying the **dependent component by its number or label**. For example, in a ternary system, if component 3 is considered as the dependent concentration variable, we enter **3** for the dependent component. Components 1 and 2 are treated as independent concentrations in the diffusion equations.
5. Next, the **type of diffusion couple** to be analyzed is specified. The current version can handle only single phase diffusion couples.

The next seven entries in the input file are:

#6. Enter the number (n_x) of partitioned regions along the distance coordinate for interpolating the concentration profiles. Here n_x can be any positive definite integer; if the entire concentration profile is picked to be one region, enter: 1

5

#7 Each partitioned region is subdivided into elements for interpolation

Note: This has no effect on the partitioning done for finding the diffusion coefficients. Both are independent.

1. For every region indicate the number of interpolation elements (nelem).
2. Total number of regions must match the number specified above.
3. The partitions must span the entire domain.
4. nelem must be a positive integer.
5. For single phase: The xright of the first region equal the xleft of the second region and so on for all subsequent regions.

Region	xleft	xright	nelem
1	0	60	1
2	60	125	2
3	125	240	3
4	240	290	2
5	290	330	2

#8. Specify the number of points to be used for calculating the first derivative using the finite differencing scheme. Allowed values are: 3, 5, 7, 9

³ The input parser looks for the **first** character of each line to decide whether the line is a comment line that needs to be ignored.

Note: The number of points in each region must exceed the number that is specified below.

5

- #9. This switch determines whether or not to read the file "deriv_nodes_data.out" in order to get the first derivative of the concentration for each component.

Switch (Yes/No) (1/0)

1

- #10. Filename or pathname of the file containing the first derivatives of Concentration (C_i). This file is same or similar to deriv_nodes_data.out.

deriv_nodes_data_a5a12.inp

- #11. Select the common Matano line from a particular Concentration C_i .

Set switch to 0 to specify your own Matano coordinate OR

Set the switch to the composition number and specify the corresponding Matano coordinate for further calculations.

Switch (0/1/2/3/...)|| User specified matano coordinate

1

- #12. Enter the molar volumes for each component of the alloy in the diffusion zone. This will be used for calculating the interdiffusion flux using a formula, which does not require the Matano line. The unit of the molar volume must be entered in $\text{g} \cdot \text{mole} / \mu\text{m}^3$.

The numbers must be separated by spaces or tabs. There should be as many numbers as the number of components (Enter 1 if molar volume is unknown).

1.0 1.0 1.0

6. The concentration profile is divided into separate regions for interpolation. In order to choose the entire concentration profile, the user should select 1.
7. This entry tells the program how many elements each partitioned region (selected in entry 6) will be divided into. Since the concentration vs. distance coordinate (C_i vs. x) data are usually given at discrete points, we use Hermite interpolation in order to obtain a cubic fit over each segment of the data. A description of the procedure is given in Section V: Methodology. When the concentration profile is divided into multiple regions, the user must specify the Region Number, the x -coordinate at the left of the region (x_{left}), the x -coordinate at the right of the region (x_{right}), and the number of elements the region is to be divided into (n_{elm}). Note that in single phase mode, the x_{right} of the first region must be equal to the x_{left} of the second region and so on for all subsequent regions.
8. This entry tells MultiDiFlux how many points are to be used for calculating the first derivatives to be used by the finite differencing scheme. Note that the number of points in each region must exceed the number of points that is specified in this entry.
9. This switch tells the program whether or not to look for a user defined derivative file. The derivative file tells the program the first derivative of each component at the x_{right} and x_{left} location in each region defined in entry 7. This file is useful for forcing the

concentration profiles to be flat in the terminal regions or for defining the derivative at a local minima or maxima to be zero. If the user does not have a pre-existing derivative file, the switch should be set to 0 and MultiDiFlux will calculate the first derivatives at the edges of all the interpolation zones. MultiDiFlux will then create a derivative file called Deriv_nodes_data.out. This file can then be copied and made into a user defined derivative file. This is further explained in Section IV. An example of how this input file should look is provided in the Section VI.

10. The next input defines the filename of the derivative file. If the switch is turned to 1 in entry 9, only then will the derivative file name be read.
11. This input allows the user to choose which Matano plane is used for flux and diffusion coefficient calculations. The user can select the Matano plane of component 1, 2 or 3 by entering its respective number. A user defined Matano plane can also be used by entering 0, followed by the location of the user defined Matano plane in micrometers.
12. This entry defines the molar volume of the alloy in the diffusion zone. The fluxes of each component can be calculated without utilizing the Matano Plane location (x_0) by using the equation⁴:

$$\tilde{J}_i(x^*) = \frac{(C_i^- - C_i^+)}{2t} \left[Y_i^* \int_{-\infty}^{x^*} \frac{(1 - Y_i)}{V_m} dx + (1 - Y_i^*) \int_{x^*}^{+\infty} \frac{Y_i}{V_m} dx \right] \quad (i = 1, 2, \dots, n)$$

$$Y_i = \frac{C_i - C_i^+}{C_i^- - C_i^+}$$

V_m = Molar Volume

For some cases, the molar volume can be taken as a constant. In that case, the V_m may be set to 1 to get fluxes calculated in units of (atom fraction- $\mu\text{m}/\text{sec}$). If V_m is entered in units of ($\text{g}^*\text{moles}/\mu\text{m}^3$), the calculated fluxes will be in ($\text{g}^*\text{moles}/\mu\text{m}^2*\text{s}$).

The next four entries to the input file are described below:

#13 No. of DIFFUSION ZONES for Diffusion Analysis.
 (Example1: If the entire concentration profile is selected as one diffusion zone, enter: 1)
 (Example2: For two diffusion zones enter 2)
 (Example3: Enter 3 if three zones are desired)

2

#14 Each diffusion zone is further subdivided into regions or concentration ranges for DIFFUSION ANALYSIS.
 Note: This has no effect on the partition done for the purpose of interpolation. Both are independent.

⁴ See reference 1 on page 5

1. For every zone, indicate the number of regions (nregion).
2. Total number of zones must match the number specified above.
3. The zones must span the entire concentration profile.
4. nregion must be a positive integer.
5. For single phase diffusion, the xright of the first zone must be equal to the xleft of the second zone and so on for all subsequent zones.

Zone	xleft	xright	nregion
1	0	196	1
2	196	330	1

#15 DIFFUSION TIME (seconds)

172800

#16 Scale Factor for DIFFUSION COEFFICIENTS

Example: 1.0e-12

1e-12

#17 Output Control: Enter the number of points for getting out concentrations and fluxes at uniformly spaced points for plotting graphs.

n_Concentration_pts

300

n_flux_pts

300

13. This entry divides the concentration profile into Diffusion Zones over which diffusion coefficients will be calculated.
14. Each Diffusion Zone will be further divided into Diffusion Regions over which a set of interdiffusion coefficients will be calculated. When the concentration profile is divided into multiple zones, the user must specify the Zone Number, the x-coordinate at the left of the zone (xleft), the x-coordinate at the right of the zone (xright), and the number of regions into which the zone is to be divided (nregion). Note that in single phase mode, the xright of the first zone must be equal to the xleft of the second zone and so on for all subsequent zones.
15. The next input corresponds to the time t of diffusion for the experiment. The time must be expressed in seconds. If the diffusion time happens to be 48 hours, then the entry would be 172800 seconds.
16. From the concentration profiles (C_i vs. x plots), the interdiffusion flux $\tilde{J}_i(x)$ of component i is calculated directly from the relation⁵

$$\tilde{J}_i(x) = \frac{1}{2t} \int_{C_i^+ \text{ or } C_i^-}^{C_i(x)} (x-x_o) dC_i \quad (i = 1,2,3)$$

or

⁵ See reference 1 on page 5

$$\tilde{J}_i(x^*) = \frac{(C_i^- - C_i^+)}{2t} \left[Y_i^* \int_{-\infty}^{x^*} \frac{(1 - Y_i)}{V_m} dx + (1 - Y_i^*) \int_{x^*}^{+\infty} \frac{Y_i}{V_m} dx \right] \quad (i = 1, 2, \dots, n)$$

$$Y_i = \frac{C_i - C_i^+}{C_i^- - C_i^+}$$

V_m = Molar Volume

where C_i^- and C_i^+ correspond, respectively, to the concentrations of component i in alloys a_5 and a_{12} employed as the terminal alloys of the diffusion couple. The calculated fluxes (atom fraction. $\mu\text{m.s}^{-1}$) can be converted to fluxes in units of [g.moles/($\text{m}^2.\text{s}$)] by multiplying by $1 \cdot 10^{-6}$ m and by the molar density ρ (g.moles/ m^3) of the alloy system, provided the density is assumed constant in the diffusion zone.

From the knowledge of the interdiffusion fluxes, interdiffusion coefficients are calculated over various segments of the concentration profiles by this program. Since these calculated coefficients will be in the units of $\mu\text{m}^2/\text{s}$, we may like to convert it to the traditional unit of m^2/s , by utilizing the conversion factor of 10^{-12} . This scale factor can be input into this section. If no scale factor is desired, then the user should enter 1 for this entry.

17. The last inputs tell MultiDiFlux how many evenly spaced concentration and flux data points to output into the output files identified as `uniform_conc_pts.out` and `uniform_flux_pts.out`. The number entered in this section corresponds to the number of evenly spaced x locations, at which the C_i and J_i are output.

III. Running the program *MultiDiFlux*

MultiDiFlux is now in the developmental stage and a graphical user interface will be available as the development reaches an advanced stage. At present, the program runs on the Intel-based personal computers using the Microsoft Windows-NT/2000/XP operating systems. At present, we begin by opening an MS_DOS command prompt window and connecting from within it to the folder (directory) containing the *MultiDiFlux* program, input file and the file containing the input data on concentration profiles. This folder is named “`multidiflux 1.2`” in the following example.

If the `multidiflux 1.2` folder is in drive C, type `cd multidiflux 1.2` after the command prompt. The MS_DOS window, as we will designate it, will look as shown below:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\> cd multidiflux 1.2
```

Press the return key and the MS_DOS window will look like:

```
C:\multidiflux 1.2>
```

Type `diffusion.exe` followed by the name of an input file. Let the name of the input file for a calculation be “`inputa5a12.inp`” where `inputa5a12.inp` refers to an input file for a couple assembled with a5 and a12 as the terminal alloys. The extension “.inp” is required for this file. At the prompt in the MS_DOS window, type the command as shown below:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\multidiflux 1.2>diffusion.exe inputa5a12.inp
```

Press the return key. This starts the program by opening the input file containing the name of the experimental data file for a couple (identified as `a5_a12.dat`, for example) for analysis and computation. Note that the name of the data file is one of the parameters to be entered in the input file. As the computation progresses, the MS_DOS window reports on the various stages of the calculations as well as their completion. At the end of the computation, all the output files (described in Chapter IV) will be stored in the “`data`” folder.

IV. The Output Files

Once the **MultiDiFlux** program is run, the program's output files will be available in the folder named "Data". The original experimental data that was loaded into the program is located in **expt_data.out**. **MultiDiFlux** smoothes the data by Hermite interpolation functions defined over each of the interpolation regions. The location of the Matano plane for each component is then determined from the interpolated concentration profiles and is put in the output file **Matano0.out**. The interpolated data points are put in the file **conc_interp.out**. The **uniform_conc_pts.out** file contains the locations of the Matano planes and as many evenly spaced interpolated concentration data points as specified in the input file (see #17 in Section II). These files will have the x-coordinate in the first column, followed by the three concentration values at that x-coordinate. If the **expt_data.out** and the **uniform_conc_pts.out** files are plotted using a standard plotting program such as "gnuplot", we can display the experimental data and the interpolated concentration profile together and also show the Matano line on the same graph. Sample output plots showing these files plotted together can be seen in Section VII, Figure 1.

Another important output file is the **deriv_nodes_data.out** file. This file contains the first derivatives calculated at the **xleft** and **xright** locations of each interpolation region. By copying this file to another folder, renaming it, and referencing it in the input file, the user can create a derivative file where the user can control the derivatives at the ends of each interpolation region. When a derivative file is referenced in the input file, the derivatives at the ends of the interpolation regions will not be calculated and will simply be read in from the derivative file.

The interdiffusion fluxes are calculated directly⁶ by integrating from either the left side or the right side of the concentration profiles utilizing the Matano plane location from the chosen component (see #11 in Section II). The outputs are stored in **flux_left.out** and **flux_right.out** respectively. The fluxes can also be calculated without utilizing the location of the Matano Plane⁶. This output is stored in **flux_mv.out** and **uniform_flux_pts.out**. The difference between the two files is that **flux_mv.out** contains the same number of data points as are contained in **expt_data.out**, while **uniform_flux_pts.out** contains as many evenly spaced data points as the user specified in the input file (see #17 in Section II). These files contain the x-coordinates in the first column and the values of the flux at each coordinate for each of the diffusing species. Sample output plots of the **Uniform_flux_pts.out** file can be seen in Section VII, Figure2.

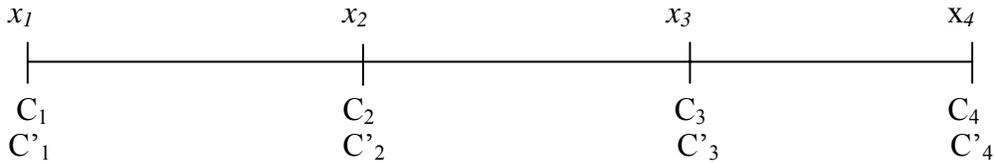
The moments of the flux over local regions are used to obtain independent equations for the determination of the local, averaged values of the interdiffusion coefficients. These coefficients are calculated from the flux profile contained in the **Flux_mv.out** file. The calculated diffusion coefficients are displayed in **diff_coeff_Jmv.out**. The number of "diffusion regions" defined in the input file specifies the number of evenly divided partitions over which the averaging process is done. The **diff_coeff_mv.out** file contains the interdiffusion coefficients \tilde{D}_{ij}^3 in each interval. This can be represented graphically in gnuplot for visualization purposes.

⁶ See reference 1 on page 5

V. Methodology

A. Data smoothing:

We break up the physical region into a number of interpolation partitions, as we call these regions. Over each region, we represent the data by cubic Hermite interpolation polynomials. At the two extreme ends of the diffusion couple, we suppose that the concentration curves have zero slopes. The two end values of the concentration are read in from the data set. We then use the cubic Hermite interpolation polynomials in each interpolation region to represent the concentration curves in terms of unknown concentration values and the slopes of the concentration curves at the two extremities of the interpolation region. In the figure below we have considered three interpolation regions with four nodes. At each of the nodes we have two “degrees of freedom” in the variables C_i and $C'_i = dC_i(x)/dx$. Here the index i refers to the coordinate nodes as shown below.



So if the initial and final points correspond to x_1 and x_4 , we suppose that C'_1 and C'_4 are zero and C_1 and C_2 are determined by the experimentally measured values, as reported in the input data file. We are then left with the pairs of unknown nodal values at each internal node.

The Hermite interpolation polynomials, denoted by $N_i(x)$, are just the shape functions used in finite element analysis. These shape functions are set to zero outside the element being considered. Thus the concentration function $C(x)$ in a given element is given by $C(x) = C_i N_i(x)$. We now extend this representation of $C(x)$ over the entire domain, and designate this by using the index α . Then at each coordinate point where we have experimental data we have $C(x_i) = N_{\alpha}(x_i)C_{\alpha} \equiv A_{i\alpha} C_{\alpha}$, say. The $A_{i\alpha}$ form the so-called *design matrix*. We now minimize the quantity

$$F = \frac{1}{2} [C_{\text{exp}}(x_i) - C(x_i)]^2$$

to obtain

$$[C_{\text{exp}}(x_i) - A_{i\alpha} C_{\alpha}] A_{i\beta} = 0$$

or

$$(A^T \cdot A)_{\beta\alpha} C_{\alpha} = A_{\beta i} C_{\text{exp}}(x_i).$$

These equations are called the normal equations in the numerical analysis literature. The right side is known and the design matrices are known, so that we can solve for the unknown C_{α} .

This method of data fitting using local elements is an extension of the method of the General Linear Least Squares fitting developed by us.

B. Determining the Matano plane

Once a piecewise fit to the concentration curves are given over each of the interpolation partitions, we can determine the Matano plane by an integration with a root finding procedure to determine its location through determination of equal areas or mass balance relative to the Matano plane. All integrations are performed using double-precision Gauss-Legendre quadrature. The root finding is done using Brent's algorithm.

C. Obtaining the flux plane and the local diffusion coefficients in diffusion partitions

We use the method of Dayananda to determine the local fluxes and the spatially averaged diffusion coefficients over each of the diffusion partitions.

D. Solving the diffusion equation

The diffusion equation is solved using the method of finite elements. We break up the physical region into small elements. The number of elements is controlled through the input file. Over each element we use a Hermite polynomial representation of the concentrations and evaluate the nodal unknowns using the Galerkin method of projection and evaluation of the weighted residuals. The interdiffusion coefficients determined by the averaging method of Dayananda are saved as values in the "concentration space". We then are able to insert the local values of the interdiffusion coefficient for each value of the concentrations obtained through a look-up table used to obtain the appropriate interdiffusion coefficients for a given point in the concentration or composition space. Once the nodal values of the concentrations are solved for in the finite element method, we then reconstruct the concentration curves using the same interpolation polynomials. We have the ability to use linear Lagrange shape functions, quadratic Lagrange shape functions, or cubic Hermite shape functions, or quintic Hermite shape functions for a controlled way of improving accuracy, besides increasing the number of elements.

The calculations use the sparse matrix techniques developed at Quantum Semiconductor Algorithms, Inc. All matrices generated for the FEM are stored as linked lists, with no zeros being stored in memory. This allows for efficient memory usage on the computer. Also, the matrix inversions are done very rapidly with methods that make use of the sparse nature of the occupancy in the matrices. The finite element global matrices are already banded in the case of 1D calculations.

VI. Sample Input File

A. Input file for a5_a12.dat:

```
#.....  
# MultiDiFlux: A Software Environment for the  
# Analysis of Multi-component, Multi-phase  
# Metallic Diffusion Data  
# Copyright (c) 2001 - 2004  
# Quantum Semiconductor Algorithms, Inc.  
#  
# All lines starting with '#' are ignored  
#.....  
#=====
```

```
#=====
```

```
# Output control: (Select program output level: 0/1)  
#.....  
1  
#.....  
# Number of Components (eg: 3 for ternary)  
#.....  
3  
#.....  
# Experimental Data File Name for Concentration(C) vs Distance(x) plot  
# Example: a5_a12.dat  
# Example: alloyA_alloyB_long.dat  
# Example: alloyA_alloyB.dat  
#.....  
a5_a12.dat  
#.....  
# Index of the Dependent Component  
# (eg: 1 or 2 or 3 for ternary couples)  
#.....  
3  
#.....  
# Single phase or Multi phase  
# Switch (Single/Multi) (0/1)  
#.....  
0  
#.....  
#=====
```

```
# Partitions for DATA INTERPOLATION: Using Distance Coordinate.  
# =====
```

```
# Enter the number (n_x) of partitioned regions along the  
# distance coordinate for interpolating the concentration profiles.  
# Here n_x can be any positive definite integer; if the entire
```

```

# concentration profile is picked to be one region, enter: 1
#.....
5
#.....
# Each partitioned region is subdivided into elements
# for interpolation
# Note: This has no effect on the partitioning done for finding
# the diffusion coefficients. Both are independent.
# 1. For every region indicate the number of interpolation
#     elements (nelem).
# 2. Total number of regions must match the number specified above.
# 3. The partitions must span the entire domain.
# 4. nelem must be a positive integer.
# 5. For single phase: The xright of the first region equal the xleft of
#     the second region and so on for all subsequent regions.
#.....
#Region    xleft    xright  nelem
  1         0      60      1
  2        60     125     2
  3       125     240     3
  4       240     290     2
  5       290     330     2
#.....
# Specify the number of points to be used for calculating the first derivative
# using the finite differencing scheme. Allowed values are:
# 3, 5, 7, 9
# Note: The number of points in each region must exceed
# the number that is specified below.
#.....
5
#.....
# This switch determines whether or not to read the file
# "deriv_nodes_data.out" in order to get the first derivative
# of the concentration for each component.
# Switch (Yes/No) (1/0)
#.....
1
#.....
# Filename or pathname of the file containing the first derivatives
# of Concentration (C_i). This file is same or similar to deriv_nodes_data.out.
#.....
deriv_nodes_data_a5a12.inp
#.....
# Selecting the common Matano line from a particular Concentration C_i.
# Set switch to 0 to specify your own Matano coordinate OR
# Set the switch to the composition number and specify the

```

```

# corresponding Matano coordinate for further calculations.
# Switch (0/1/2/3/...)|| User specified matano coordinate
#.....
1
#.....
#Enter the molar volumes for each component of the alloy in the diffusion zone.
#This will be used for calculating the interdiffusion flux using a formula, which
#does not require the Matano line. The unit of the molar volume must be entered
#in g*mole/ $\mu\text{m}^3$ .
#The numbers must be separated by spaces or tabs. There should be as many
#numbers as the number of components (Enter 1 if molar volume is unknown).
#.....
1.0 1.0 1.0
#.....
#=====
#=====
# No. of DIFFUSION ZONES for Diffusion Analysis.
# (Example1: If the entire concentration profile is selected as one
# diffusion zone, enter: 1)
# (Example2: For two diffusion zones enter 2)
# (Example3: Enter 3 if three zones are desired)
#.....
2
#.....
#=====
# Each diffusion zone is further subdivided into regions or
#concentration ranges for DIFFUSION ANALYSIS.
# Note: This has no effect on the partition done for the purpose
# of interpolation. Both are independent.
# 1. For every zone, indicate the number of regions (nregions).
# 2. Total number of zones must match the number specified above.
# 3. The zones must span the entire concentration profile.
# 4. nregion must be a positive integer.
# 5. For single phase diffusion, the xright of the first zone must
# be equal to the xleft of the second zone and so on for all
# subsequent zones.
#.....
#Zone xleft xright nregion
1 0 196 1
2 196 330 1
#.....
#=====
# DIFFUSION TIME (seconds)
#.....
172800
#.....

```

```

=====
# Scale Factor for DIFFUSION COEFFICIENTS
# Example: 1.0e-12
# .....
1e-12
# .....
# .....
=====
# Output Control: Enter the number of points for getting out
# concentrations and fluxes at uniformly spaced points for plotting
# graphs.
# n_Concentration_pts
# .....
300
# .....
# n_flux_pts
# .....
300
# .....
=====

```

B. Derivative file for a5_a12.dat:

```

=====
# Filename: deriv_nodes_data_a5a12.inp
# Date:
# .....
# This file lists the derivatives of C_i at the beginning and
# the end of each interpolation region. Each time the program
# is run, this file will be over-written. So, copy this file
# and rename it (e.g.deriv_nodes_data.inp). Modify your input
# file accordingly. Change the entries if the derivatives
# need adjustment before calculating diffusion coefficients.
# .....
# Region  xleft      xright      dC1_left    dC1_right
# .....
1      +0.0000e+000 +6.0000e+001 0          0
2      +6.0000e+001 +1.2500e+002 0          -8.2875e-004
3      +1.2500e+002 +2.4000e+002 -8.2875e-004 -1.3817e-003
4      +2.4000e+002 +2.9000e+002 -1.3817e-003 0
5      +2.9000e+002 +3.3000e+002 0          0
# .....
# Region  xleft      xright      dC2_left    dC2_right
# .....
1      +0.0000e+000 +6.0000e+001 0          0
2      +6.0000e+001 +1.2500e+002 0          0
3      +1.2500e+002 +2.4000e+002 0          0
# .....

```

4	+2.4000e+002	+2.9000e+002	0	+0.0000e+000	
5	+2.9000e+002	+3.3000e+002	+0.0000e+000	+0.0000e+000	
#=====					
#	Region	xleft	xright	dC3_left	dC3_right
#=====					
1		+0.0000e+000	+6.0000e+001	0	0
2		+6.0000e+001	+1.2500e+002	0	+9.1583e-004
3		+1.2500e+002	+2.4000e+002	+9.1583e-004	+1.8983e-003
4		+2.4000e+002	+2.9000e+002	+1.8983e-003	0
5		+2.9000e+002	+3.3000e+002	0	0

VII. Sample Output File for a5 vs a12

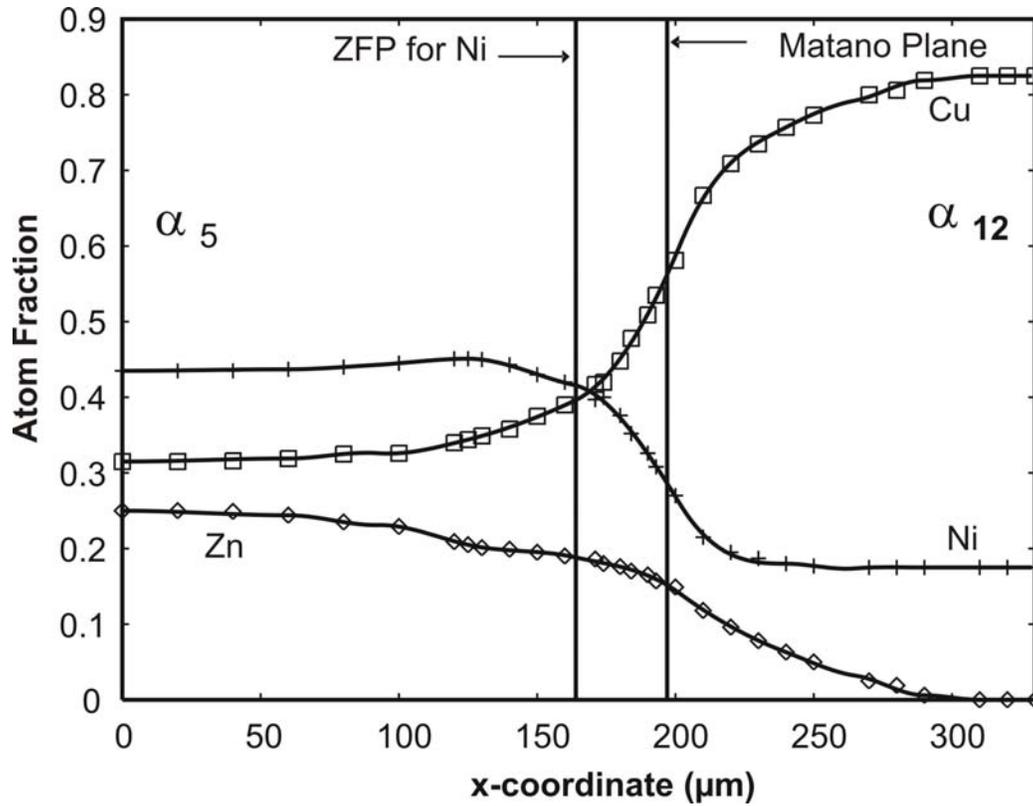


Figure 1: Experimental data (Expt_data.out) and interpolated curves (Uniform_conc_pts.out) for the a5 vs. a12 couple are shown. Data for components 1, 2, and 3 (Zn, Ni, Cu) are identified, respectively, by a “ \diamond ”, “+”, and a “ \square ”.

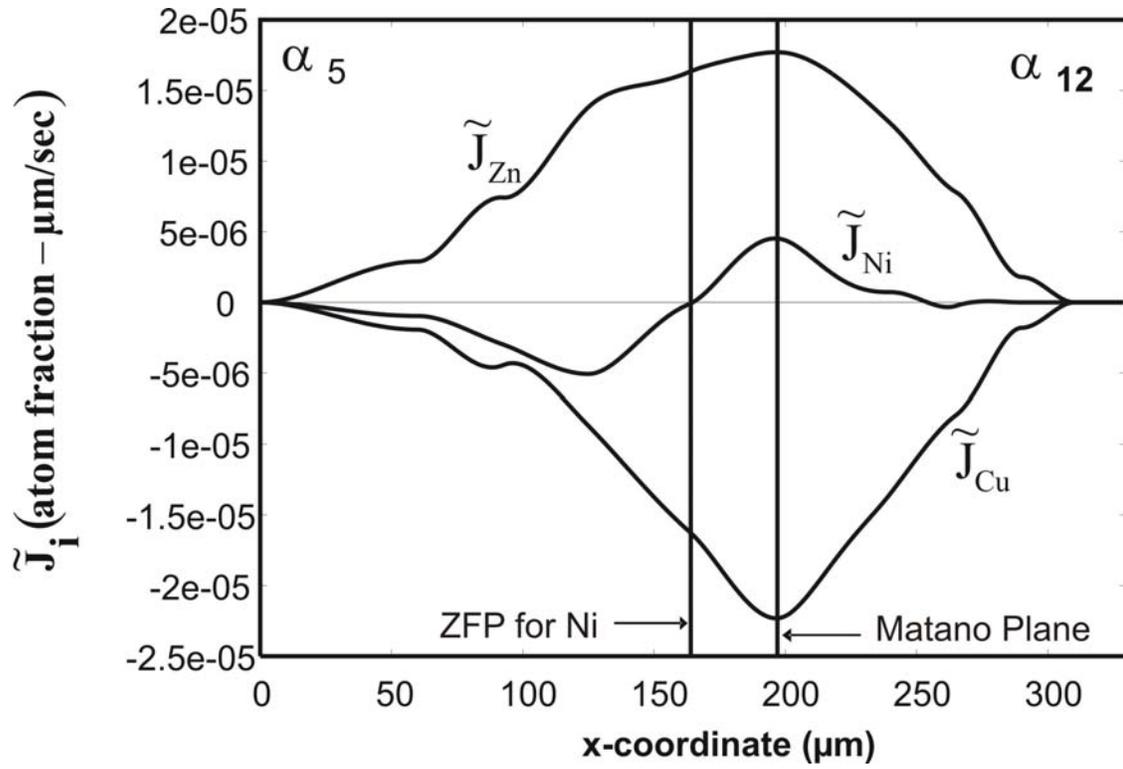


Figure 2: The flux profiles (Uniform_flux_pts.out) calculated for various components 1, 2, 3 (Zn, Ni, Cu) for the a5 vs. a12 couple.