

Diffusion Module (DICTRA) Quick Start Guide - Console Mode

Thermo-Calc Version 2021a



Copyright 2021 Thermo-Calc Software AB. All rights reserved.

Information in this document is subject to change without notice. The software or database described in this document is furnished under a license agreement or nondisclosure agreement. The software or database may be used or copied only in accordance with the terms of those agreements.

Thermo-Calc Software AB

Råsundavägen 18, SE-169 67 Solna, Sweden

+46 8 545 959 30

documentation@thermocalc.com

www.thermocalc.com

Introduction to the Diffusion Module (DICTRA)

The Diffusion Module (DICTRA) is an Add-on Module to the Thermo-Calc software package. It is used for simulation of diffusion controlled transformations in multicomponent systems. The simulations are both time- and space-dependent.

The Diffusion Module (DICTRA) is suitable for solving diffusion problems that include a moving boundary (so-called *Stefan problems*). The multicomponent diffusion equations in the various regions of a material are solved under the assumption that thermodynamic equilibrium holds locally at all phase interfaces. The concentration fields only vary along one spatial coordinate, but the geometry may be either planar, cylindrical or spherical.

In this section:

About this Quick Start Guide	4
About the DICTRA File Formats	4
Opening the Module and Macro Files	4
Basic Terminology	7
Setting up a Simulation in Console Mode	8

About this Quick Start Guide

The following introduces you to the basic concepts, steps and theory to start using the Diffusion Module (DICTRA). An understanding of this information is necessary to ensure accurate calculations.

- ▶ More details about the use of DICTRA can be found in Andersson et. al (2002)¹ and Borgenstam et al. (2000)².



If you are not familiar with Graphical Mode, it is recommended you review the available documentation and training material.

About the DICTRA File Formats

The Diffusion Module (DICTRA) in Console Mode uses different file formats: *log*, *macro* and *workspace* files.

Log Files

Log files (*.TCM) are plain text files and are just macro files used to save a sequence of commands. These files can be edited in a text editor.

Macro Files

Macro files (*.DCM) are plain text files used to save a sequence of commands that can be loaded and executed. Macro files can be edited in a text editor. When creating a command sequence, you add comments to the file by starting a line with @@.

Workspace Files

Workspaces files for the Diffusion Module (DICTRA) have the file extension *.DIC, which allow you to save all the data in your current workspace. A workspace contains all the data, specified settings and the result of any calculations performed. The saved data includes original and modified thermodynamic data, the last set of conditions and options, and the results of any simulations.

Opening the Module and Macro Files

1. Andersson, J.-O., Helander, T., Höglund, L., Shi, P., and Sundman, B., *Calphad*, 26 (2), pp. 273–312.

2. Borgenstam, Al, Höglund, L., Ågren, J. and Engström, A., *J. Ph. Equilib.*, 21 (3), pp. 269–280.

You can go into the Console Mode to explore the command options, or continue with this quick start guide, which uses macro files to demonstrate how to use the software.

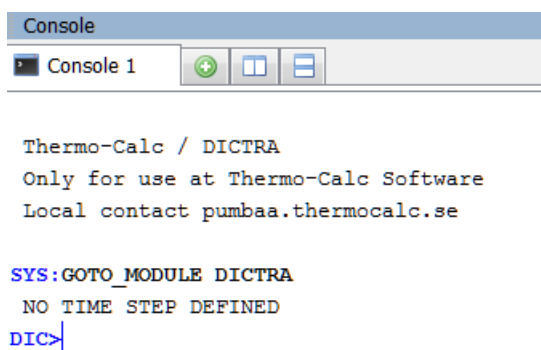
How to Open the DICTRA Module in Thermo-Calc

1. Open Thermo-Calc in Console Mode.



By default, Thermo-Calc opens in the most recently used state. If it opens in Graphical Mode, on the toolbar, click **Switch to Console Mode**.

2. In the `SYS` module, type `GOTO_MODULE DICTRA`. For a list of all the available commands in the current module, at the prompt type a question mark (?) and press <Enter>. For a description of a specific command, type `Help` followed by the name of the command.



```
Thermo-Calc / DICTRA
Only for use at Thermo-Calc Software
Local contact pumbaa.thermocalc.se

SYS:GOTO_MODULE DICTRA
NO TIME STEP DEFINED
DIC>
```

How to Open the Beginner Guide Examples



For the Console Mode examples included with your installation, you can open the .DCM files from the main menu. You can also open any macro file with a command or by dragging it into the Console window.

How to Open a Macro File from the Examples Folder

To open a specific macro file from the Examples folders included with your installation:

1. Open Thermo-Calc.
2. Confirm you are in Console Mode. If not, click the **Switch to Console Mode** button.
3. From the main menu, select **File →** or **Help → Examples Files**. The Console Mode folder opens (these are the Thermo-Calc macro files).
4. Double-click the **Diffusion Module - DICTRA** folder and navigate to an example folder.



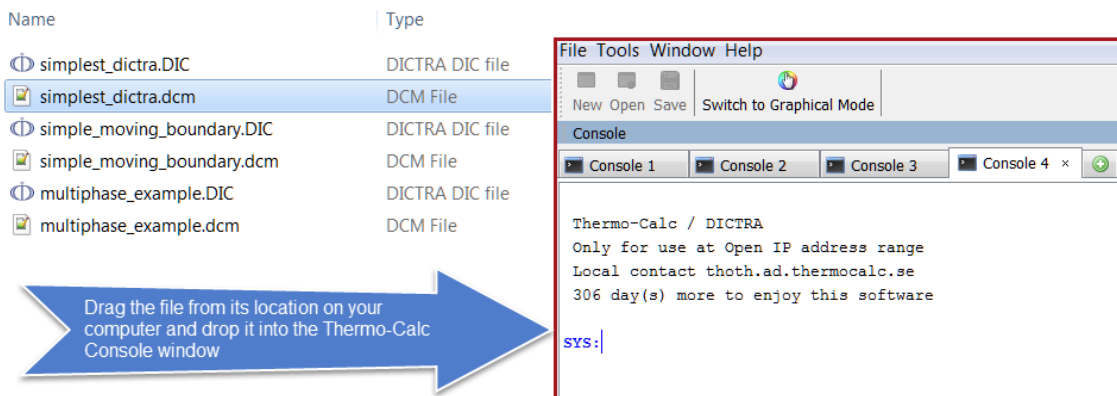
The three Quick Start example macros and the PDF guide are in the first folder called **Beginner Guide**.

How to Use a Command in Console Mode

At the `sys` prompt, type `MACRO_FILE_OPEN` and press <Enter>. In the file dialogue window that opens, locate and select the macro file on your computer.

Drag and Drop a Macro File into Console Mode

For example, navigate to the **Beginner Guide** folder and then drag the file from its location into Thermo-Calc Console window.



The macro automatically runs after you drop it into the Console window.



Basic Terminology

The following definitions are useful to help you understand the examples and commands in this guide. For more detailed information search or browse the online help (**Help → Online Help**).

Cells and Regions in a System

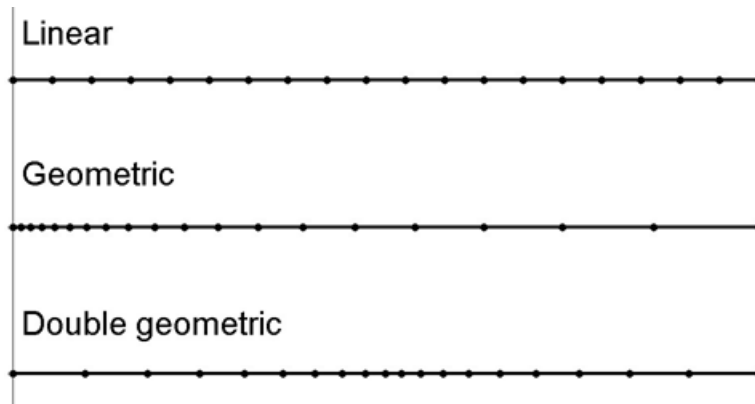
A system normally consists of one *cell*. This cell contains one or several *regions* in which the diffusion problem is to be solved. A system of diffusion equations is solved for each region. The size of the system is the sum of sizes of all the regions in all the system's cells.

Regions and Grids

A *region* contains one or more phases. If a region contains several phases one of those must be entered as a *matrix phase* whereas the others are entered as *spheroid*.

A region must also contain a number of *grid points*. The composition is only known at these grid points and the software assumes that the composition varies linearly between them. The amount and composition of all the phases present at a single grid point in a certain region are those given by thermodynamic equilibrium keeping the over-all composition at the grid point fixed.

These grid types are used in diffusion simulations: linear, geometric and double geometric. When you are setting up the simulation, in either Console Mode or Graphical Mode, a region must have a grid defined.



Phases

Each region contains one or more *phases*. A phase can be of the *matrix*, *spheroid* or *lamellar* type. A phase can be introduced with the status *active* (the default) or *inactive*.

Composition

Two types of composition variables are used within the program: site-fractions and u-fractions, although you can also use weight fraction or mole fraction, for example.

The *site-fractions* are used to set up the problem and interface with POLY-3 (in Console Mode).

The *u-fractions* are used in the diffusion equations.

Geometry and Coordinates

The Diffusion Module (DICTRA) can only handle diffusion problems where composition vary along one spatial coordinate. The geometry of your system can be *planar*, *cylindrical* or *spherical*.

Planar Geometry

This geometry corresponds to an infinitely wide plate of a certain thickness. If the system has a planar geometry then the lower boundary (the zero coordinate) is at the left side of the system. The upper boundary (the coordinate with the highest value) is at the right side of the system.

Boundary Conditions

Boundary conditions are conditions that define how matter behaves at the boundaries of your system. By default, matter is not allowed to cross the system boundaries.

You can change the setting for both the lower boundary (left side/centre) and the upper boundary (right side/surface) of the system.

Setting up a Simulation in Console Mode

Although there are several commands to learn to correctly set up a DICTRA module simulation, the sequence in which to proceed is basically the same:

1. Define and read **thermodynamic** and **kinetic** data.
2. Set **global conditions** (i.e. temperature, T , and pressure, P).
3. Enter **region(s)**.
4. Enter **grid(s)** in region(s).
5. Enter **phase(s)** in region(s).
6. Enter **composition(s)** of phase(s).
7. Set **geometry** (optional – by default, the geometry is planar).
8. Set **boundary conditions** (optional – by default, the boundaries are closed).
9. Set the simulation **time**.
10. **Run** the simulation.

▶ For definitions of the above, see ["Basic Terminology" on page 7](#).

The following sections describe each step in more detail for these simulation types:

- ["About the Single Phase Simulation" on page 11](#)
- ["About the Moving Phase Boundary Simulation" on page 23](#)
- ["About the Multiphase Simulation" on page 32](#)

Single Phase Simulations

In this section:

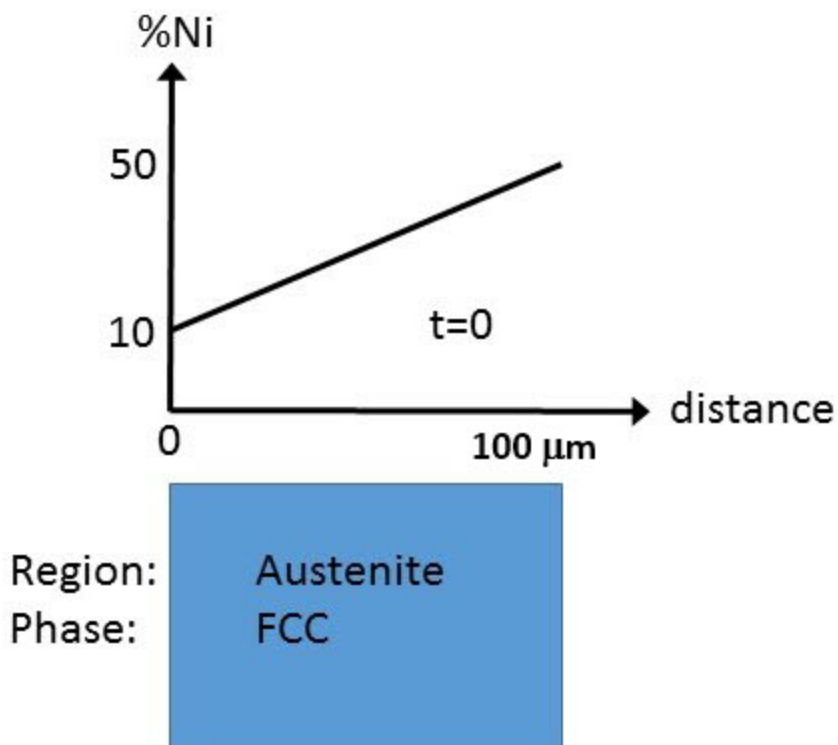
- About the Single Phase Simulation 11
- Theory for the Single Phase Simulation 11
- Running the Single Phase Simulation 12
- Single-Phase Command Details 13

About the Single Phase Simulation

In a 100 μm wide planar domain, the single phase example simulates the diffusion of Fe and Ni at a temperature of 1400 K. At this temperature the material is fully austenitic, i.e. the only phase present is the so-called fcc (face centered cubic) phase (in Thermo-Calc the name of this phase is FCC_A1). Initially, there is a linear variation in Ni going from 10 mass-% on the left-hand side to 50 mass-% on the right-hand side.

The simulation time is slightly different between the modes. For Console Mode it is 10^5 s and for Graphical Mode it is 30 hours.

By default the boundaries are closed. The plot generated shows the initial and final Ni profile.



Schematic view of the initial state of Single Phase Diffusion example. The width of the domain is 100 μm and there is a linear gradient in composition going from 10 to 50 mass-% Ni. There is a single region named Austenite that consists of an fcc phase.

Theory for the Single Phase Simulation

The flux of a component k in the z-direction in an isobarothermal system is in general given by

$$J_k = \sum_{i=1}^n L_{ki} \frac{\partial \mu_i}{\partial z}$$

where L_{ki} is a matrix of kinetic coefficients and μ_i is the chemical potential of component i . The correlation effects, i.e. the coupling of the flux of k component to the chemical potential gradients of the other elements, are normally neglected

$$(L_{ki} = 0, k \neq i)$$

and thus

$$J_k = L_{kk} \frac{\partial \mu_k}{\partial z} = M_k c_k \frac{\partial \mu_k}{\partial z}$$

where c_k is the concentration and M_k the mobility of component k .

The equation for the flux is combined with the equation of continuity, which takes the following form in a planar domain,

$$\frac{\partial c_k}{\partial t} = \frac{\partial}{\partial z} (-J_k)$$

which relates the local evolution of the concentration of k to the divergence of the flux.

The expression for the flux can be expanded in terms of concentration gradients

$$J_k = - \sum_{i=1}^n M_k c_k \frac{\partial \mu_k}{\partial c_j} \frac{\partial c_j}{\partial z} = - \sum_{i=1}^n D_{kj} \frac{\partial c_j}{\partial z}$$

where the diffusion coefficient of component k with respect to the concentration gradient of component j has been introduced.

$$D_{kj} = M_k c_k \frac{\partial \mu_k}{\partial c_j}$$



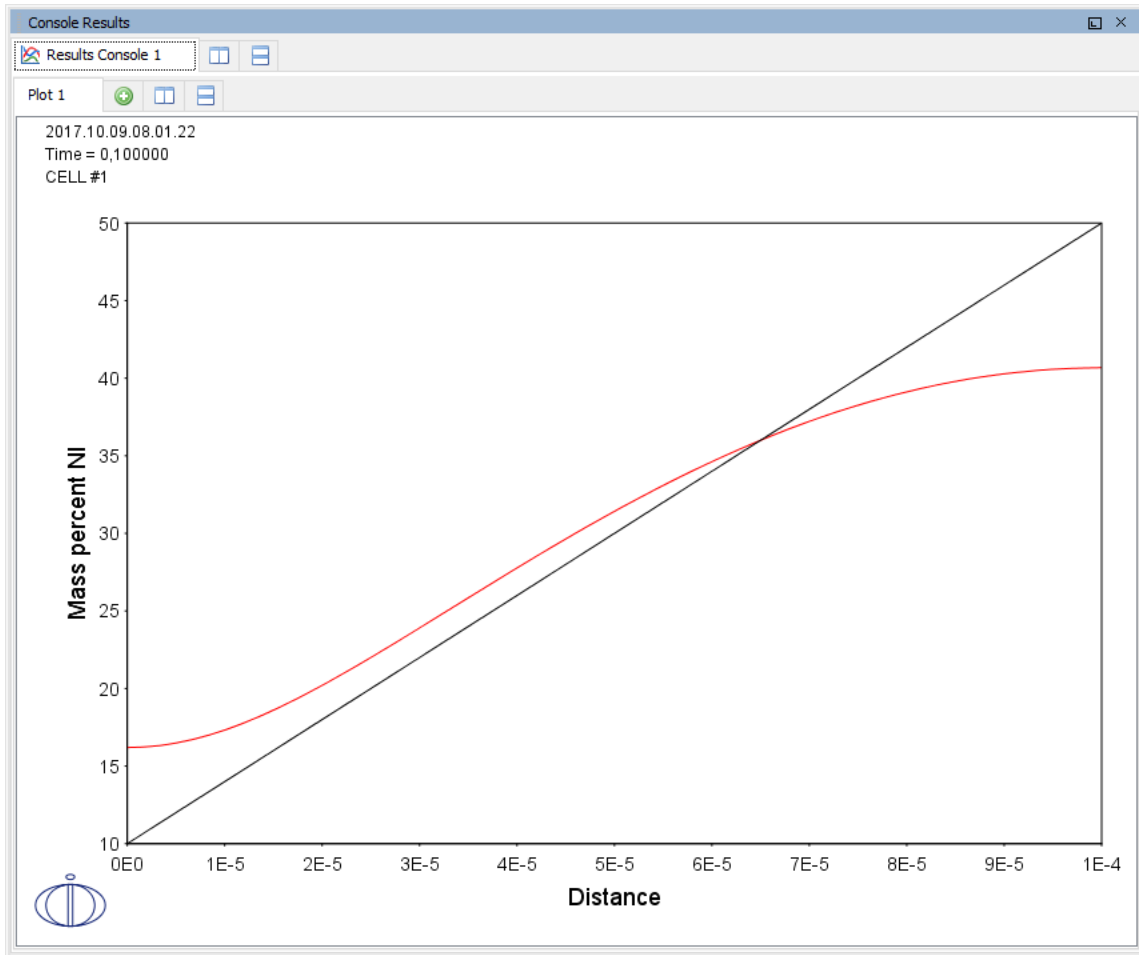
The flux expressions above are given in the so-called lattice-fixed frame of reference. In practical calculations it is more common to use a volume-fixed frame of reference. For a discussion of these concepts, see Andersson and Ågren (1992)¹.

Running the Single Phase Simulation

1. Andersson, Jan-Olof, and John Ågren. 1992. "Models for Numerical Treatment of Multicomponent Diffusion in Simple Phases." *Journal of Applied Physics* 72 (4): 1350–55.

For the single phase simulation, you use an example macro file called `simplest_dictra.dcm`. Open this file in Console Mode as described in ["Opening the Module and Macro Files" on page 4](#).

The simulation should automatically be set up, executed and finished within a matter of seconds. Results are plotted in the **Console Results** plot window.

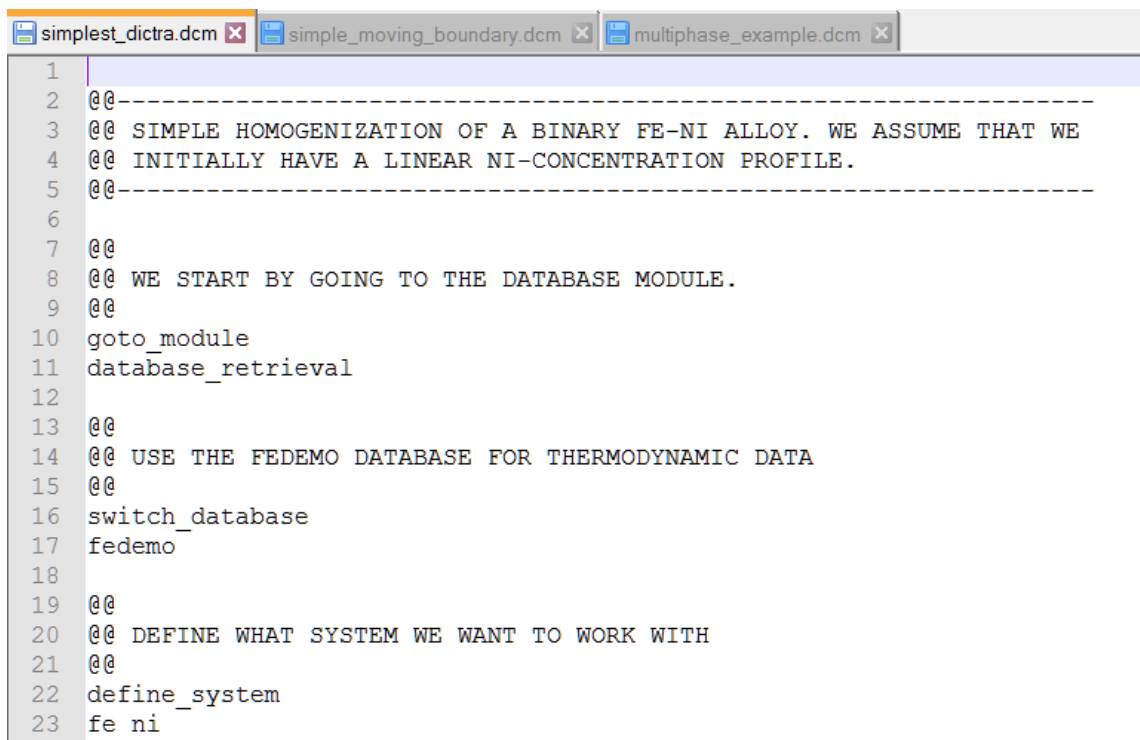


Output from the Single Phase example showing the initial and final Ni profile.

For the rest of the exercise, you will learn how to build this example step-by-step.

Single-Phase Command Details

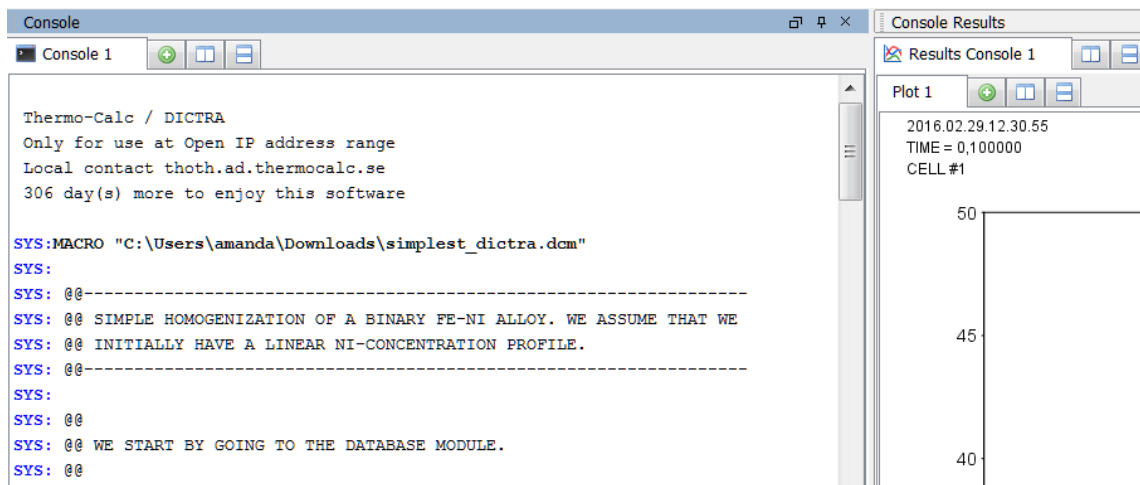
To examine the commands in the macro file, you can either open the file `simplest_dictra.dcm` in a text editor or scroll to the top of the **Console** window.



```

1
2 @@-----
3 @@ SIMPLE HOMOGENIZATION OF A BINARY FE-NI ALLOY. WE ASSUME THAT WE
4 @@ INITIALLY HAVE A LINEAR NI-CONCENTRATION PROFILE.
5 @@-----
6
7 @@
8 @@ WE START BY GOING TO THE DATABASE MODULE.
9 @@
10 goto_module
11 database_retrieval
12
13 @@
14 @@ USE THE FEDEMO DATABASE FOR THERMODYNAMIC DATA
15 @@
16 switch_database
17 fedemo
18
19 @@
20 @@ DEFINE WHAT SYSTEM WE WANT TO WORK WITH
21 @@
22 define_system
23 fe ni

```



In the following sections, the commands are discussed in the order they are executed. The first time a command appears it is listed in its general form in UPPERCASE and then using code font to detail the specific form of the example. For subsequent examples, only the new commands are detailed. All commands may be abbreviated as long as they are unambiguous.

You can also review the comments included in the example files. The first command you will use is shown in the macro file as follows:

In a Text Editor

Comments	@@ @@ WE START BY GOING TO THE DATABASE MODULE. @@
Commands	goto_module database_retrieval

In Thermo-Calc

Comments	SYS: @@ SYS: @@ WE START BY GOING TO THE DATABASE MODULE. SYS: @@
Commands	SYS: goto_module MODULE_NAME: database_retrieval

Single Phase - Specifying the Thermodynamic System

Thermo-Calc Console Mode has several modules used for different purposes. In order to define the thermodynamic system the DATABASE_RETRIEVAL module is used.

```
GOTO_MODULE <MODULE>
goto_module database_retrieval
```

There are many databases for different purposes and material types. In this example the *fedemo* database is used. This is the thermodynamic iron demonstration database.

```
SWITCH_DATABASE <DATABASE>
switch_database fedemo
```

Determine what elements to use in the simulation, in this case Fe and Ni.

```
DEFINE_SYSTEM <LIST OF ELEMENTS>
define_system fe,ni
```

Only the FCC_A1 phase takes part in this simulation and therefore all phases are initially rejected (the wildcard * means *all phases* in this context) and then, with the next command, the FCC_A1 phase is restored.

```
REJECT <ELEMENTS, SPECIES, PHASES, CONSTITUENT OR SYSTEM> <LIST OF THE RELEVANT TYPE>
reject phases *

RESTORE <ELEMENTS, SPECIES, PHASES OR CONSTITUENTS> <LIST OF THE RELEVANT TYPE>
restore phases fcc_a1
```

The thermodynamic system now consists of the elements Fe and Ni and the fcc_a1 phase.

To actually read the data from file into computer memory the GET_DATA command must be executed.

```
GET_DATA
```

```
get_data
```

At this stage only the thermodynamic data is defined and read. The kinetic data (mobilities) are, in this case, stored in another database. This is normally the case though it is possible to have both thermodynamic and kinetic data in a single database.

To add data from a different database the APPEND_DATABASE command is used. The kinetic data in this example is stored in the *mfedemo* database. This is the kinetic iron demonstration database.

```
APPEND_DATABASE <DATABASE>
```

```
append_database mfedemo
```

The same sequence of commands used for thermodynamic data is used to read kinetic data from this database.

```
define_system fe,ni
reject phases *
restore phases fcc_al
get_data
```

Setting up the Single-Phase Simulation

The simulation set-up is performed in the DICTRA_MONITOR module.

```
goto_module
dictra_monitor
```

In this example only one *global* condition is set, temperature. The other global condition that can be set is pressure, but that is rarely done as the default value (1 atm) is set automatically and usually accepted.

```
SET_CONDITION <GLOBAL OR BOUNDARY>
```

```
set_condition global
```

The SET_CONDITION command is immediately followed by these sub-prompts:

```
<T OR P>
```

```
T
```

The T is for temperature. Then follows a standard procedure for entering temporally piece-wise functions:

<LOW TIME LIMIT> <FUNCTION> <HIGH TIME LIMIT> <Y/N> <IF Y, THEN NEXT FUNCTION> <...>

```
0 1400; * N
```

These prompts start with the *low time limit*, here zero (0), then the *function* (here the constant value 1400) closed by a semicolon (;), then the *high time limit* (here a wildcard * is entered), then a Y or a N depending on whether more functions will be entered (in this case N because only a single function is entered).

The wildcard * used for the upper time limit means that the entered function is valid for the whole simulation. What this means is that a constant temperature equal to 1400 K is used during the whole simulation.

In any simulation at least one *region* must be defined. This is a named container that designates a certain part of, in this case the whole, domain. The name of a region is arbitrary and specified by you.

ENTER_REGION <NAME OF REGION>

```
enter_region austenite
```



The use of regions becomes clearer in the context of so-called moving phase boundary simulations (see ["About the Moving Phase Boundary Simulation" on page 23](#)).

A region must contain a numerical *grid* specified by width and type.

ENTER_GRID_COORDINATES

<NAME OF REGION FOR WHICH GRID IS TO BE ENTERED>

<WIDTH OF REGION>

<TYPE OF GRID>

<NUMBER OF GRID POINTS>

```
enter_grid_coordinates
```

```
austenite
```

```
1e-4
```

```
linear
```

```
60
```

In the entries above, a 10^{-4} m wide (1e-4), linear, i.e. equidistant, grid is used with 60 grid points. There are many types of grids, see below for examples, and for other types than the linear there are additional sub-prompts.

A region must contain one *active* phase of type *matrix*. The meaning of active/inactive and phase type is explained in the next examples.

```
ENTER_PHASE_IN_REGION
<ACTIVE OR INACTIVE>
<NAME OF REGION>
<PHASE TYPE IN REGION>
<PHASE NAME>
enter_phase_in_region
active
austenite
matrix
fcc_a1
```

In this example only one phase participates, the `fcc_a1` phase, and it must therefore be *active* and of type *matrix*.

The initial composition profile must be specified for all phases that take part in a simulation. Since there are only two elements, Fe and Ni, in the present simulation the initial *composition profile* must only be given for one of them.

```
ENTER_COMPOSITIONS
<REGION NAME>
<PHASE NAME>
<DEPENDENT COMPONENT>
<COMPOSITION TYPE>
<COMPONENT NAME>
<TYPE OF COMPOSITION PROFILE>
<COMPOSITION PROFILE DEPENDENT INPUT>
enter_compositions
austenite
fcc_a1
fe
weight_percent
ni
```

```
linear
10
50
```

In the above entries, Fe is chosen as the dependent component and the initial profile for Ni is a `linear` variation going from 10 mass-% on the lower/left hand side of the system to 50 mass-% on the upper/right hand side of the system. The composition type can be chosen among a number of different types, e.g. `weight percent` or `mole fraction`. There are also many different ways of specifying the composition profile, e.g. an arbitrary position dependent function or reading it for each grid point from file. Here the simplest possible type is used a `linear` function.

Now you set the simulation time. If not specified the initial time is set to zero.

```
SET_SIMULATION_TIME
<SIMULATION END TIME>
<AUTOMATIC TIMESTEP CONTROL Y/N>
<MAXIMUM TIMESTEP>
<INITIAL SIZE OF TIMESTEP>
<SMALLEST ALLOWED TIMESTEP>
set_simulation_time
1E5
Y
1E4
1E-7
1E-7
```

Here the simulation time is set to 10^5 [s] (`1E5`), the maximum timestep size to 10^4 (`1E4`) and the initial and smallest acceptable to 10^{-7} (both `1E-7`). The *automatic timestep control* should normally always be turned on, which it is by default (you answer `Y` above).

Now you save the workspace:

```
SAVE_WORKSPACES <FILENAME> <OVERWRITE EXISTING FILE Y/N>
save_workspaces simplest_dictra y
```



The command name `SAVE_WORKSPACES` can be misleading. The meaning of this command is roughly set result file. The set-up is saved immediately, but all results during the simulation are also saved to the selected file.

Running and Plotting the Simulation

The following command starts the simulation.

```
SIMULATE_REACTION
simulate_reaction
```

In order to plot results it is necessary to enter the post processor sub-module (note that the GOTO_MODULE command is not used).

```
POST_PROCESSOR
post_processor
```

In most cases, either a specific position in the domain or one or more specific times must be specified with the SET_PLOT_CONDITION command, depending on whether time or spatial position is chosen as independent (x-axis) variable.

```
SET_PLOT_CONDITION <CONDITION WITH ARGUMENTS>
set_plot_condition time 0,1e5
```

Here, the composition profile of Ni at the initial and final time is plotted and thus `time` is chosen as plot condition.

In other cases distance can be given relative to, for example, a phase interface, hence the global specification in the following.

```
SET_DIAGRAM_AXIS <X OR Y> <AXIS QUANTITY>
set_diagram_axis x distance global
```

In this example the only meaningful measure of distance is relative to the domain as a whole, i.e. the leftmost position is zero and the rightmost position corresponds to the width of the region (10^{-4} m).

The dependent property to be plotted is mass-% Ni.

```
set_diagram_axis y weight_percent ni
```

Then plot the diagram:

```
PLOT_DIAGRAM
plot_diagram
```

In the **Console Results** plot window, graphs of the initial and final Ni profile display.

Finally, the SET_INTERACTIVE_MODE command is executed to return control to the user. If the command is absent in a macro the program terminates when the end of the macro file is reached.

```
SET_INTERACTIVE_MODE
```

set_interactive_mode

Moving Phase Boundary Simulations

In this section:

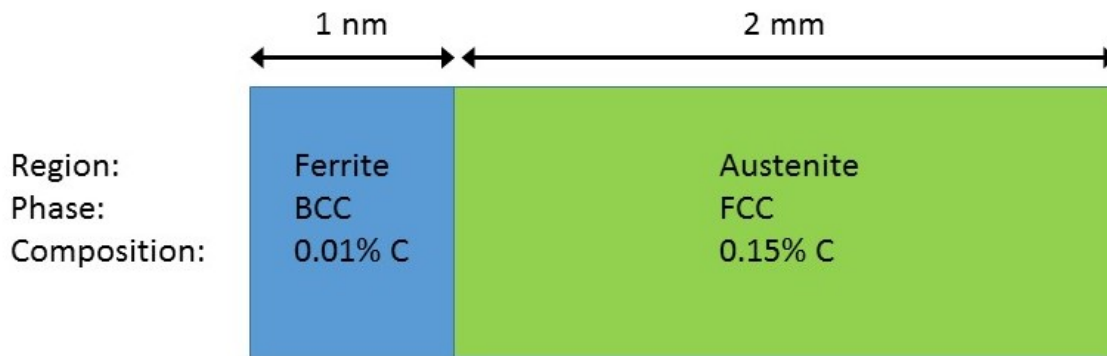
About the Moving Phase Boundary Simulation 23

Theory for the Moving Phase Boundary Simulation 24

Run the Macro File and Learn the Commands 24

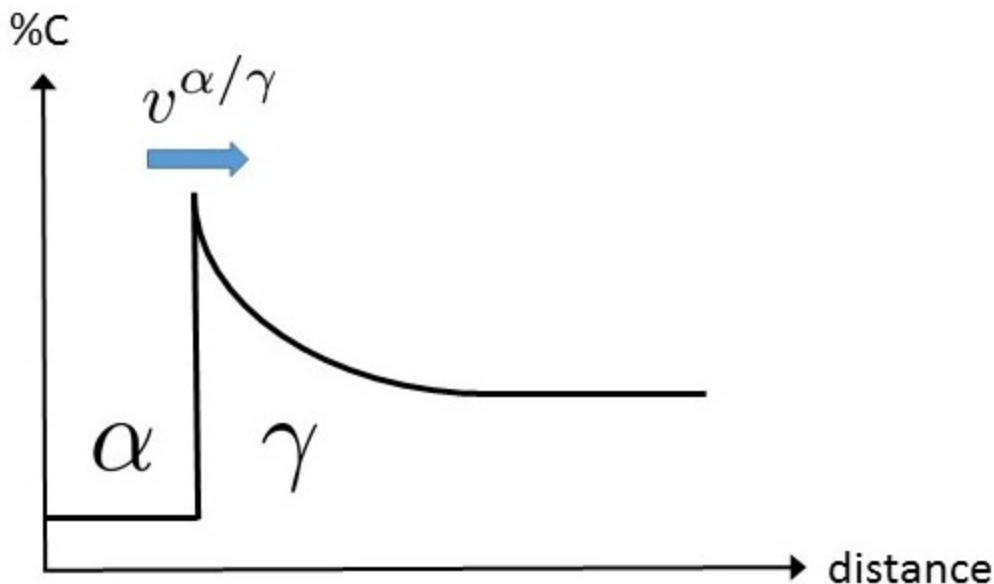
About the Moving Phase Boundary Simulation

The growth of ferrite (bcc) into austenite (fcc) is simulated in this example. The austenite is assumed to be initially homogeneous with the composition Fe - 0.15 mass-% C and the transformation temperature is 1050 K. The initial thickness of the austenite is 2 mm and an initially very thin ferrite (1 nm) is also present at the start of the simulation. The initial state is as below.



The set-up of the Moving Boundary example. There are two regions, Ferrite and Austenite, consisting of bcc and fcc, respectively.

A schematic graph of the carbon profile during the transformation is shown in the image below. Crusius et al. (1992)¹ is recommended as further reading.



1. Crusius, Sabine, Lars Höglund, Ursula Knoop, Gerhard Inden, and John Ågren. 1992. "On the Growth of Ferrite Allotriomorphs in Fe-C Alloys." *Zeitschrift Fur Metallkunde* 83 (10): 729–38.

Schematic view of the carbon profile during the transformation from austenite (γ) to ferrite (α) in the Moving Boundary example.

Theory for the Moving Phase Boundary Simulation

It is assumed that local equilibrium holds at the phase interface, i.e. the chemical potentials of all elements are continuous across the boundary, but the chemical potential gradients are in general discontinuous.

In order to maintain mass balance a set of flux balance equations must be satisfied at the phase interface:

$$v^{\alpha/\gamma}(c_k^{\alpha} - c_k^{\gamma}) = J_k^{\alpha} - J_k^{\gamma} \quad k = 1, \dots, n - 1$$

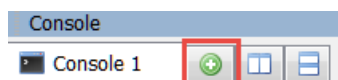
where v is the interfacial velocity. The concentrations and the fluxes c and J are those at the interface on the α and γ sides, respectively.

In the present case there are no degrees of freedom at the interface and the concentrations can be taken directly from a phase diagram. There is only one flux balance equation and the interface velocity can be evaluated once the fluxes at the interface are known. However, for ternary and higher systems the state at the interface must be found by some iterative procedure.

For more information on moving phase boundary problems, see, for example, Ågren (1982)¹, Crusius et al. (1992)² and Höglund (1997)³.

Run the Macro File and Learn the Commands

If the [single phase example](#) has just been run, then either restart the software program or open another Console window by clicking the green button with the plus sign.



At the `SYS` prompt, type `macro` (short for the `MACRO_FILE_OPEN` command) and press `<Enter>`. Locate and select the macro file called `simple_moving_boundary.dcm` as described in ["Opening the Module and Macro Files" on page 4](#). The commands in the macro file set up, run and plot the simulation in the Console Results window.

In the macro file some abbreviations of the commands are used. You can shorten command names in an arbitrary manner as long as the abbreviations are unambiguous.

1. Ågren (1982) "Numerical Treatment of Diffusional Reactions in Multicomponent Alloys," *J. Phys. Chem. Solids*, 43 (4), pp. 385–391.

2. Crusius, Sabine, Gerhard Inden, Ursula Knoop, Lars Höglund, and John Ågren. 1992. "On the Numerical Treatment of Moving Boundary Problems." *Zeitschrift Fur Metallkunde* 83 (9): 673–78.

3. Höglund (1997). *Computer simulation of diffusion controlled transformations in multicomponent alloys*, PhD Thesis, KTH Royal Institute of Technology, Stockholm, Sweden.

In the macro file most commands are first given as comments with the full name and then abbreviated. Commands that were not used in the single-phase simulation, or are used differently, are explained in more detail.



Comments in the macro file can be entered after two @@ symbols. In most cases the arguments of a command can be entered on the same line as the command itself.

► ["Single-Phase Command Details" on page 13](#)

Moving Phase - Specifying the Thermodynamic System

This set of commands is the same as in the [single-phase simulation](#) example. Thermodynamic and kinetic data is read for Fe and C and the bcc and fcc phases.

```
@@ goto_module database_retrieval
go da
@@ switch_database fedemo
sw fedemo
@@ define_system fe,c
def-sys fe c
@@ reject phase *
rej ph *
@@ restore phase fcc_a1,bcc_a2
rest ph fcc,bcc
@@ get_data
get
@@ append_database mfedemo
app mfedemo
@@ define_system fe,c
def-sys fe c
@@ reject phase *
rej ph *
@@ restore phase fcc_a1,bcc_a2
rest ph fcc,bcc
@@ get_data
Get
```

Setting Up the Moving Phase Boundary Simulation

The first region to be entered is entered in the same way as in the [single-phase simulation](#).

```
@@ goto_module dictra_monitor
go d-m
@@ set_condition global T 0 1050; * N
set-cond glob T 0 1050; * N
```

A constant temperature equal to 1050K is used in this simulation.

```
@@ enter_region ferrite
ent-reg ferrite
```

The phase interface between ferrite and austenite is created by introducing a region called austenite that is attached to the ferrite region. Regions are always separated by phase interfaces and must therefore always contain different *matrix* type phases.

```
ENTER_REGION
<NAME OF REGION>
<NAME OF EXISTING REGION TO ATTACH TO>
<ATTACH TO THE RIGHT OF EXISTING REGION Y/N>
enter_region
austenite
ferrite
y
```

The domain now consists of two regions: *ferrite* and *austenite*.

As in the single-phase simulation example, the regions must contain both grids and one *active* phase of type *matrix*. These are entered separately for both regions.

```
enter-grid
ferrite
1e-9
linear
10
enter-grid
austenite
2e-3
geometric
50
1.05
```

In the above entries, the grid type for the `ferrite` region is the same as in the single-phase simulation example, i.e. equidistant. For the `austenite` region the grid type is called `geometric` and for these grids the grid point spacing changes by a constant factor between every grid point. In this case the spacing increases by 5%, i.e. a factor `1.05`, from the lower/left side for each grid point.

Now the initial composition of both phases is entered. In this case Fe is automatically chosen to be the dependent component.

```
enter-phase
active
ferrite
matrix
bcc
enter-phase
active
austenite
matrix
fcc
```

Both phases are assumed to be initially homogeneous, the bcc phase having `0.01` mass-% C and the fcc phase `0.15` mass-% C, as shown in the following:

```
enter-composition
ferrite
bcc
w-p
C
linear
0.01
0.01
enter-composition
austenite
fcc
w-p
C
linear
0.15
0.15
```

Set the simulation time and other parameters related to the time-step size.

```
set-simulation-time
1e5
Y
1e4
1E-7
1E-7
```

Finally, save the simulation set-up to file.

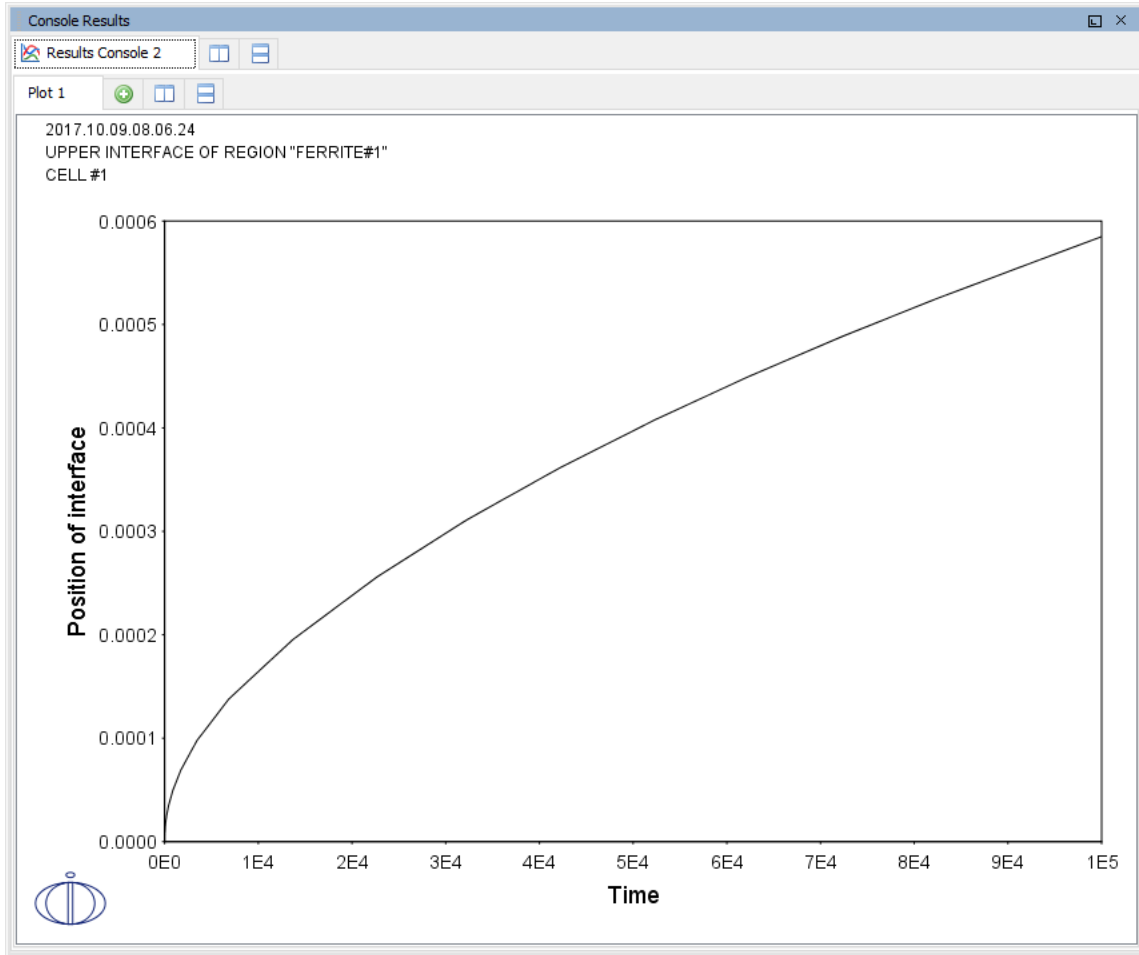
```
save simple_moving_boundary yes
```

Running and Plotting the Moving Phase Simulation

These commands are the same as in the [single-phase example](#).

```
@@simulate_reaction
sim
@@ post_processor
post
```

The ferrite/austenite phase interface position as a function of time is plotted.



Output from the example `simple_moving_boundary` showing the phase interface position as a function of time.

As mentioned in the single phase example, in general either a specific spatial position or a specific time must be set as a plot condition. Here the phase `interface` is set as plot condition and it can be referenced as the lower interface of the austenite region or as the upper interface of the `ferrite` region.

```
SET_PLOT_CONDITION
```

```
<TYPE OF PLOT CONDITION>
```

```
<FOR INTERFACE CONDITION, SPECIFY REGION>
```

```
<LOWER OR UPPER INTERFACE OF REGION>
```

```
set-plot-condition
```

```
interface
```

```
ferrite
```

```
upper
```

Set time as x-axis variable

```
@@set-diagram-axis x time  
s-d-a x time
```

Now set the interface position as y-axis variable

```
s-d-a y  
  
POSITION_OF_INTERFACE  
  
<SPECIFY REGION>  
  
<LOWER OR UPPER INTERFACE OF REGION>  
  
pos-of-int ferrite upper  
  
@@ plot_diagram  
  
plot
```

The plot has the parabolic appearance characteristic for many types of diffusion simulations.

Multiphase Simulations

In this section:

About the Multiphase Simulation32

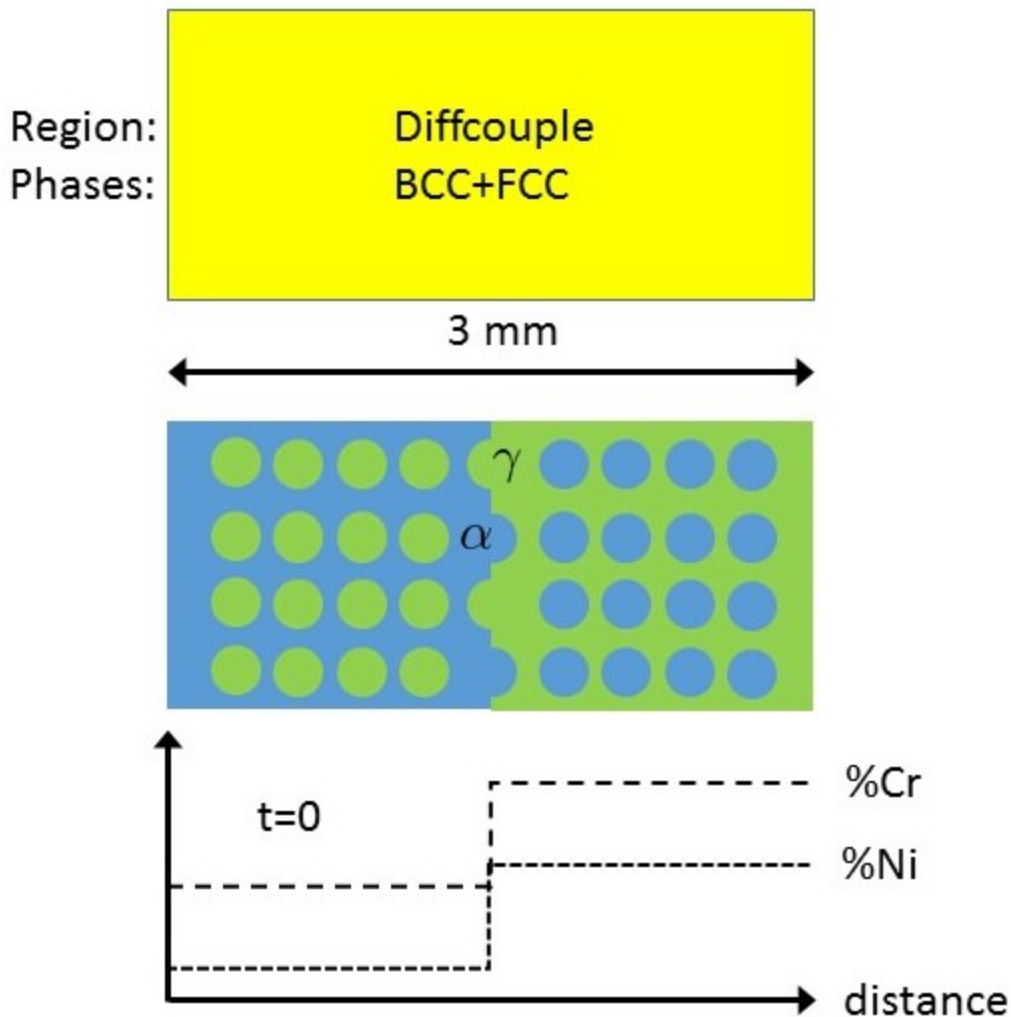
Theory for the Multiphase Simulation33

Run the Macro File and Learn the Commands33

About the Multiphase Simulation

This example simulates the evolution of an Fe-Cr-Ni diffusion couple during a 100 hour heat treatment at 1100° C (1373.15 K). Both end members of the diffusion couple are duplex ferrite plus austenite, but the majority phase is ferrite on the left-hand side and austenite on the right.

With this type of simulation it is assumed that the material is fully equilibrated at each grid point, i.e. the local phase fractions, phase compositions and so forth are obtained from an equilibrium calculation with the local overall composition as a condition.



Schematic view of the initial state of the Multiphase example. Two ternary Fe-Cr-Ni alloys form a diffusion couple. There is a single region (diffcouple) where both the bcc and fcc phases are entered. Both alloys are duplex bcc+fcc, but the majority phase is bcc (α) in the left-hand side alloy and fcc (γ) in the right.

Theory for the Multiphase Simulation

More details about the homogenization model for multiphase simulations can be found in Larsson and Engström (2006)¹ and Larsson and Höglund (2009)². As mentioned in the description of this simulation, it is assumed that the material is locally fully equilibrated and that the local phase fractions, phase compositions and so forth are obtained from an equilibrium calculation with the local overall composition as a condition. From a numerical point of view the homogenization model treats the multiphase material as a single phase having the “average”, or “effective”, properties of the local phase mixture.

When estimating the effective kinetics of a multiphase mixture the product of solubility and mobility in each phase is considered. It is therefore convenient to define

$$\Gamma_k^\phi = M_k^\phi c_k^\phi$$

for each phase ϕ . The effective kinetics of the multiphase mixture is denoted and in this example it was evaluated using the lower Hashin-Shtrikman bound. The default method is a simple rule of mixtures

$$\Gamma^* = \sum_{\phi} f^{\phi} \Gamma_k^{\phi}$$

where f^{ϕ} is the volume fraction of ϕ . The flux is obtained as

$$J_k = -\Gamma_k^* \frac{\partial \mu_k}{\partial z}$$

which replaces the single-phase equation

$$J_k = L_{kk} \frac{\partial \mu_k}{\partial z} = M_k c_k \frac{\partial \mu_k}{\partial z}$$

for multiphase simulations.

Run the Macro File and Learn the Commands

1. Larsson, Henrik, and Anders Engström. 2006. “A Homogenization Approach to Diffusion Simulations Applied to $\alpha + \gamma$ Fe–Cr–Ni Diffusion Couples.” *Acta Materialia* 54 (9): 2431–39.

2. Larsson, Henrik, and Lars Höglund. 2009. “Multiphase Diffusion Simulations in 1D Using the DICTRA Homogenization Model.” *Calphad* 33 (3): 495–501.

At the `SYS` prompt, type `macro` and press <Enter>. Locate and select the macro file called `multiphase_example.dcm` as described in ["Opening the Module and Macro Files" on page 4](#). The commands in the macro file should set up the simulation, run it and then produce a graph in the Console Results window.

Most of the set-up is the same as the [single phase](#) and [moving phase boundary](#) examples. Detailed comments are only given for the new commands for this simulation.

► ["Single-Phase Command Details" on page 13](#)

Specifying the Multiphase Thermodynamic System

Data for Fe, Cr and Ni and the fcc and bcc phases are read from the *fedemo* (thermodynamics) and *mfedemo* (kinetics) databases.

```
go da
sw fedemo
def-sys fe cr ni
rej ph *
rest ph bcc, fcc
get
app mfedemo
def-sys fe cr ni
rej ph *
rest ph bcc, fcc
get
```

Setting Up the Multiphase Simulation

Set the temperature `T` during the simulation to 1100° C (1373.15 K) and enter a region `diffcouple`.

```
go dictra-monitor
set-cond glob T 0 1373.15; * N
ent-reg diffcouple
ent-grid
diffcouple
3e-3
double
60
0.85
1.15
```

In the above example a so-called `double` geometric grid is used. This is similar to the geometric grid used in the moving phase boundary example, but instead two geometrical factors are entered, one for the lower and one for the upper half of the region. In the lower half of the region the distance between subsequent grid points decreases by 15 % (`0.85`) and in the upper half it increases by 15 % (`1.15`). The resulting grid has a much larger grid point density toward the middle of the domain.

As always, a region must contain exactly one `active` phase that is of type `matrix`. In this example it is arbitrary whether it is the `bcc` or the `fcc` phase that is entered as matrix.

```
enter-phase
active
diffcouple
matrix
fcc
```

The `bcc` phase is entered in the same manner as the `fcc` phase, except that the type is set as `spheroid`.

```
enter-phase
active
diffcouple
spheroid
bcc
```

The initial composition profile of the `fcc` phase is entered as a `function` of distance.

```
enter-composition
diffcouple
fcc
fe
m-f
cr
function
2.57e-1+(4.23e-1-2.57e-1)*hs(x-1.5e-3);
ni
function
6.47e-2+(2.75e-1-6.47e-2)*hs(x-1.5e-3);
```

In the above entries, the `hs` denotes the Heaviside step function that has the following properties:

$$hs(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

The resulting Cr profile is thus 0.257 on the left half of the domain and 0.423 on the right with a sharp step at the center.

► As shown in ["About the Multiphase Simulation" on page 32](#).

The initial composition of the fcc phase is actually the initial overall composition. This is due to the choice of initial composition of the spheroid bcc phase.

```
ENTER_COMPOSITIONS
<REGION NAME>
<PHASE NAME>
<USE EQUILIBRIUM VALUE Y/N>
enter-comp
diffcouple
bcc
y
```

When entering the composition for a type spheroid phase it is possible to either, as here, let the composition of the matrix phase be the overall composition or manually enter a start composition and volume fraction.

The local kinetics of the multiphase mixture must be estimated by means of some function that may depend on the local phase fractions, phase compositions and the mobilities of the individual phases. The command for choosing the function to use is ENTER_HOMOGENIZATION_FUNCTION. There are many such homogenization functions to choose from. In this particular example the so-called lower Hashin–Shtrikman bound¹ is a good choice.

```
ENTER_HOMOGENIZATION_FUNCTION
<ENTER FUNCTION NUMBER>
enter-homo-fun
1
```

Finally, set simulation time (100 h) and select the name of the result file:

```
set-sim-time
3.6e5
```

¹ Z. Hashin and S. Shtrikman (1962), *J. Appl. Phys.*, 33 (10), pp. 3125–3131.

```
yes
3.6e4
1e-7
1e-7
save multiphase_example y
```

Running and Plotting the Multiphase Simulation Results

Run the simulation.

```
simulate
```

Go to the post processor.

```
post
```

Plot the mole fraction fcc phase as a function of distance at the end of the heat treatment.

```
set-plot-condition time last
set-diagram-axis x distance global
set-diagram-axis y npm(fcc)
```

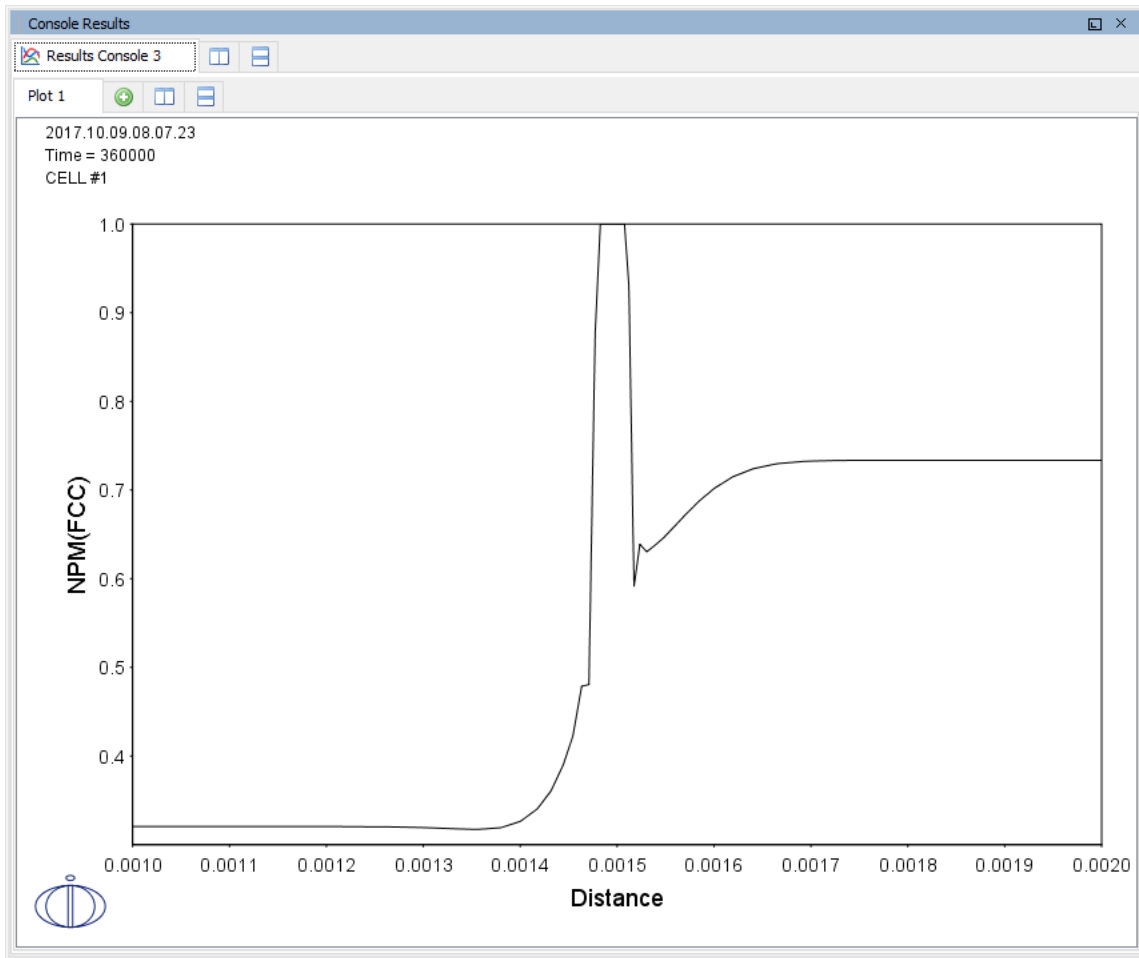
Instead of explicitly entering the final time it is possible to use the keyword `last`. The `np` of `npm` stands for *number of moles* of the phase argument and the `m` is the normalizing quantity (moles). The limits of an axis can be set with the `SET_SCALING_STATUS` command. You manually set the scaling of the axis and then restore automatic scaling:

```
SET_SCALING_STATUS
<X OR Y AXIS>
<USE AUTOMATIC SCALING Y/N>
<IF MANUAL SCALING, ENTER LIMITS>
set-scaling-status x n 1e-3 2e-3
```

Plot the diagram.

```
plot
```

The plot below shows that a single phase fcc zone has formed and this was also observed experimentally, see A. Engström, 1995: "Interdiffusion in multiphase, Fe-Cr-Ni diffusion couples", *Scand. J. Metall.*, 24 (1), pp. 12–20.



Output from the *multiphase_example* showing the mole fraction fcc phase as a function of distance at the end of the heat treatment.

References

1. J.-O. Andersson, T. Helander, L. Höglund, P. Shi, and B. Sundman, "Thermo-Calc and DICTRA, computational tools for materials science," *Calphad*, vol. 26, no. 2, pp. 273–312, Jun. 2002.
2. A. Borgenstam, L. Höglund, J. Ågren, and A. Engström, "DICTRA, a tool for simulation of diffusional transformations in alloys," *J. Phase Equilibria*, vol. 21, no. 3, pp. 269–280, May 2000.
3. J.-O. Andersson and J. Ågren, "Models for numerical treatment of multicomponent diffusion in simple phases," *J. Appl. Phys.*, vol. 72, no. 4, pp. 1350–1355, Aug. 1992.
4. S. Crusius, L. Höglund, U. Knoop, G. Inden, and J. Ågren, "On the growth of ferrite allotriomorphs in Fe-C alloys," *Zeitschrift für Met.*, vol. 83, no. 10, pp. 729–738, 1992.
5. J. Ågren, "Numerical Treatment of Diffusional Reactions in Multicomponent Alloys," *J. Phys. Chem. Solids*, vol. 43, no. 4, pp. 385–391, 1982.
6. S. Crusius, G. Inden, U. Knoop, L. Höglund, and J. Ågren, "On the numerical treatment of moving boundary problems," *Zeitschrift für Met.*, vol. 83, no. 9, pp. 673–678, 1992.
7. L. Höglund, *Computer simulation of diffusion controlled transformations in multicomponent alloys*, PhD Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 1997.
8. Z. Hashin and S. Shtrikman, "A Variational Approach to the Theory of the Effective Magnetic Permeability of Multiphase Materials," *J. Appl. Phys.*, vol. 33, no. 10, pp. 3125–3131, 1962.
9. A. Engström, "Interdiffusion in multiphase, Fe-Cr-Ni diffusion couples," *Scand. J. Metall.*, vol. 24, no. 1, pp. 12–20, 1995.
10. H. Larsson and A. Engström, "A homogenization approach to diffusion simulations applied to $\alpha + \gamma$ Fe-Cr-Ni diffusion couples," *Acta Mater.*, vol. 54, no. 9, pp. 2431–2439, May 2006.
11. H. Larsson and L. Höglund, "Multiphase diffusion simulations in 1D using the DICTRA homogenization model," *Calphad Comput. Coupling Phase Diagrams Thermochem.*, vol. 33, no. 3, pp. 495–501, 2009.

Next Steps

Below are suggestions for you to learn more about using the Diffusion (DICTRA) Module.

- Browse the Online Help (Help → Online Help) or open one of the available PDFs that is included with your installation (Help → Manuals Folder).
- You have been working with in Console Mode. You might also want to try the *Diffusion Module Graphical Mode Quick Start Guide* (Help → Manuals Folder).
- Run other Console Mode examples (Help → Examples Files), which contain many simulation types and advanced functionality. It is a good starting point when you are thinking of creating a new simulation. When you are setting up a new simulation there are two recommended ways to proceed:
 - Copy and gradually adjust an existing macro file from the examples collection.
 - Enter the set-up directly while recording a log file. The command for recording a log file is SET_LOG_FILE and this should be done immediately when starting the program in the SYS module. The log file can then be directly used as a macro file (it has the file extension *.TCM).
- You can also go to the Thermo-Calc website to see if there are other options suited to you, such as a course or video tutorials. (Help → Video Tutorials and Help → Thermo-Calc Website).