ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Seoul National University
College of Engineering
Dept. of Materials Science and Engineering

FactSage™

FactSage Macro Processing

A tool for process simulations

M.-A. Van Ende and I.-H. Jung

October 2018 (FactSage 7.2)

Contents

- Concept of FactSage macro processing
- Creating and running a macro file
- Macro variables and functions
- Macro commands
- Application to desulfurization of hot metal

FactSage Macro Processing??

- Repetitive calculations → automate them with a macro
- Enables to run automatically Equilib files as background job
- Write each step as a series of commands and functions stored in a macro file (Visual basic)
- Macro processor will repeat the commands:

  Load Equilib file → calculate the equilibrium → save results → go to next Equilib files

Useful to treat large and lengthy calculations

CREATING AND RUNNING A MACRO FILE

Before creating a macro file… (1/4)

1. **Create Equilib files**
2. **Save them in a directory**

Example 1 Equi1.dat

400 g Fe_2O_3

400 g CaO

200 g MgO

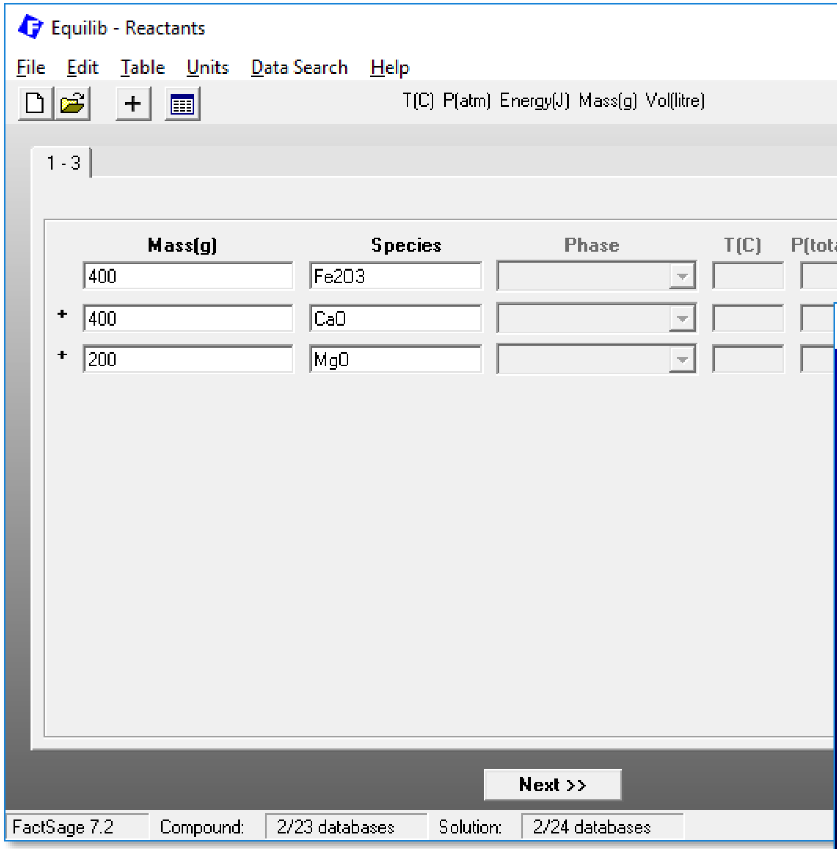
Databases: FactPS and FToxid

T = 1500 °C

Save in C:\Example\

Before creating a macro file… (2/4)

Setup the reactants and databases

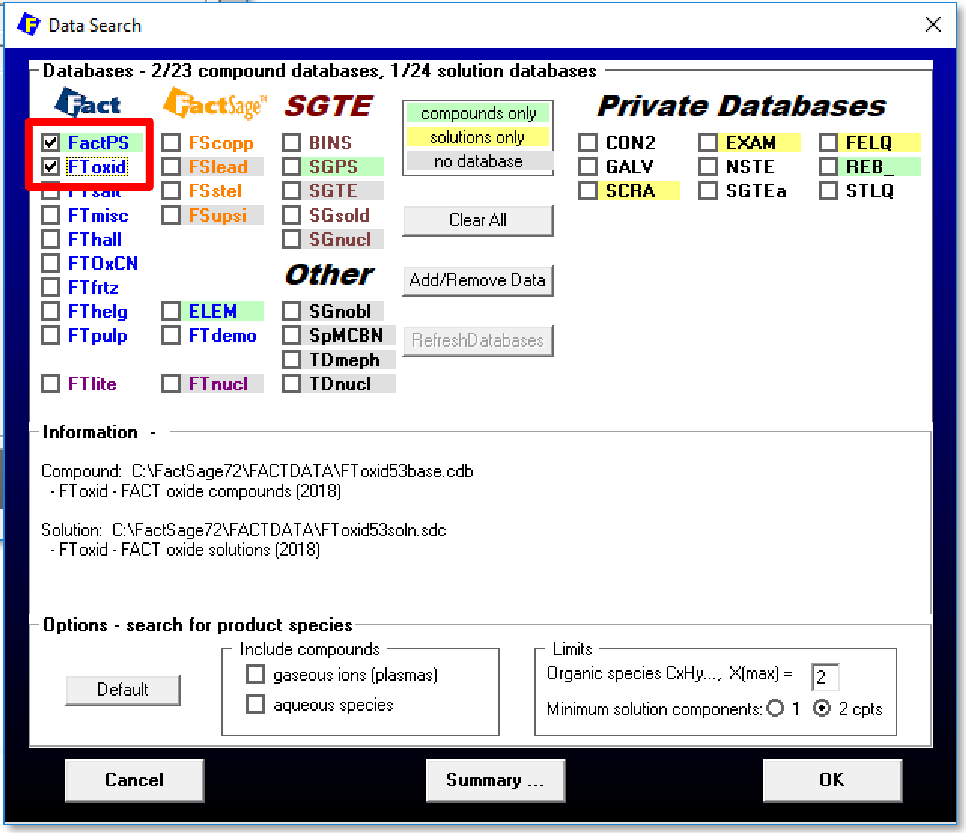


Equilib - Reactants

File Edit Table Units Data Search Help
T(C) P(atm) Energy(J) Mass(g) Vol(litre)
1 - 3
Next >>

	Mass(g)	Species	Phase	T(C)	P(total)**	Stream#	Data
	400	Fe2O3				1	
+	400	CaO					
+	200	MgO					

FactSage 7.2 | Compound: 2/23 databases | Solution: 2/24 databases



Data Search

Databases - 2/23 compound databases, 1/24 solution databases

☑ FactPS
☑ FToxid
☐ FTsalt
☐ FTmisc
☐ FThall
☐ FTOxCN
☐ FTfrtz
☐ FThelg
☐ FTpulp
☐ FTlite

☐ FScopp
☐ FSlead
☐ FSstel
☐ FSupsi
☐ ELEM
☐ FTdemo
☐ FTnucl

☐ BINS
☐ SGPS
☐ SGTE
☐ SGsold
☐ SGnucl
☐ SGnobl
☐ SpMCBN
☐ TDmeph
☐ TDnucl

☐ CON2
☐ GALV
☐ SCRA
☐ EXAM
☐ NSTE
☐ SGTEa
☐ FELQ
☐ REB_
☐ STLQ

Information -

Compound: C:\FactSage72\FACTDATA\FToxid53base.cdb
- FToxid - FACT oxide compounds (2018)

Solution: C:\FactSage72\FACTDATA\FToxid53soln.sdc
- FToxid - FACT oxide solutions (2018)

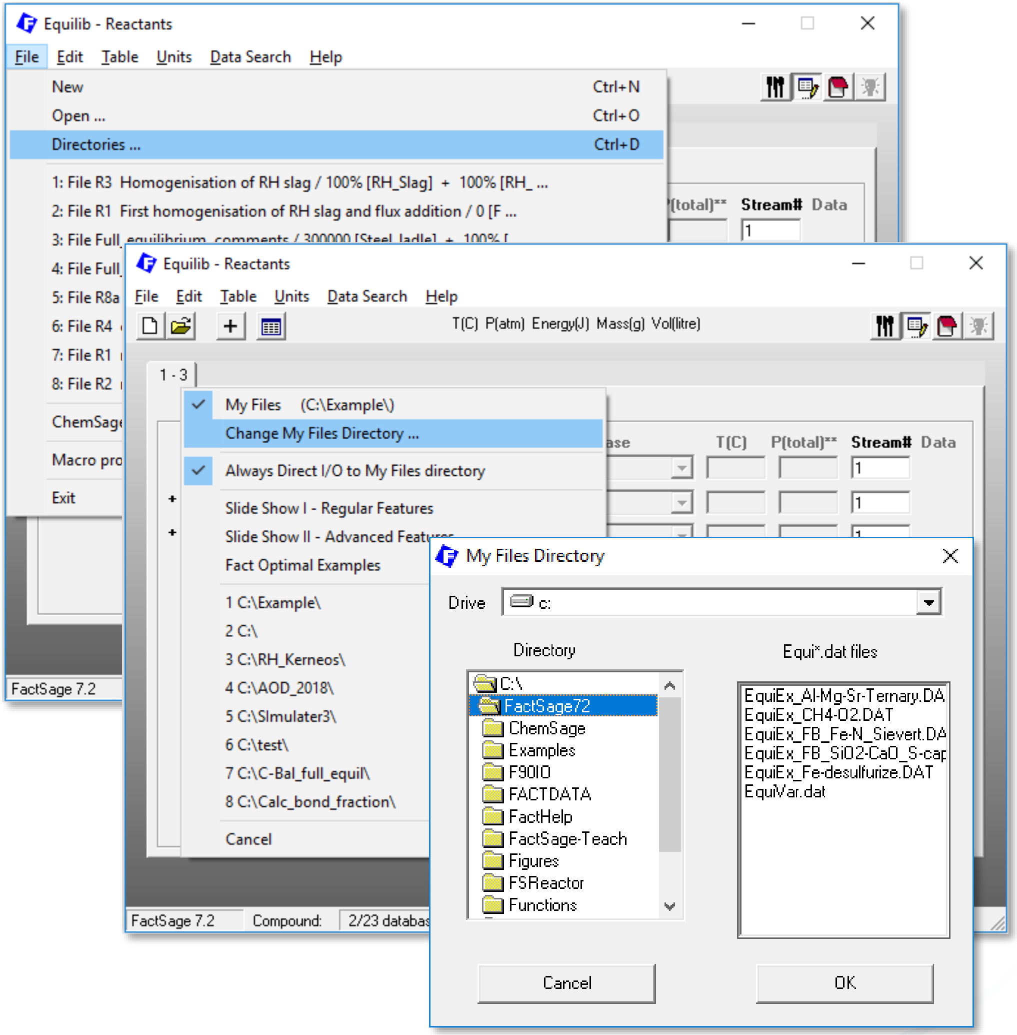
Options - search for product species

Include compounds
☐ gaseous ions (plasmas)
☐ aqueous species

Limits
Organic species CxHy..., X(max) = 2
Minimum solution components: ○ 1 ● 2 cpts

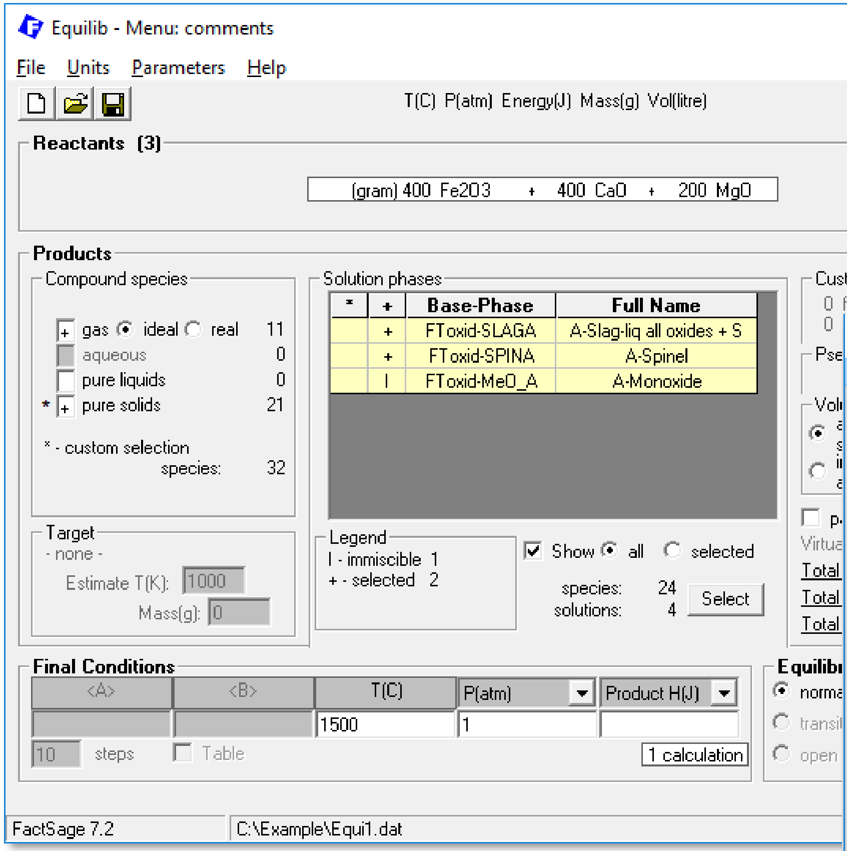
Before creating a macro file… (3/4)

- FactSage macro processing looks for Equilib and stream files in the defined *My Files* directory
- The default *My Files* directory when installing FactSage is C:\FactSage\


Equilib - Reactants
File Edit Table Units Data Search Help
New Ctrl+N
Open ... Ctrl+O
Directories ... Ctrl+D
1: File R3 Homogenisation of RH slag / 100% [RH_Slag] + 100% [RH_ ...
2: File R1 First homogenisation of RH slag and flux addition / 0 [F ...
3: File Full equilibrium, comments / 300000 [Steel ladle] + 100% [
4: File Full
5: File R8a
6: File R4
7: File R1
8: File R2
ChemSage
Macro pro
Exit
FactSage 7.2
(total)** Stream# Data
1
Equilib - Reactants
File Edit Table Units Data Search Help
T(C) P(atm) Energy(J) Mass(g) Vol(litre)
1 - 3
My Files (C:\Example\)
Change My Files Directory ...
Always Direct I/O to My Files directory
Slide Show I - Regular Features
Slide Show II - Advanced Features
Fact Optimal Examples
1 C:\Example\
2 C:\
3 C:\RH_Kerneos\
4 C:\AOD_2018\
5 C:\Slmulater3\
6 C:\test\
7 C:\C-Bal_full_equil\
8 C:\Calc_bond_fraction\
Cancel
ase T(C) P(total)** Stream# Data
1
1
1
FactSage 7.2 Compound: 2/23 databas
My Files Directory
Drive c:
Directory
C:\
FactSage72
ChemSage
Examples
F90IO
FACTDATA
FactHelp
FactSage-Teach
Figures
FSReactor
Functions
Equi*.dat files
EquiEx_Al-Mg-Sr-Ternary.DA
EquiEx_CH4-O2.DAT
EquiEx_FB_Fe-N_Sievert.DA
EquiEx_FB_SiO2-CaO_S-cap
EquiEx_Fe-desulfurize.DAT
EquiVar.dat
Cancel
OK

Before creating a macro file… (4/4)

Select the compound and solution species, T and P…
Save the Equilib file



Equilib - Menu: comments

T(C) P(atm) Energy(J) Mass(g) Vol(litre)

Reactants (3)

(gram) 400 Fe2O3 + 400 CaO + 200 MgO

Products

Compound species

+	gas ● ideal ○ real	11
	aqueous	0
	pure liquids	0
* +	pure solids	21

* - custom selection species: 32

Solution phases

*	+	Base-Phase	Full Name
	+	FToxid-SLAGA	A-Slag-liq all oxides + S
	+	FToxid-SPINA	A-Spinel
	I	FToxid-MeO_A	A-Monoxide

Target - none -
Estimate T(K): 1000
Mass(g): 0

Legend
I - immiscible 1
+ - selected 2

☑ Show ● all ○ selected
species: 24
solutions: 4
Select

Custom Solutions
0 fixed activities Details ...

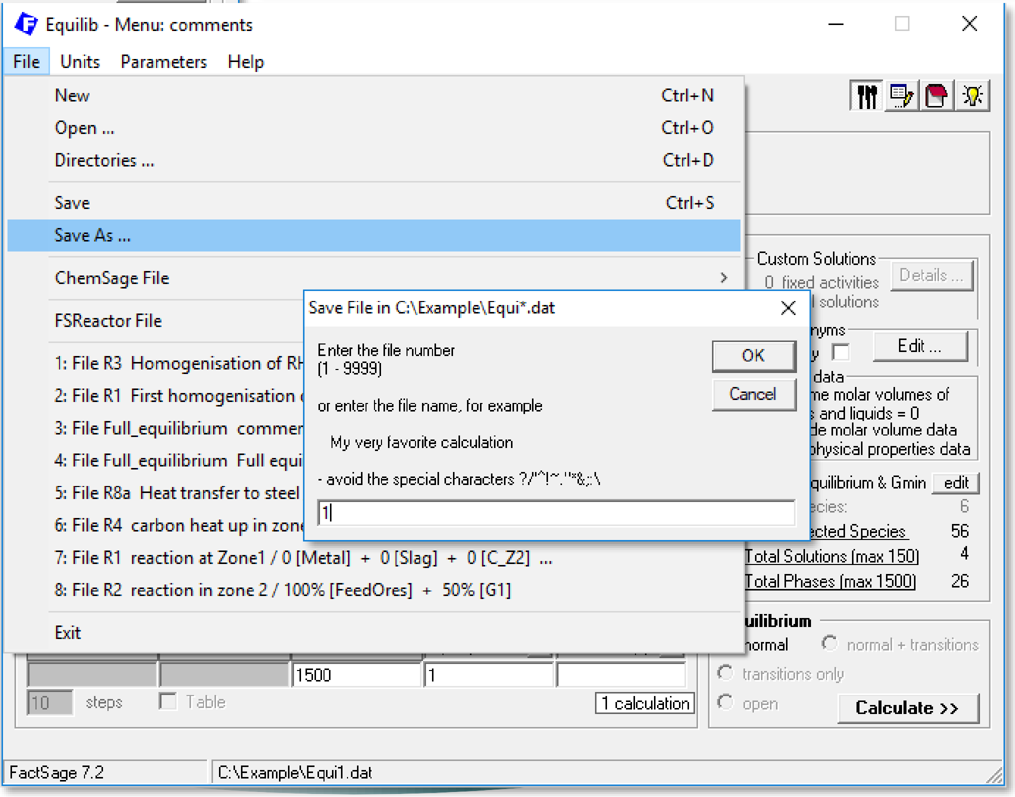
Final Conditions

<A>	<B>	T(C)	P(atm)	Product H(J)
		1500	1	

10 steps □ Table 1 calculation

Equilib
● norma
○ transit
○ open

FactSage 7.2 C:\Example\Equi1.dat



Equilib - Menu: comments

1: File R3 Homogenisation of RH
2: File R1 First homogenisation
3: File Full_equilibrium commen
4: File Full_equilibrium Full equi
5: File R8a Heat transfer to steel
6: File R4 carbon heat up in zone
7: File R1 reaction at Zone1 / 0 [Metal] + 0 [Slag] + 0 [C_Z2] ...
8: File R2 reaction in zone 2 / 100% [FeedOres] + 50% [G1]

Save File in C:\Example\Equi*.dat

Enter the file number
(1 - 9999)

or enter the file name, for example

My very favorite calculation

- avoid the special characters ?/"^!~.'"*&;:\

1

OK Cancel

Custom Solutions
0 fixed activities Details ...
solutions
Edit ...
data
me molar volumes of
and liquids = 0
de molar volume data
physical properties data
quilibrium & Gmin edit
ecies: 6
ected Species 56
Total Solutions (max 150) 4
Total Phases (max 1500) 26

uilibrium
normal ○ normal + transitions
○ transitions only
○ open
Calculate >>

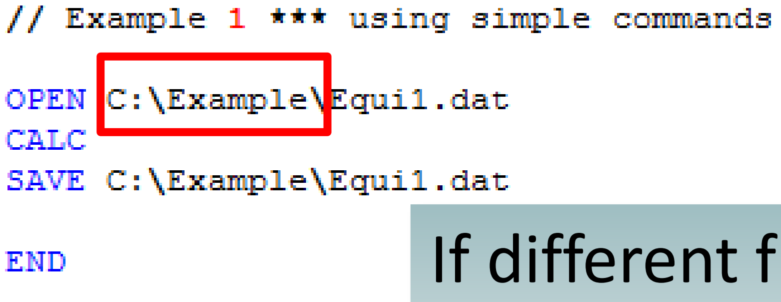
1500 1
10 steps □ Table 1 calculation

FactSage 7.2 C:\Example\Equi1.dat

Creating a (very) simple macro file

- ***.mac** file, editable with text editor (Notepad, Notepad++)
- 3 principal commands:
  - OPEN
  - CALC
  - SAVE

Example 1:

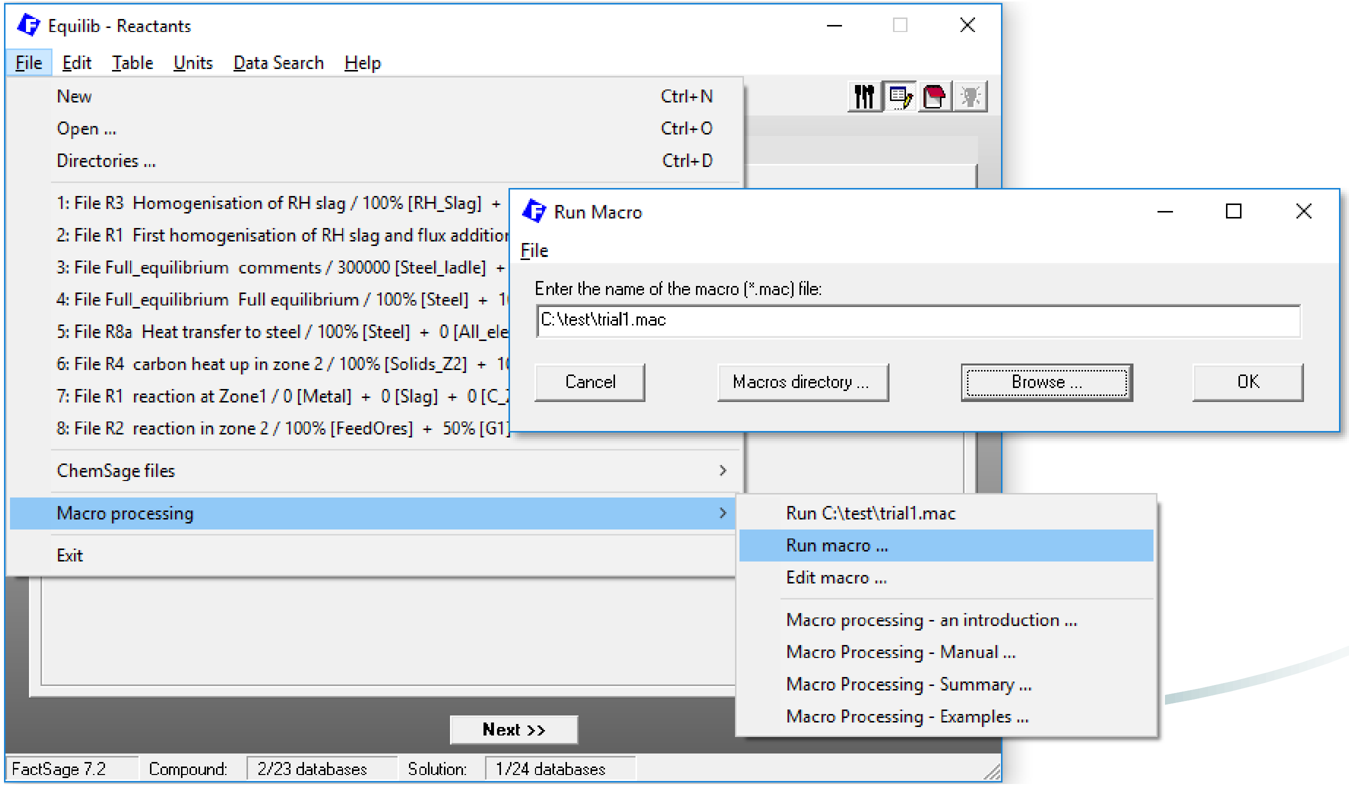


```
// Example 1 *** using simple commands

OPEN C:\Example\Equi1.dat
CALC
SAVE C:\Example\Equi1.dat

END
```

If different from FactSage folder, the path has to be specified at all times

Running the macro file

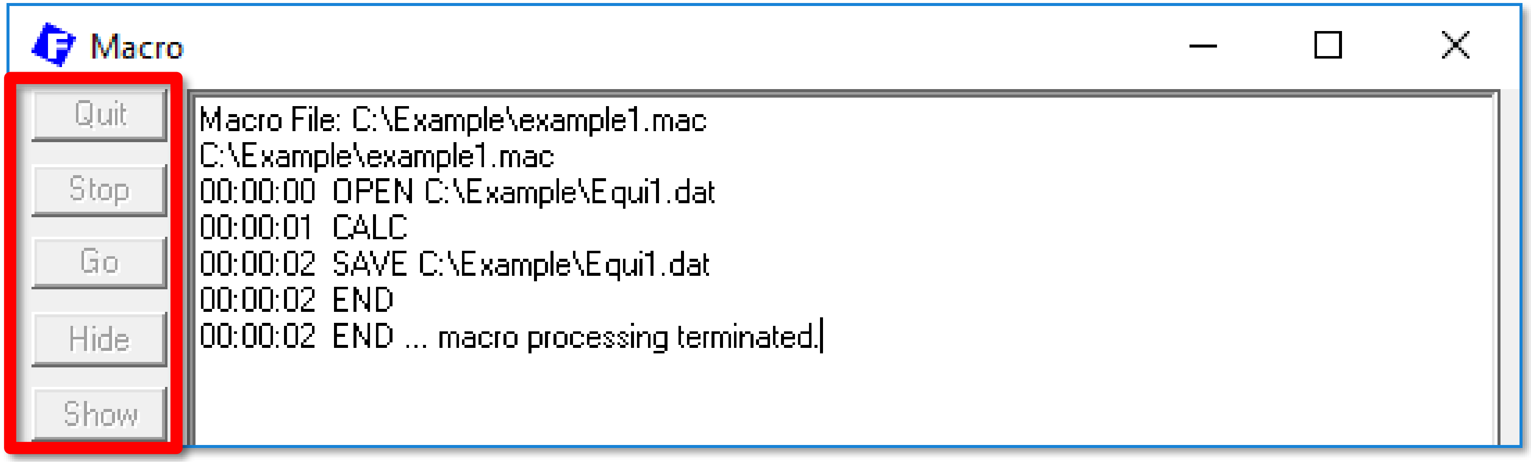
- Go to the Equilib reactants window
- File → Macro processing → Run macro → Browse → Locate your file and open → ok



Enter the name of the macro (*.mac) file: C:\test\trial1.mac

Equilib - Reactants
File
Edit
Table
Units
Data Search
Help
New Ctrl+N
Open ... Ctrl+O
Directories ... Ctrl+D
1: File R3 Homogenisation of RH slag / 100% [RH_Slag] +
2: File R1 First homogenisation of RH slag and flux additio
3: File Full_equilibrium comments / 300000 [Steel_ladle] +
4: File Full_equilibrium Full equilibrium / 100% [Steel] + 1
5: File R8a Heat transfer to steel / 100% [Steel] + 0 [All_ele
6: File R4 carbon heat up in zone 2 / 100% [Solids_Z2] + 1
7: File R1 reaction at Zone1 / 0 [Metal] + 0 [Slag] + 0 [C_
8: File R2 reaction in zone 2 / 100% [FeedOres] + 50% [G1
ChemSage files
Macro processing
Exit
Run C:\test\trial1.mac
Run macro ...
Edit macro ...
Macro processing - an introduction ...
Macro Processing - Manual ...
Macro Processing - Summary ...
Macro Processing - Examples ...
Run Macro
File
Cancel
Macros directory ...
Browse ...
OK
Next >>
FactSage 7.2
Compound:
2/23 databases
Solution:
1/24 databases

Macro status window



```
Macro File: C:\Example\example1.mac
C:\Example\example1.mac
00:00:00  OPEN C:\Example\Equi1.dat
00:00:01  CALC
00:00:02  SAVE C:\Example\Equi1.dat
00:00:02  END
00:00:02  END ... macro processing terminated.
```

- *Quit*: Terminate the execution of the macro
- *Stop*: Pause Macro processing
- When on pause: - *Stop* will terminate processing
  - *Go* will resume processing
- *Hide*: Hide the Equilib windows during processing
- *Show*: Show the Equilib windows during processing

MACRO VARIABLES AND FUNCTIONS

Macro variables

3 types of macro variables:

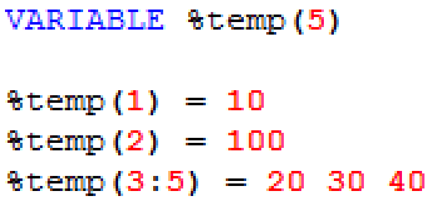
1. **%variable** denotes a user-defined variable – **can** be modified by the user
2. **$variable** denotes a system variable (built-in) – **cannot** be modified by the user
3. **$variable$** denotes a computed or defined thermochemical variable (built-in) – **cannot** be modified by the user

1. User variables - %var

- Identified by ‘**%**’ at the beginning of the name
- The variable type (string, word, number) is not specified
- 9 pre-declared entities (built-in notations): %1 to %9
- %Dir refers to the default FactSage directory (can be reset)
- User variables besides %1 to %9 and %Dir must be declared by the **VARIABLE** statement (only once, unique and non-ambiguous name, best practice is to list all the variables at the beginning of the code)
- Value allocation to %var using ‘=‘ with space before and after

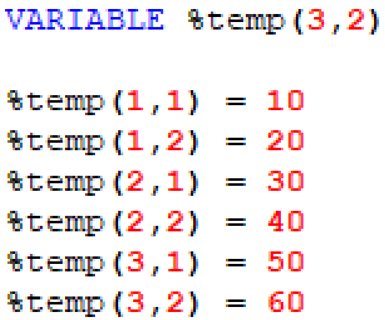
1. User variables - %var

- %var can be a one-dimensional array, declared as %var(*n*), where *n* is the array size



```
VARIABLE %temp(5)

%temp(1) = 10
%temp(2) = 100
%temp(3:5) = 20 30 40
```

- %var can be a 2-dimensional array, declared as %var(*m*,*n*), where *m* is the number of columns and *n* the number of rows (*n* > 1)



```
VARIABLE %temp(3,2)

%temp(1,1) = 10
%temp(1,2) = 20
%temp(2,1) = 30
%temp(2,2) = 40
%temp(3,1) = 50
%temp(3,2) = 60
```

2. System variables - $var

- Identified by '**$**' at the beginning of the name
- Denotes system values (time, date, etc.)
- Cannot be modified by the user

3. Thermochemical variables - var

- Identified by ‘**$**’ at the beginning AND end of the name
- System dependent variables that are defined by the Equilib file or generated during the Equilib calculation:
  - Extensive property values
  - Name, mass, mole, wt.%, activity, etc. for:
    - Elements in solution
    - Product species
    - Solutions
- Cannot be modified by the user

3. Thermochemical variables - var

- Important Thermochemical $variables$:

$variable$	Meaning
E_ET	Temperature
E_DH	Delta enthalpy (ΔH)
E_ei_N	Element **i** name
E_ei_mgk	Total grams of element **i** in solution **k**
E_mNk	Solution **k** name

$variable$	Meaning
E_sNj	Species **j** name
E_sgj	Species **j** grams
E_sWj	Species **j** wt%
E_smj	Species **j** moles
E_saj	Species **j** activity
E_mgk	Total grams of solution **k**

- **i**, **j** and **k** are code numbers from the Equilib result file

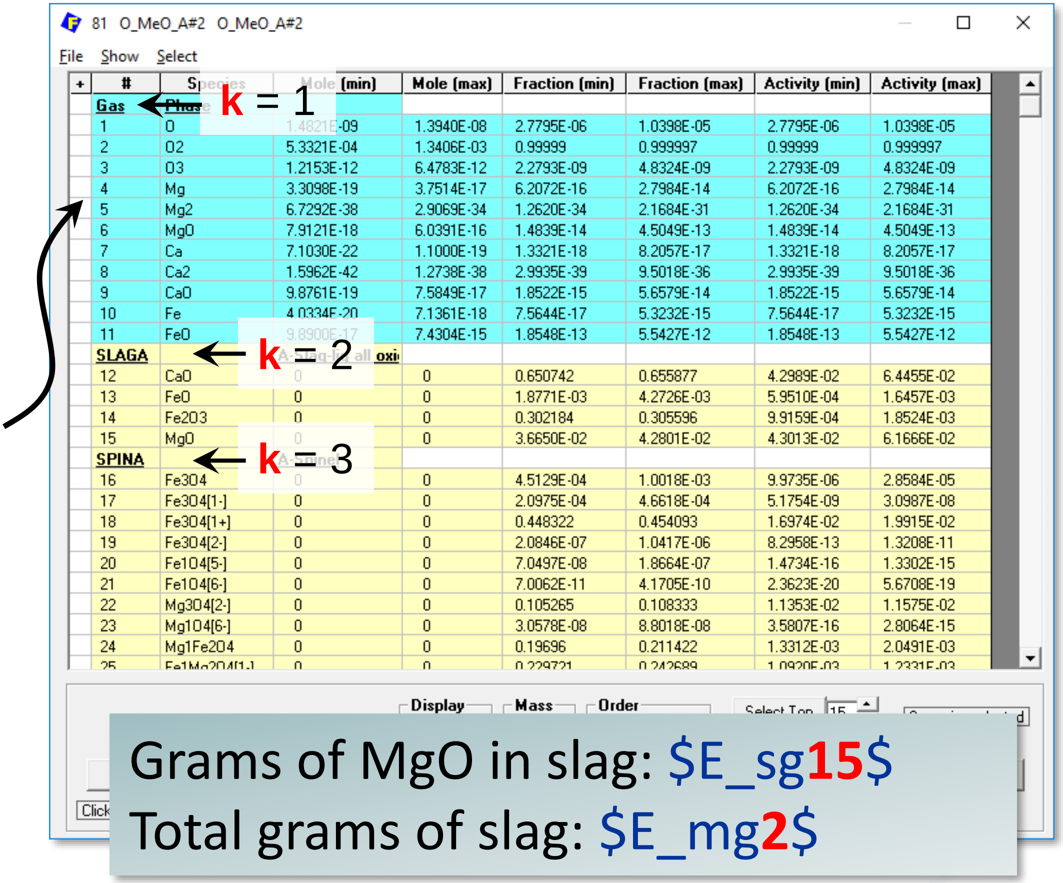
3. Thermochemical variables - var

How to find the code numbers??

Make a dummy calculation at two temperatures

Use Output → Plot → Plot results... → Select species

- Species (**j**): use the same numbers as in the table
- Solutions (**k**): Gas is always 1, then follow the order in the table



+	#	Species	Mole (min)	Mole (max)	Fraction (min)	Fraction (max)	Activity (min)	Activity (max)
	Gas	**Phase**						
	1	O	1.4821E-09	1.3940E-08	2.7795E-06	1.0398E-05	2.7795E-06	1.0398E-05
	2	O2	5.3321E-04	1.3406E-03	0.99999	0.999997	0.99999	0.999997
	3	O3	1.2153E-12	6.4783E-12	2.2793E-09	4.8324E-09	2.2793E-09	4.8324E-09
	4	Mg	3.3098E-19	3.7514E-17	6.2072E-16	2.7984E-14	6.2072E-16	2.7984E-14
	5	Mg2	6.7292E-38	2.9069E-34	1.2620E-34	2.1684E-31	1.2620E-34	2.1684E-31
	6	MgO	7.9121E-18	6.0391E-16	1.4839E-14	4.5049E-13	1.4839E-14	4.5049E-13
	7	Ca	7.1030E-22	1.1000E-19	1.3321E-18	8.2057E-17	1.3321E-18	8.2057E-17
	8	Ca2	1.5962E-42	1.2738E-38	2.9935E-39	9.5018E-36	2.9935E-39	9.5018E-36
	9	CaO	9.8761E-19	7.5849E-17	1.8522E-15	5.6579E-14	1.8522E-15	5.6579E-14
	10	Fe	4.0334E-20	7.1361E-18	7.5644E-17	5.3232E-15	7.5644E-17	5.3232E-15
	11	FeO	9.8900E-17	7.4304E-15	1.8548E-13	5.5427E-12	1.8548E-13	5.5427E-12
	SLAGA		A-Slag-liq all **oxi**					
	12	CaO	0	0	0.650742	0.655877	4.2989E-02	6.4455E-02
	13	FeO	0	0	1.8771E-03	4.2726E-03	5.9510E-04	1.6457E-03
	14	Fe2O3	0	0	0.302184	0.305596	9.9159E-04	1.8524E-03
	15	MgO	0	0	3.6650E-02	4.2801E-02	4.3013E-02	6.1666E-02
	SPINA		A-Spinel					
	16	Fe3O4	0	0	4.5129E-04	1.0018E-03	9.9735E-06	2.8584E-05
	17	Fe3O4[1-]	0	0	2.0975E-04	4.6618E-04	5.1754E-09	3.0987E-08
	18	Fe3O4[1+]	0	0	0.448322	0.454093	1.6974E-02	1.9915E-02
	19	Fe3O4[2-]	0	0	2.0846E-07	1.0417E-06	8.2958E-13	1.3208E-11
	20	Fe1O4[5-]	0	0	7.0497E-08	1.8664E-07	1.4734E-16	1.3302E-15
	21	Fe1O4[6-]	0	0	7.0062E-11	4.1705E-10	2.3623E-20	5.6708E-19
	22	Mg3O4[2-]	0	0	0.105265	0.108333	1.1353E-02	1.1575E-02
	23	Mg1O4[6-]	0	0	3.0578E-08	8.8018E-08	3.5807E-16	2.8064E-15
	24	Mg1Fe2O4	0	0	0.19696	0.211422	1.3312E-03	2.0491E-03
	25	Fe1Mg2O4[1-]	0	0	0.229721	0.242689	1.0920E-03	1.2331E-03

k = 1

k = 2

k = 3

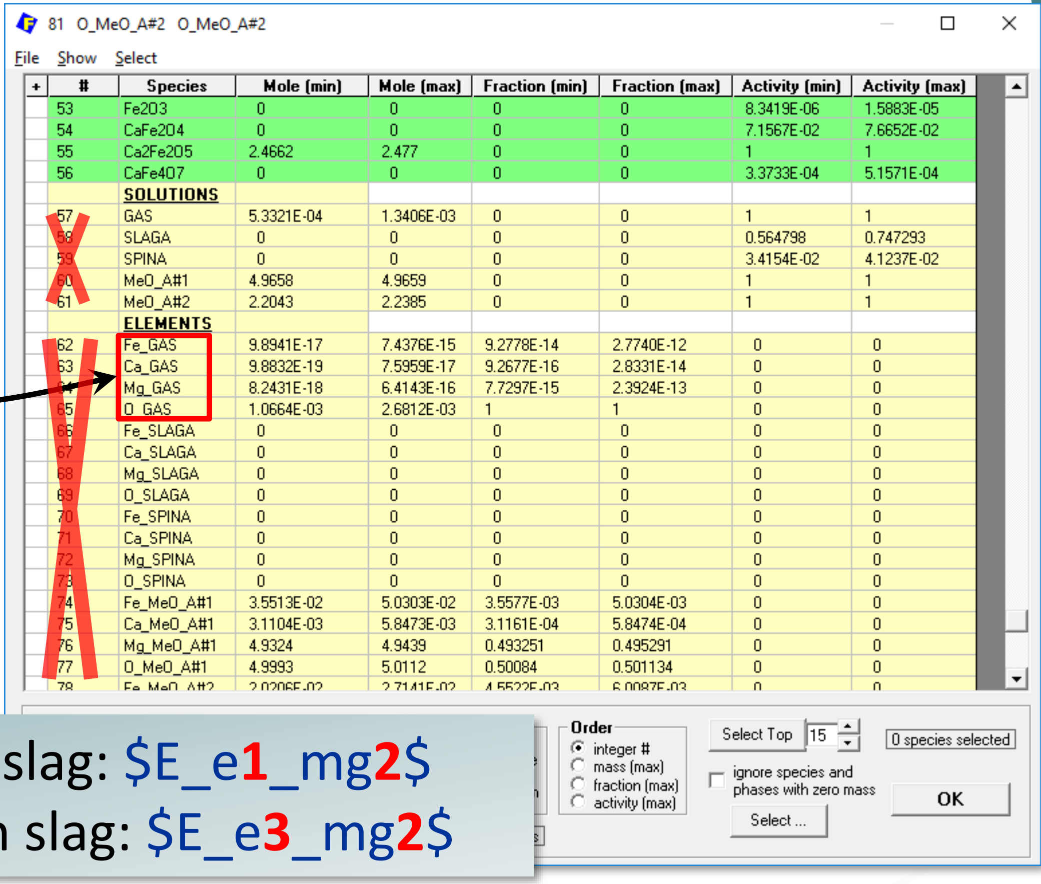
Grams of MgO in slag: $E_sg**15**$

Total grams of slag: $E_mg**2**$

3. Thermochemical variables - var

How to find the code numbers??

Code number for elements (**i**): follow the order given in table



81 O_MeO_A#2 O_MeO_A#2

+	#	Species	Mole (min)	Mole (max)	Fraction (min)	Fraction (max)	Activity (min)	Activity (max)
	53	Fe2O3	0	0	0	0	8.3419E-06	1.5883E-05
	54	CaFe2O4	0	0	0	0	7.1567E-02	7.6652E-02
	55	Ca2Fe2O5	2.4662	2.477	0	0	1	1
	56	CaFe4O7	0	0	0	0	3.3733E-04	5.1571E-04
		SOLUTIONS						
	57	GAS	5.3321E-04	1.3406E-03	0	0	1	1
	58	SLAGA	0	0	0	0	0.564798	0.747293
	59	SPINA	0	0	0	0	3.4154E-02	4.1237E-02
	60	MeO_A#1	4.9658	4.9659	0	0	1	1
	61	MeO_A#2	2.2043	2.2385	0	0	1	1
		ELEMENTS						
	62	Fe_GAS	9.8941E-17	7.4376E-15	9.2778E-14	2.7740E-12	0	0
	63	Ca_GAS	9.8832E-19	7.5959E-17	9.2677E-16	2.8331E-14	0	0
	64	Mg_GAS	8.2431E-18	6.4143E-16	7.7297E-15	2.3924E-13	0	0
	65	O_GAS	1.0664E-03	2.6812E-03	1	1	0	0
	66	Fe_SLAGA	0	0	0	0	0	0
	67	Ca_SLAGA	0	0	0	0	0	0
	68	Mg_SLAGA	0	0	0	0	0	0
	69	O_SLAGA	0	0	0	0	0	0
	70	Fe_SPINA	0	0	0	0	0	0
	71	Ca_SPINA	0	0	0	0	0	0
	72	Mg_SPINA	0	0	0	0	0	0
	73	O_SPINA	0	0	0	0	0	0
	74	Fe_MeO_A#1	3.5513E-02	5.0303E-02	3.5577E-03	5.0304E-03	0	0
	75	Ca_MeO_A#1	3.1104E-03	5.8473E-03	3.1161E-04	5.8474E-04	0	0
	76	Mg_MeO_A#1	4.9324	4.9439	0.493251	0.495291	0	0
	77	O_MeO_A#1	4.9993	5.0112	0.50084	0.501134	0	0
	78	Fe_MeO_A#2	2.0206E-02	2.7141E-02	4.5522E-03	6.0087E-03	0	0

Order
● integer #
○ mass (max)
○ fraction (max)
○ activity (max)

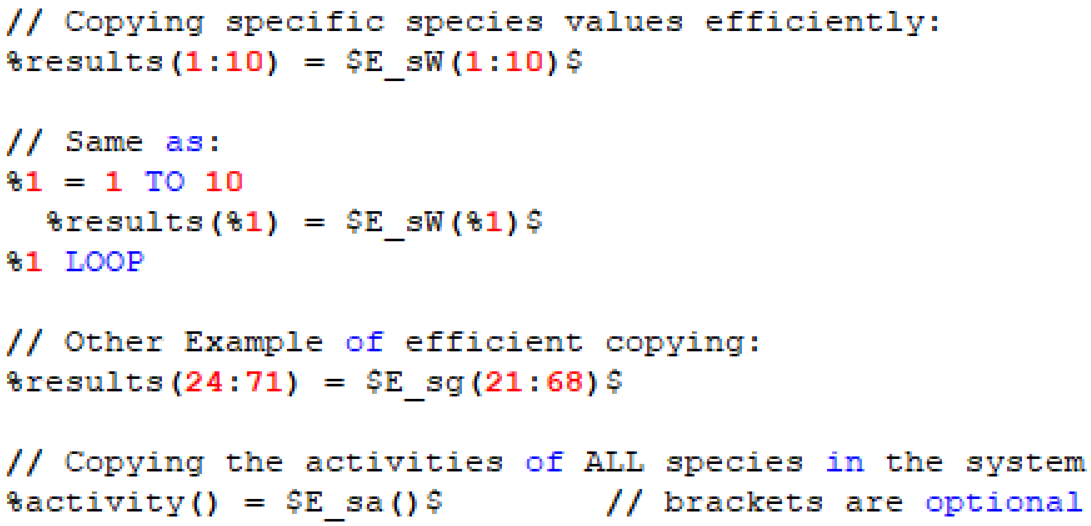
☐ ignore species and phases with zero mass

Grams of Fe in slag: $E_e**1**_mg**2**$

Grams of Mg in slag: $E_e**3**_mg**2**$

3. Thermochemical variables - var

- Thermochemical variables related to species are built like 1D-arrays: concise writing to copy a range of values



```
// Copying specific species values efficiently:
%results(1:10) = $E_sW(1:10)$

// Same as:
%1 = 1 TO 10
  %results(%1) = $E_sW(%1)$
%1 LOOP

// Other Example of efficient copying:
%results(24:71) = $E_sg(21:68)$

// Copying the activities of ALL species in the system:
%activity() = $E_sa()$        // brackets are optional
```

Macro functions - $function()

- Identified by '**$**' at the beginning of the name and argument in-between **()** at the end of the name
- Several system functions (with arguments) for manipulating strings, formatting output and mathematical functions (built-in)
- Most useful function: $MATH() → calculate mathematical expressions including the operators + - / * ^ and other common mathematical functions
- Arguments can be numbers, %variables or thermochemical $variable$
- Other functions of interests: $MIN(), $MAX(), $ABS()

User-defined $functions()

- In addition to the pre-existing functions in FactSage macro processing, the user can define its own function(s) that can be used in the macro code
- The function is a subprogram that receives some arguments from the main body, performs some tasks with those arguments and returns a single value to the main body

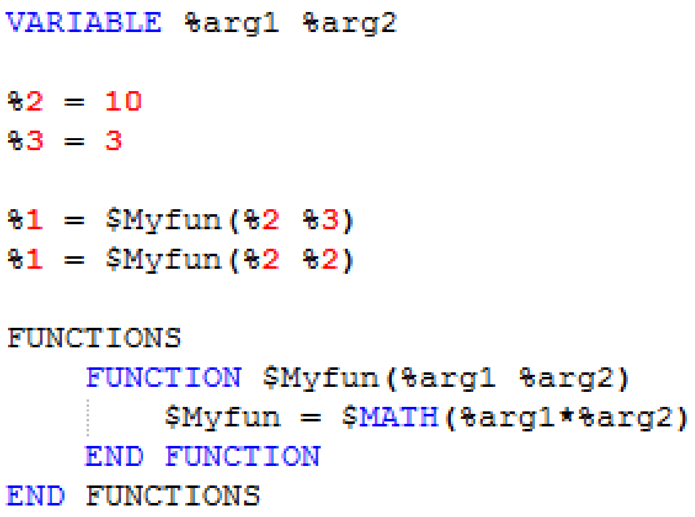
User-defined $functions()

User-defined functions are defined at the **end of the macro**:

```
FUNCTIONS
    FUNCTION $Myfunction1(%arg1 %arg2 …)
            (macro lines)
             $Myfunction1 = ‘value’
    END FUNCTION
    FUNCTION $Myfunction2(%arg1 %arg2 …)
            (macro lines)
             $Myfunction2 = ‘value’
    END FUNCTION
    …
END FUNCTIONS
```

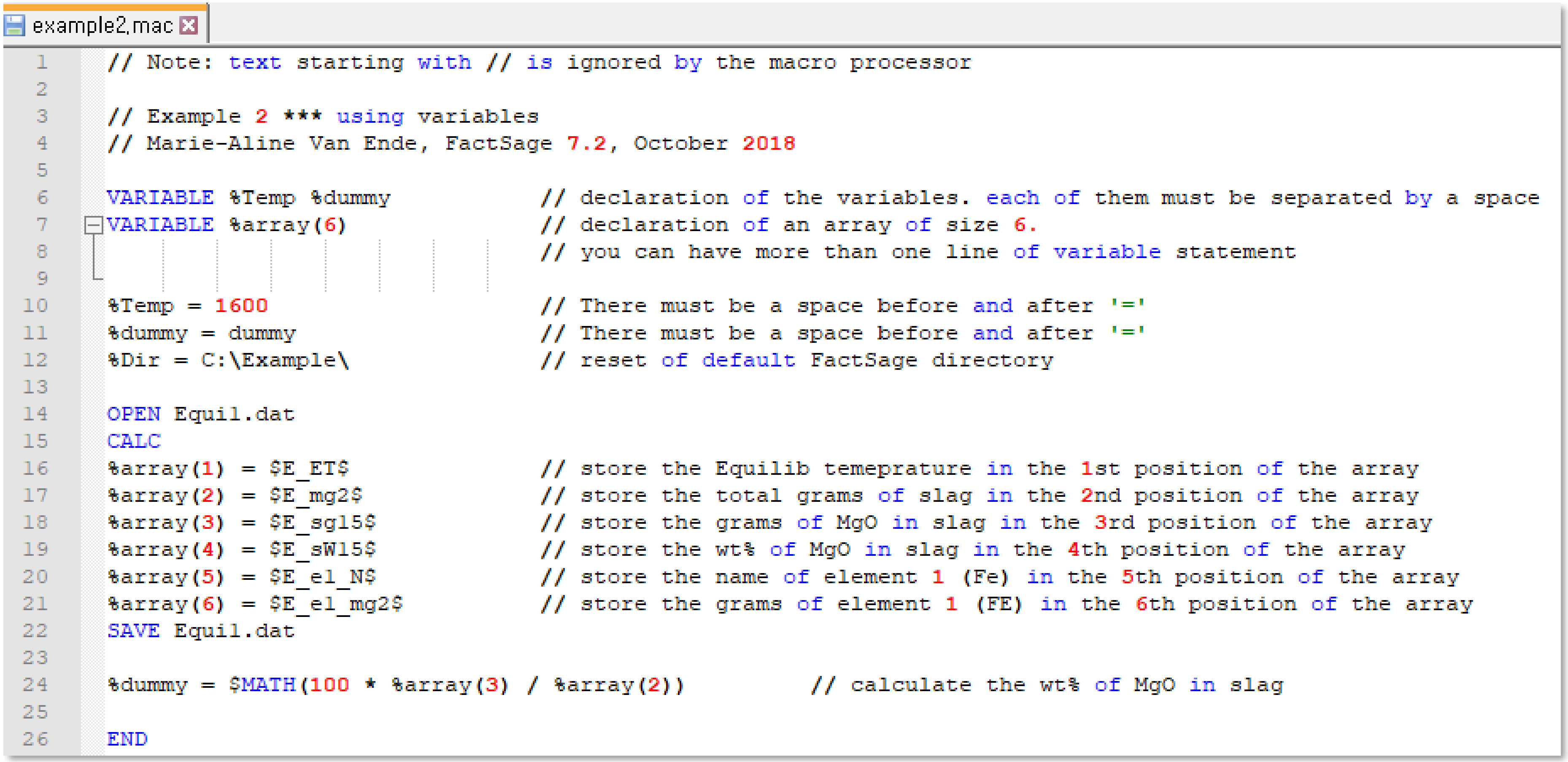
User-defined $functions()

User-defined functions are called in the main body identically to built-in functions:



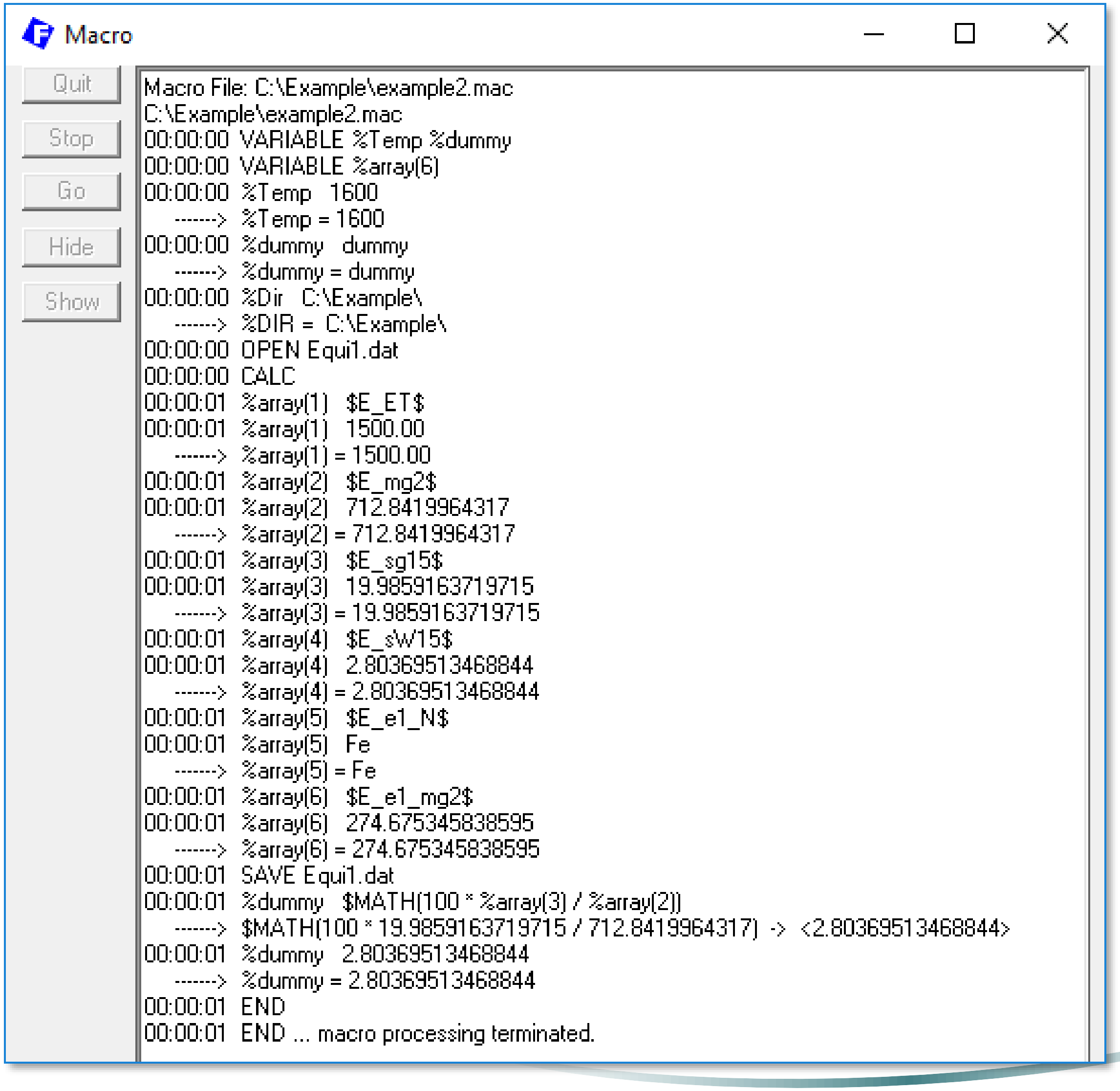
```
VARIABLE %arg1 %arg2

%2 = 10
%3 = 3

%1 = $Myfun(%2 %3)
%1 = $Myfun(%2 %2)

FUNCTIONS
    FUNCTION $Myfun(%arg1 %arg2)
        $Myfun = $MATH(%arg1*%arg2)
    END FUNCTION
END FUNCTIONS
```

Ex. 2: Macro variables and functions



```
// Note: text starting with // is ignored by the macro processor

// Example 2 *** using variables
// Marie-Aline Van Ende, FactSage 7.2, October 2018

VARIABLE %Temp %dummy           // declaration of the variables. each of them must be separated by a space
VARIABLE %array(6)              // declaration of an array of size 6.
                                // you can have more than one line of variable statement

%Temp = 1600                    // There must be a space before and after '='
%dummy = dummy                  // There must be a space before and after '='
%Dir = C:\Example\              // reset of default FactSage directory

OPEN Equil.dat
CALC
%array(1) = $E_ET$              // store the Equilib temeprature in the 1st position of the array
%array(2) = $E_mg2$             // store the total grams of slag in the 2nd position of the array
%array(3) = $E_sg15$            // store the grams of MgO in slag in the 3rd position of the array
%array(4) = $E_sW15$            // store the wt% of MgO in slag in the 4th position of the array
%array(5) = $E_el_N$            // store the name of element 1 (Fe) in the 5th position of the array
%array(6) = $E_el_mg2$          // store the grams of element 1 (FE) in the 6th position of the array
SAVE Equil.dat

%dummy = $MATH(100 * %array(3) / %array(2))         // calculate the wt% of MgO in slag

END
```

Ex. 2: Macro variables and functions



```
Macro File: C:\Example\example2.mac
C:\Example\example2.mac
00:00:00  VARIABLE %Temp %dummy
00:00:00  VARIABLE %array(6)
00:00:00  %Temp   1600
    ------->  %Temp = 1600
00:00:00  %dummy   dummy
    ------->  %dummy = dummy
00:00:00  %Dir   C:\Example\
    ------->  %DIR =  C:\Example\
00:00:00  OPEN Equi1.dat
00:00:00  CALC
00:00:01  %array(1)   $E_ET$
00:00:01  %array(1)   1500.00
    ------->  %array(1) = 1500.00
00:00:01  %array(2)   $E_mg2$
00:00:01  %array(2)   712.8419964317
    ------->  %array(2) = 712.8419964317
00:00:01  %array(3)   $E_sg15$
00:00:01  %array(3)   19.9859163719715
    ------->  %array(3) = 19.9859163719715
00:00:01  %array(4)   $E_sW15$
00:00:01  %array(4)   2.80369513468844
    ------->  %array(4) = 2.80369513468844
00:00:01  %array(5)   $E_e1_N$
00:00:01  %array(5)   Fe
    ------->  %array(5) = Fe
00:00:01  %array(6)   $E_e1_mg2$
00:00:01  %array(6)   274.675345838595
    ------->  %array(6) = 274.675345838595
00:00:01  SAVE Equi1.dat
00:00:01  %dummy   $MATH(100 * %array(3) / %array(2))
    ------->  $MATH(100 * 19.9859163719715 / 712.8419964317)  ->  <2.80369513468844>
00:00:01  %dummy   2.80369513468844
    ------->  %dummy = 2.80369513468844
00:00:01  END
00:00:01  END ... macro processing terminated.
```

MACRO COMMANDS

1. STEP and LOOP
2. Simple and branched program flow
3. Macro commands for Excel Worksheets (OLE)
4. Macro thermochemical commands SET and SAVE

1. STEP and LOOP

- STEP:

  %var STEP value: adds ‘value’ to %var (default 1)

  (Identical to %var = $MATH(%var + value))

- LOOP:

  %var = 'FirstValue' TO 'LastValue' STEP 'IncrementValue'

  (macro lines)

  %var LOOP // End of loop.

  If %var ≠ 'LastValue’, then go back to beginning, increment %var and repeat.

  A LOOP is always executed at least once ('FirstValue').

  If STEP is undefined, '1' is assumed.

  Nested LOOPs are permitted.

2. Simple program flow

- MARK ‘word’
  - A marker used with GOTO ‘word’
  - MARK statement cannot be inside IF… nor in a LOOP
- GOTO ‘word’

  Jump to the line MARK ‘word’

  The jump can go forward or backward in the code

  Can be used to exit a loop before its normal termination (! Not clean termination: the loop variable cannot be reused in another loop!)

2. Simple program flow

- END and GOTO END
  - END indicates the end of processing. All commands after END are ignored.
  - GOTO END will create a jump to the line END, i.e. quit.
- IF … GOTO

  IF %var oper ‘value’ GOTO END

  oper is a logical operator: =, <, >, <>, <=, >=

  If statement true, then go to line END

  Multiple operators AND or OR are permitted

2. Branched program flow

- IF … THEN … ELSEIF … ENDIF

```
IF %var1 oper 'value' THEN
     (macro lines)
ELSEIF %var2 oper 'value' THEN                    <optional>
     (macro lines)
ELSE                                              <optional>
     (macro lines)
ENDIF
```

oper is a logical operator: =, <, >, <>, <=, >=

Multiple operators AND or OR are permitted

Nested IFs are **NOT** permitted.

2. Branched program flow

- DOCASE … CASE … ENDDO

```
DOCASE %var
CASE ‘value1’ [‘value2’ …]          <to specify specific value(s)>
      (macro lines)
CASE ‘value1’ TO ‘value2’           <to specify a range of values>
      (macro lines)
CASE IS ‘operator’ ‘value1’         <to specify a condition>
      (macro lines)
CASE ELSE                           <if none of the above are true>
      (macro lines)
ENDDO
```

3. Linking with Excel worksheet via OLE

- Object Linking and Embedding (OLE) with Excel worksheet
- Up to 9 simultaneous dynamic links can be created (OLE1 to OLE9)
- Enables to read and write in Excel worksheets at any time during macro processing
- Ideal to read inputs and store intermediate Equilib calculation results

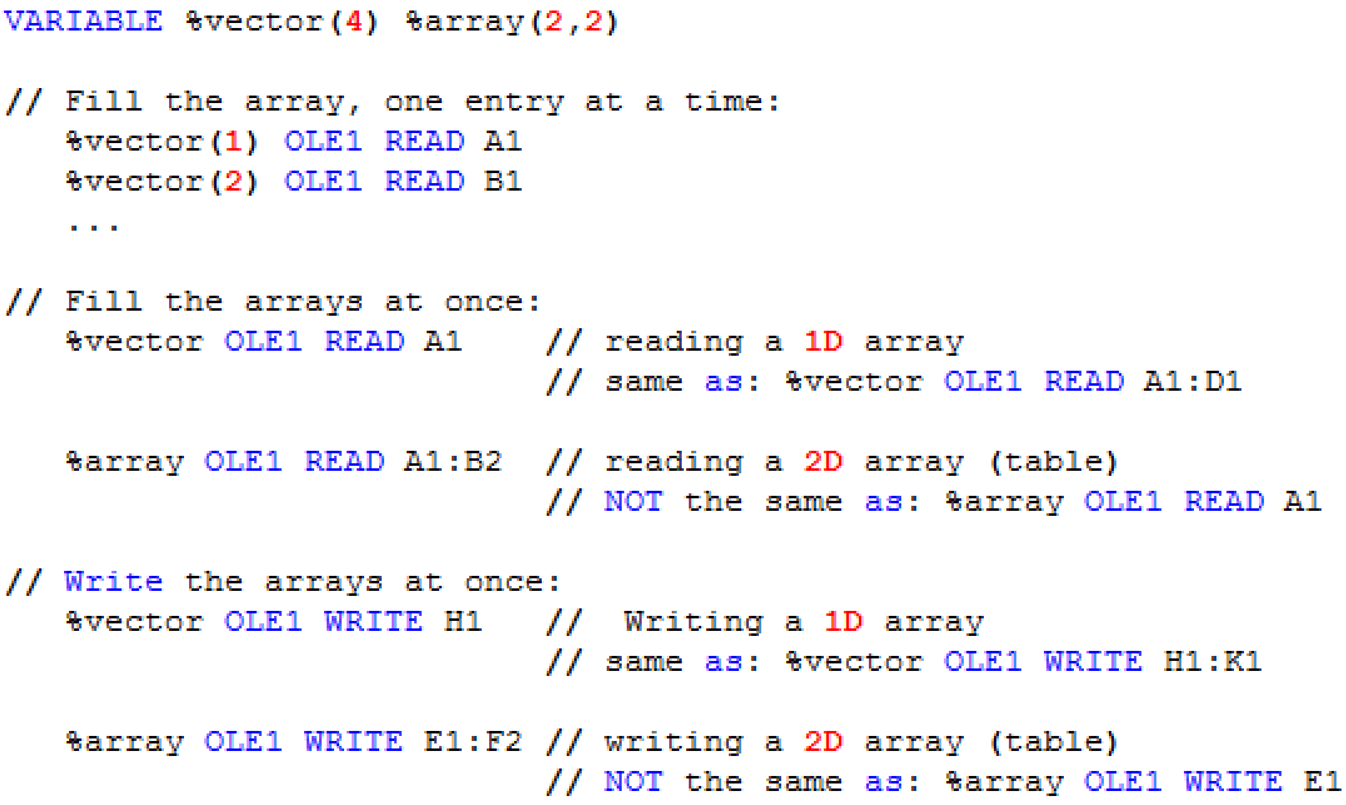
3. Linking with Excel worksheet via OLE

Principal OLE commands:

- Create an OLE link (n = 1 to 9)
  - OLEn Excelfilename.xls(x) sheetname (**.xls** and **.xlsx**)
- Clear all cells or range in the worksheet
  - OLEn CELLS ALL CLEAR - OLEn RANGE A1:C5 CLEAR
- Read a variable value from cell
  - %var OLEn READ B5 or %var OLEn READ R5C2
- Write variable value to cell
  - %var OLEn WRITE B5 or %var OLEn WRITE R5C2

3. Linking with Excel worksheet via OLE

Read from and write to Excel using 1D and 2D variables:



```
VARIABLE %vector(4) %array(2,2)

// Fill the array, one entry at a time:
   %vector(1) OLE1 READ A1
   %vector(2) OLE1 READ B1
   ...

// Fill the arrays at once:
   %vector OLE1 READ A1    // reading a 1D array
                           // same as: %vector OLE1 READ A1:D1

   %array OLE1 READ A1:B2  // reading a 2D array (table)
                           // NOT the same as: %array OLE1 READ A1

// Write the arrays at once:
   %vector OLE1 WRITE H1   //  Writing a 1D array
                           // same as: %vector OLE1 WRITE H1:K1

   %array OLE1 WRITE E1:F2 // writing a 2D array (table)
                           // NOT the same as: %array OLE1 WRITE E1
```

4. Macro thermochemical command - SET

- SET command is used to redefine input reactants, initial amounts, final conditions that are initially defined in the equilib file
- Most important SET commands:
  - SET REACTANT ‘reactant#’ MASS ‘value’ (T,P)
  - SET FINAL T ‘value’ (or other intensive property)
  - SET ESTIMATE T ‘value’

4. Macro thermochemical command - SET

- Additional SET commands (use with caution!!):
  - SET REACTANT DEL ‘chemical formula’
  - SET REACTANT DEL [stream_name]
  - SET REACTANT DEL ‘reactant#’
  - SET REACTANT ADD ‘chemical formula’
  - SET REACTANT ADD [stream_name]
  - SET SELECT GAS + (or + - !, COMPOUND or SOLUTION)
  - SET SELECT SOLUTION I FToxid-SLAGA (or + -)
  - SET SELECT SOLID + FToxid (or GAS, COMPOUND, SOLUTION)
  - SET SELECT SOLID + FToxid MgO

4. Macro thermochemical command - SAVE

Command SAVE:

- Save results file in various format, Excel files, etc.
- Save computed phases in a stream file:
  - SAVE Mixt*Name*.DAT GAS
  - SAVE Mixt*Name*.DAT SOLIDS
  - SAVE Mixt*Name*.DAT SOLUTIONS
  - SAVE Mixt*Name*.DAT FTmisc-FeLQ
  - SAVE Mixt*Name*.DAT FToxid-SLAGA#1

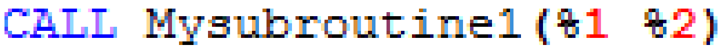
    Where *Name* is the stream name

Other interesting macro commands

Command	Signification
//	Beginning of comment, only in macro file
REM	User comment printed in macro status window
HIDE	Hide Equilib window
SHOW	Show Equilib window
PAUSE	Pause macro processing

Subroutines (PROCEDURES)

- A subroutine is a sequence of code that is designed to perform a specific task
- Subroutines are useful to structure the code, rendering it more efficient and easier to maintain
- In FactSage macro processing, the user can define its own subroutine(s), called procedure
- A procedure is called in the main body by the CALL statement followed by the name of the procedure and arguments, if any, between brackets:



```
CALL Mysubroutine1(%1 %2)
```

Subroutines (PROCEDURES)

- Procedures are defined at the end of the macro code:

```
PROCEDURES
    PROCEDURE Mysubroutine1(%arg1 %arg2 …)
            (macro lines)
    END PROCEDURE
     PROCEDURE Mysubroutine2(%arg1 %arg2 …)
            (macro lines)
    END PROCEDURE
    …
END PROCEDURES
```

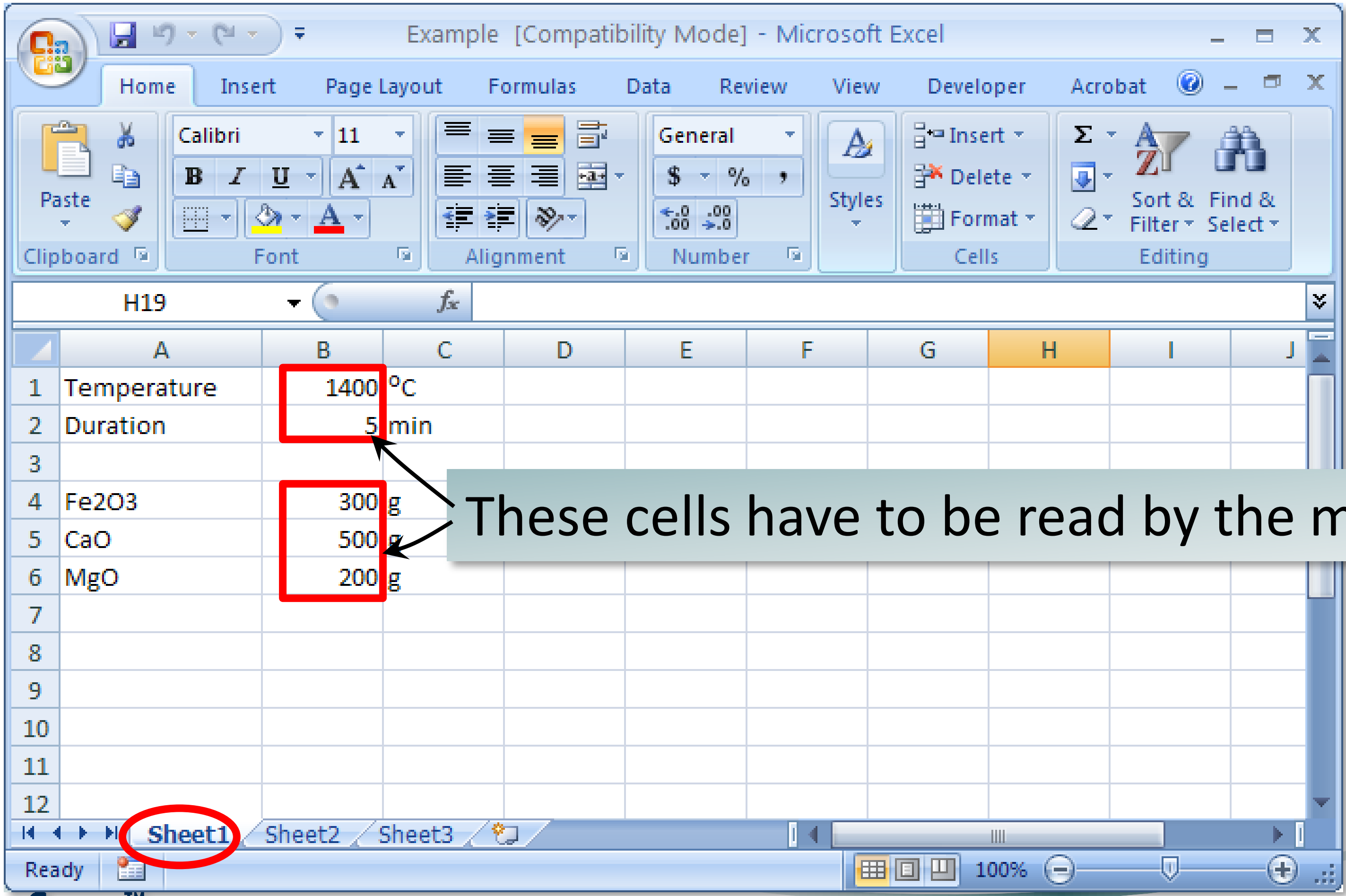
Example 3

- Equi1.dat
- Process of heating the feed by 50 °C/min
- Initial temperature, feed composition and process duration stored in Excel file (Example.xls)

- We want to know and print out the amount of slag and amount of Fe in slag with temperature in Excel sheet

Example 3

Create Example.xls and enter the inputs in ‘Sheet1’

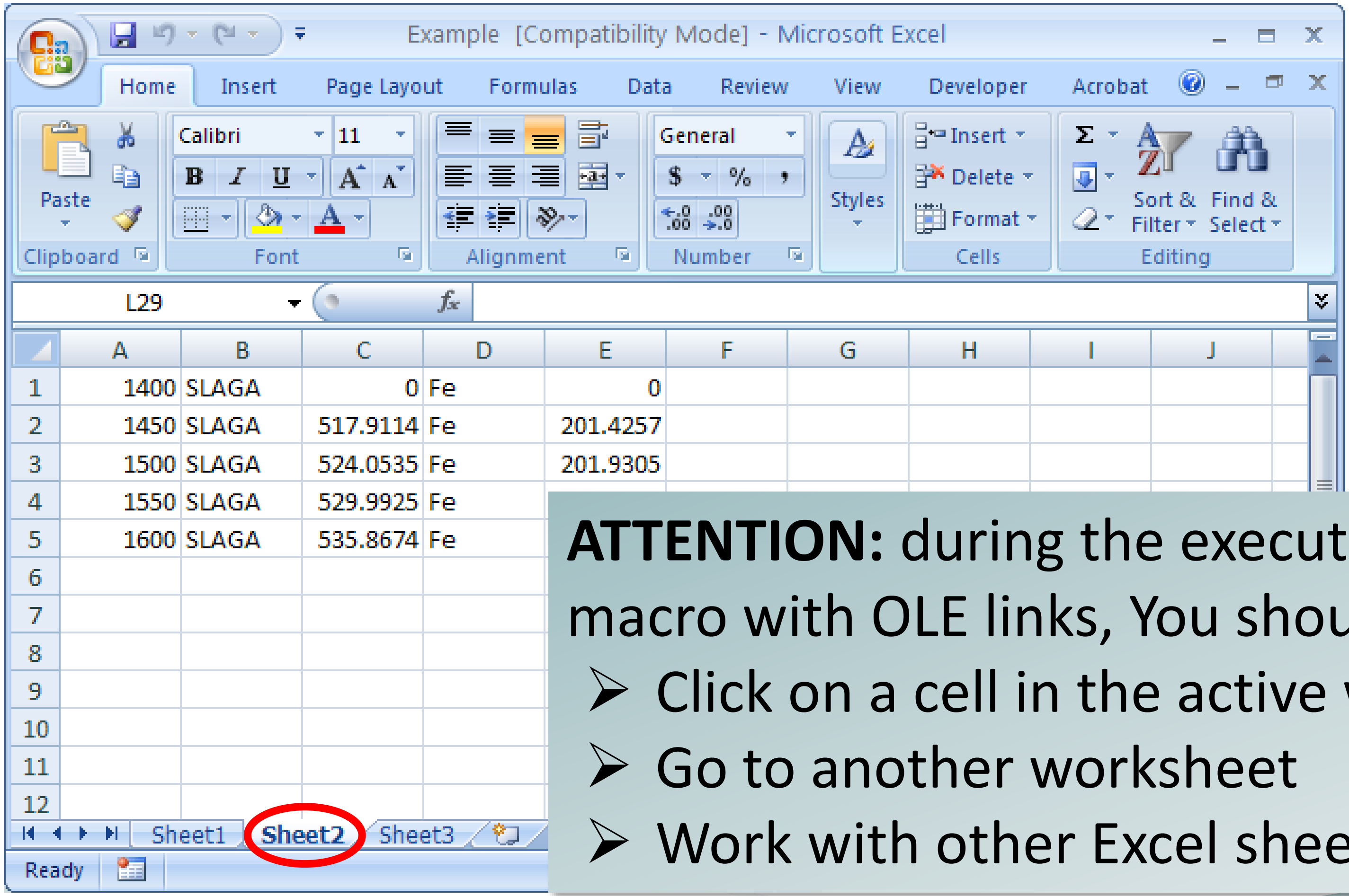


	A	B	C
1	Temperature	1400	°C
2	Duration	5	min
3			
4	Fe2O3	300	g
5	CaO	500	g
6	MgO	200	g

These cells have to be read by the macro

Example 3

Results are written in ‘Sheet2’



	A	B	C	D	E
1	1400	SLAGA	0	Fe	0
2	1450	SLAGA	517.9114	Fe	201.4257
3	1500	SLAGA	524.0535	Fe	201.9305
4	1550	SLAGA	529.9925	Fe	
5	1600	SLAGA	535.8674	Fe	

ATTENTION: during the execution of a macro with OLE links, You should not:

- Click on a cell in the active worksheet
- Go to another worksheet
- Work with other Excel sheets

For more information…

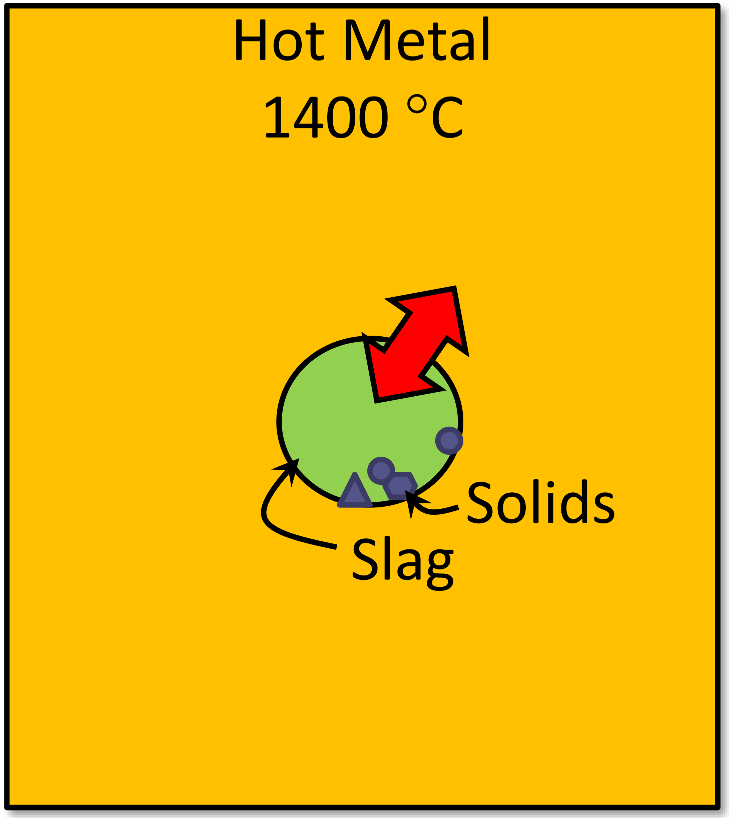
- Help and more information on commands, functions, macro thermochemical variables, etc. can be found in the macro processing manual (FS window → info)
- Don’t hesitate to contact me:

  vanende@snu.ac.kr

  Building 30 room 205

APPLICATION TO DESULFURIZATION OF HOT METAL

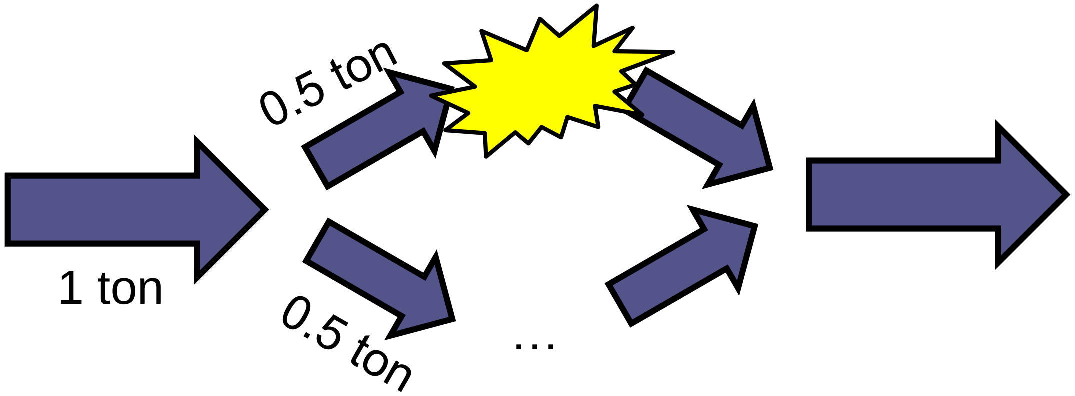
Desulfurization of hot metal



- X g Hot metal, simplified to the Fe-C-Si-S system
- Y g Slag, simplified to the CaO-SiO_2-Al_2O_3 system
- Hot metal is well mixed
- Slag is well mixed and in equilibrium with solids
- Solids stay in slag
- Constant temperature
- Kinetics on Hot Metal/Slag equilibration

Kinetics vs. equilibrium

- Real process is dictated by kinetics
- Kinetics can be simulated with FactSage by allowing only a portion of the feeds to react

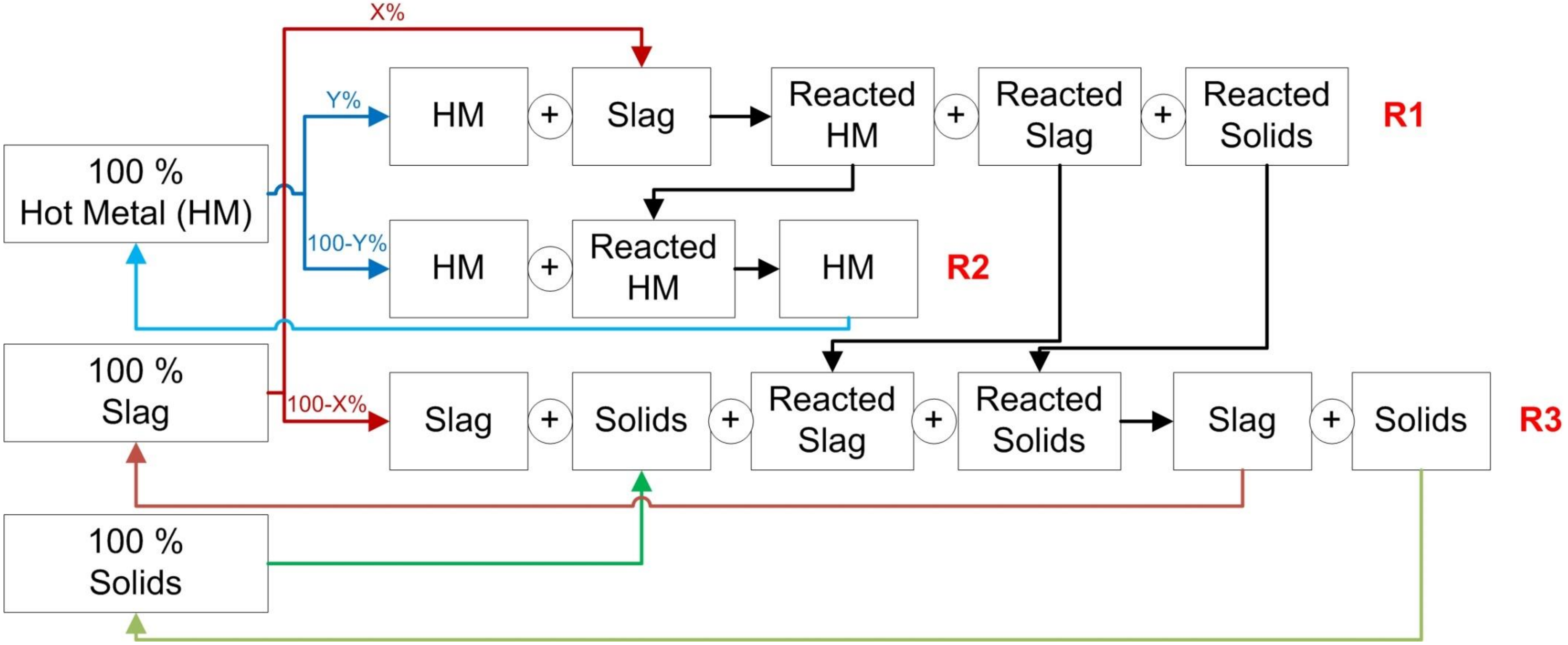


- Finding the suitable proportion of material reacting and non-reacting

Procedure

1. Draw flowsheet
2. Prepare Excel file with input data
3. Prepare Equilib files for initialisation and save the streams. Write the associated macro commands
4. Prepare Equilib files for reactions and save the streams. Write the associated macro commands
5. Organise the Excel file to print the outputs. Write the associated macro commands

Flow sheet of the process


X%
Y%
HM
+
Slag
Reacted HM
+
Reacted Slag
+
Reacted Solids
R1
100 % Hot Metal (HM)
100-Y%
HM
+
Reacted HM
HM
R2
100 % Slag
100-X%
Slag
+
Solids
+
Reacted Slag
+
Reacted Solids
Slag
+
Solids
R3
100 % Solids