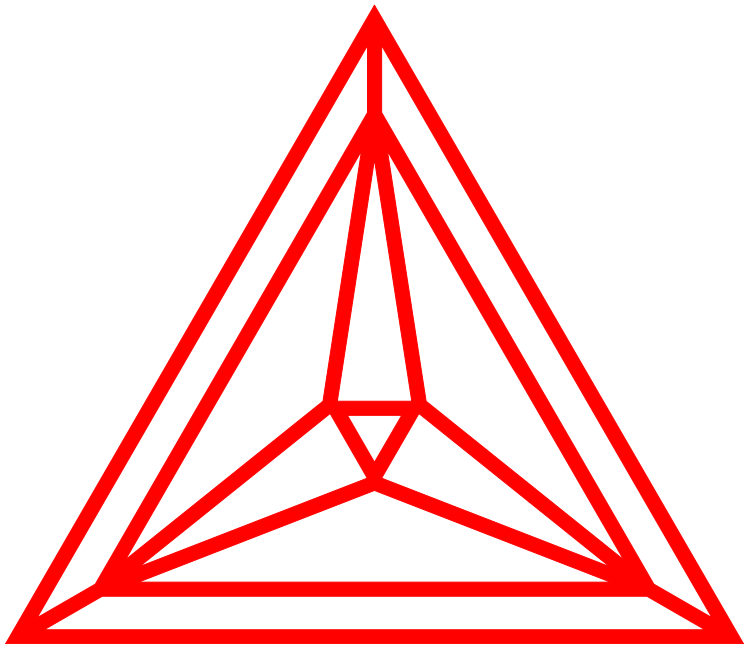
Thermodynamic Calculation Interface

TC MATLAB® Toolbox

(Version 6)

Programmer's Guide



Thermo-Calc Software AB
Norra Stationsgatan 93 5 tr
SE-113 64 Stockholm, Sweden

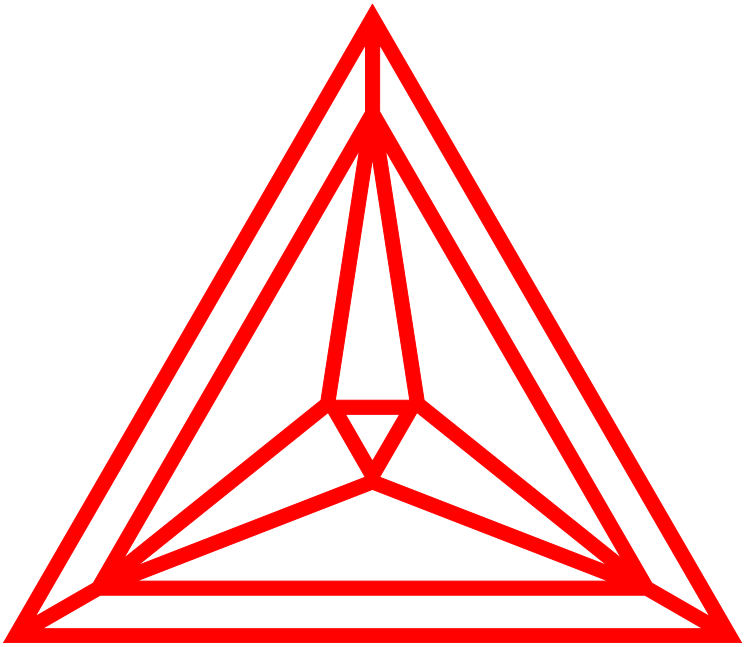
2013

Thermodynamic Calculation Interface

TC MATLAB® Toolbox

(Version 6)

Programmer's Guide



Developer: Thermo-Calc Software AB
Norra Stationsgatan 93 5 tr
SE - 113 64 Stockholm, Sweden

Copyright:

The Thermo-Calc and DICTRA software are the exclusive copyright properties of the STT Foundation (Foundation of Computational Thermodynamics, Stockholm, Sweden). All rights are reserved worldwide.

Thermo-Calc Software AB has the exclusive rights for further developing and marketing all kinds of versions of Thermo-Calc and DICTRA software/database/interface packages, worldwide.

This ***TC MATLAB® Toolbox Programmer's Guide***, as well as all other related documentation, is the copyright property of Thermo-Calc Software AB.

It is absolutely forbidden to make any illegal copies of the software, databases, interfaces, and their manuals (User's Guide and Examples Book) and other technical publications (Reference Book and Technical Information). Any unauthorized duplication of such copyrighted products, is a violation of international copyright law. Individuals or organizations (companies, research companies, governmental institutes, and universities) that make or permit to make unauthorized copies may be subject to prosecution.

The utilization of the Thermo-Calc and DICTRA software/database/interface packages and their manuals and other technical information are extensively and permanently governed by the ***Thermo-Calc Software AB END USER LICENSE AGREEMENT (EULA)***, which is connected with the software.

Disclaimers:

Thermo-Calc Software AB and the STT Foundation reserve the rights to further developments of the Thermo-Calc and DICTRA software and related software/database/interface products, and to revisions of their manuals and other publications, with no obligation to notify any individual or organization of such developments and revisions. In no event shall Thermo-Calc Software AB and the STT Foundation be liable to any loss of profit or any other commercial damage, including but not limited to special, consequential or other damage.

Please visit the Thermo-Calc Software web site (www.thermocalc.com) for any modification and/or improvement that have been incorporated into the programs, or for any amendment that have made to the contents of the various User's Guides and to the FAQ lists and other technical information publications.

Acknowledgement of Copyright and Trademark Names:

Various third-party software names that are protected by copyright and/or trademarks are mentioned for descriptive purposes, within this User's Guide and other documents of the Thermo-Calc and DICTRA software/database/interface packages. Due acknowledgement is herein made of all such protections.

Contents

1. Introduction

Thermo-Calc is a general software package for manipulation of thermodynamic quantities and multi-component phase equilibrium calculations. Currently, there are three application programming interfaces available for Thermo-Calc: TQ, TC-API and TC MATLAB Toolbox (see Fig. 1). In this guide TC MATLAB Toolbox, the interface between Thermo-Calc and MATLAB, is treated. The idea behind the different application programming interfaces for Thermo-Calc is that the application programmer should not have to bother about the complexity of the Thermo-Calc kernel but still be able to use its powerful features in their own programs.

MATLAB is flexible software for technical computing and visualization of data. The software comes with more than 600 mathematical, statistical and engineering functions and great graphical capabilities. It can be considered a matrix-oriented programming language and contains compilers, links and libraries for different scientific applications. This flexibility is now enhanced even more with the possibility to retrieve thermodynamic and kinetic quantities through the TC MATLAB Toolbox. This programming interface is ideal for fast realization of ideas during research and development activities.

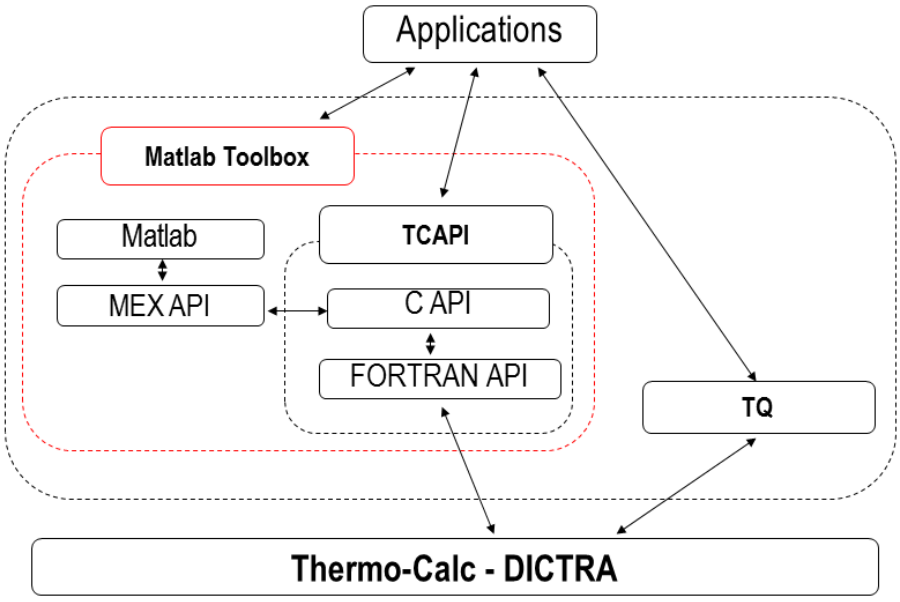


Fig. 1. Illustration of the different API's available for Thermo-Calc.

To be able to call MATLAB from programs written in C or FORTRAN there are so-called MEX-files (MATLAB Executable) included with the MATLAB software. These MEX-files were utilised when interfacing MATLAB with Thermo-Calc, see Fig. 1.

For every Thermo-Calc function implemented in the MEX-files there is a corresponding m-file, thus it is possible to call Thermo-Calc from MATLAB just by running the corresponding m-file.

In the current version of the TC MATLAB Toolbox more than 50 commands are available for the application programmer. For more information, general functionality and applications of the MATLAB software please refer to the documentation provided by the MathWorks Ltd. (www.mathworks.com).

2. How to install and run TC MATLAB Toolbox

As with TQ-I and TC-API it is also necessary to have the Thermo-Calc 3.0 software/database package installed on the same computer or on a server in order to run the TC MATLAB Toolbox. If interaction with DICTRA is desired a licensed version of that software is also required. At present TC MATLAB Toolbox is only available for the PC-Windows environment.

The installation of TC MATLAB Toolbox is straightforward. Run the installation program and follow the instructions. The path to the location where TC MATLAB Toolbox was installed will be given to both Windows and MATLAB by the installation script.

To test the installation, start MATLAB and type: “tc_init_root” in the command window and press return. This should result in no return message if the installation was successful. All of the commands available in the toolbox will be described in section 3 of this document. A short description of each command can be obtained by typing “help ’thermo-calc toolbox 6’” in the command window. Examples can be found in thermo-calc toolbox folder.

3. Description of commands in TC MATLAB Toolbox

The commands in the TC MATLAB Toolbox can be divided into six groups depending on their purposes. The groups are:

TC_ROOT	refers to general information and miscellaneous commands
TC_DATABASE	refers to information and commands in the database module
TC_SYSTEM	refers to information and commands in the database module
TC_UTIL	refers to various commands e.g. “tc_define_system”
TC_GES5	refers to information and commands in the GES5 module
DIC_DICTRA	refers to information and commands in DICTRA

In order to avoid conflict with reserved names all commands in the TC MATLAB Toolbox starts with “tc_” an exception are the commands referring to commands in DICTRA which start with “dic_”.

3.1 TC_ROOT

Name	Arguments	Description
tc_init_root	None	initialize the Thermo-Calc subsystem. Must be called before any other command in the Toolbox.
tc_deinit	None	closes the Thermo-Calc session and returns the license key
tc_version	string: version_name	returns the current version of the Thermo-Calc subsystem.
tc_poly3_command	string: command	sends a command “command” to the POLY-3 module.
tc_read_poly3_file	string: file_name	reads stored POLY-3 file “file_name”.
tc_save_poly3_file	string: file_name	saves a POLY-3 file in “file_name”.

3.2 TC_DATABASE

Name	Arguments	Description
tc_append_database	string: database_name	appends database “database_name”.
tc_element_select	string: element_name	selects an element “element_name” from the current database.
tc_get_data	None	executes the GET_DATA command.
tc_open_database	string: database_name	opens database “database_name”.
tc_phase	integer: no_phases string array phase_names	returns the number of phases in “no_phases” and phase names in “phase_names”.
tc_phase_reject	string: phase_name	rejects phase “phase_name” in the current database.
tc_phase_select	string: phase_name	selects phase “phase_name” in the current database.

3.3 TC_SYSTEM

Name	Arguments	Description
tc_error	integer: error_code string: error_message	checks if an error occurred. If so an error code “error_code” and an error message “error_message” are returned.
tc_reset_error	None	resets the error handling in the Thermo-Calc subsystem
tc_compute_equilibrium	None	executes the COMPUTE_EQULIBRIUM command in POLY-3
tc_component_status	string: status string: comp_name	returns the status “status” for component “comp_name”, “status” could be either ‘ENTERED’ or ‘SUSPENDED’.
tc_create_new_equilibrium	integer: eq_number	command to create a new equilibrium with equilibrium number “eq_number”.
tc_define_components	string: new_components	changes the set of components to those in “new_components”
tc_degrees_of_freedom	integer: number	returns the degrees of freedom “number” in the system,
tc_delete_condition	string: condition_name	deletes the condition “condition_name”.
tc_delete_symbol	string: symbol_name	deletes the symbol “symbol_name”.
tc_enter_constant	string: constant_name double: value	enters a symbol of type ‘CONSTANT’ with name “constant_name” and value “value”.
tc_enter_function	string: function_name string: function_expression	enters a symbol of type ‘FUNCTION’ with name “function_name” and expression “expression”.
tc_enter_symbol	string: symbol_name string: symbol_type integer: argument_type integer: int_value double: double_value string: char_value	enters a symbol ”symbol_name” of type ”symbol_type” (=’CONSTANT’, ‘FUNCTION’, ‘TABLE’ or ‘VARIABLE’) with an argument of type ”argument_type” (=1 for integer, 2 for double or 3 for string).
tc_enter_table	string: table_name string: table_expression	enters a symbol of type ‘TABLE’ with name “table_name” and expression “expression”.
tc_enter_variable	string: variable_name double: value	enters a symbol of type ‘VARIABLE’ with name “variable_name” and value “value”.
tc_get_derivatives	string: phase string array: arr1 string array: arr2	returns the Gibbs energy and the first and second derivatives with respect to site-fractions for phase “phase”. The array “arr1” will contain the Gibbs energy and the first derivatives and the array “arr2” will contain the second derivatives.
tc_get_value	string: expression double: value	retrieves the current value “value” of any state variable, function or variable set in “expression”.
tc_list_component	integer: no_components string array: components	returns the number of components in “no_components” and a list of all components in “components”.
tc_list_conditions	integer: no_conditions string array: conditions	returns the number of conditions in “no_conditions” and a list of all conditions in “conditions”.
tc_list_phase	integer: no_phases string array: phases	returns the number of phases in “no_phases” and a list of all phases in “phases”.
tc_list_species	integer: no_species string array: species	returns the number of species in “no_species” and a list of all species in “species”.
tc_list_symbols	integer: no_symbols string array: symbols	returns the number of symbols in “no_symbols” and a list of all defined symbols in “symbols”.
tc_phase_status	string: status string: phase_name	returns the status “status” for the phase in “phase_name”.
tc_select_equilibrium	integer: eq_number	command to switch to another set of conditions and equilibria. The desired set of conditions and equilibria are indicated by its equilibrium number “eq_number”.
tc_set_component_status	string: comp_name string: status	sets the status “status” (‘ENTERED’ or ‘SUSPENDED’) for component “comp_name”.

tc_set_condition	string: expression double: value	sets a condition for “expression” to “value”.
tc_set_minimization	string: flag	turns global minimization on or off by setting the string flag to ‘on’ or ‘off’
tc_set_phase_addition	string: phase_name double: value	command to add a value “value” to the Gibbs energy expression of phase “phase_name”.
tc_set_phase_status	string: phase_name string: status double: value	sets status “status” (‘ENTERED’, ‘DORMANT’, ‘FIXED’ or SUSPENDED) to phase “phase_name”. A value “value” is to set for status ‘ENTERED’ and ‘FIXED’.
tc_set_start_value	string: name double: value	sets a start value “value” for a state variable “name”
tc_species_status	string: status string: species_name	returns the status “status” for a specie “species_name”

3.4 TC_UTIL

Name	Arguments	Description
tc_check_error	string:	checks for errors and resets them. This command could be considered a combination of “tc_error” and “tc_reset_error”.
tc_define_system	string: database_name string: element_names string: reject_phases string: restore_phases	One single command to define a system with database “database_name”, elements “element_names”, phases to reject in “reject_phases” and phases to restore in “restore_phases”.
tc_prompt	string: tprompt integer: defval	prompts the user to input a integer value
tc_promptr	string: tprompt double:defval	prompts the user to input a double value
tc_prompts	string: tprompt string: defval	prompts the user to input a string
tc_promptsn	string: tprompt string array: defval	prompts the user to input a string array

3.5 TC_GES5

Name	Arguments	Description
tc_enter_ges5_parameter	string: parameter_name string: parameter_expression	enters a parameter “parameter_name” in “parameter_expression”.
tc_ges5_command	string: command	sends a command “command” to the GES5 monitor.
tc_get_ges5_parameter	string: parameter_expression string: parameter_name	returns a parameter expression “parameter_expression” for parameter “prameter_name”.

3.6 DIC_DICTRA

Name	Arguments	Description
dic_command	string: command	sends a command to the DICTRA monitor
dic_convert_sitefractions	double array: new_fractions string: phase_name double array: sitefractions integer: fraction_type	convert site fractions in “sitefractions” for phase “phase_name”. New fractions is set in “new_fractions”. “fraction_type”=1, 2, 3 will return mole-, mass- or u-fractions respectively.
dic_get_independent_component	integer: no_idepc string array: comp_names string region_name	returns the number of independent components in “no_idepc” and a list of component names in “comp_names” for region “region_name”.
dic_list_profile	integer: no_gridpoints integer: no_sitefractions double: sitefractions double array: gridpoints string: region_name string: phase_name	returns a stored profile for phase “phase_name” and region “region_name”.
dic_list_timesteps	integer: no_timesteps double array: timesteps	returns the number of time steps in “no_timesteps” and a list of time steps in “timesteps”.
dic_read_workspace	string: file_name	reads the stored DICTRA simulation file in “file_name”.
dic_region_info	integer: no_gridpoints double: region_size double: start_coordinate string: region_name	returns information about region “region_name”: the size of the region in “region_size”, number of grid points in “no_gridpoints” and value of the first coordinate in “start_coordinate”.
dic_save_workspace	string: file_name	saves a DICTRA simulation file to “file_name”.
dic_select_timestep	integer: time_step	selects a time step from a stored DICTRA simulation file.
dic_simulate_reaction	None	command to start the simulation.