RETRIEVAL OF THERMODYNAMIC DATA FOR MODELLING OF SOLIDIFICATION USING TQ INTERFACE

Marek WRÓBEL*, Andriy BURBELKO

*AGH University of Science and Technology, Cracow, Poland, EU, * marek.wrobel@agh.edu.pl*

Abstract

CALPHAD-based calculations, for quite some time now, have been very useful and convenient way to obtain data needed for modelling of solidification process. Among scientists, interested in and working on the topic, knowledge of existence of software for thermodynamic calculations is widely spread, but at the same time its usage is limited. A significant amount of time has to be invested if one is to be able to utilize such program. This may, unfortunately, prevent new users from arising. An example of the application of data retrieved employing Thermo-Calc will be presented. Restrictions regarding calculated system and data itself will be specified. With employment of the programming interface, namely TQ, comes new look upon the problem. One no longer needs to derive formula for desired property as a function of specified condition(s) to input it into modelling algorithm. Instead one can make thermodynamic computations and "on the flight" send the results to the working modelling program. Still, knowledge of Thermo-Calc is essential to use the programming interface. Here with help comes specially designed code that enables those not working with TC to exploit benefits of thermodynamic calculations combined with TQ interface. Examples of using the program will be given together with display of simplicity of theirs employment. Justification of applying such approach in modelling of, among the others, solidification process will be given.

Keywords: CALPHAD, TQ interface, modelling

1. INTRODUCTION

When describing the CALPHAD method it feels natural to mention the name Larry Kaufman, who was the first to present its general description in [1]. In 1977, under his editorship, first issue of CALPHAD journal appeared. For the brief history of the method - as very title of the paper says - one is referred to [2], whereas readers interested in detailed description of the topic are sent to [3].

According to authors' observations, the field of application of the CALPHAD-based programs is not yet entirely filled - there are still many institutions and science groups that would benefit from assigning this method of data acquisition. At the same time there is tendency that units that would gain access to such software are selecting an individual to learn and use it. This paper is directed to both new users and such groups that would like to benefit from the skills of a single operator.

The idea of laying responsibility for making the thermodynamic calculations on one user is generally understandable and has proven to work very well. An example of application of data retrieved form Thermo-Calc (TC) by one scientist and incorporating it into modelling program of the other will be given. It will be explained that when trying to simulate more complex systems the need for assigning more sophisticated tool, namely TQ: programming interface for TC, appears. At this point, if work related with thermodynamic calculations is still being distinct from the other duties, the user of TC no longer provides formulae that can be input into modelling program's algorithm but the very code that uses TQ subroutines. In other words modelling program gains another author and simultaneously issues of combining codes of modelling algorithm and data retrieval becomes challenging. That is the reason why maintaining this division becomes problematic. This is especially true when programming in Fortran language - one still used commonly by older generation of scientists but being limited to procedural programming. Overcoming this programming difficulty is the main topic of this paper. An approach of designing stand-alone code that can conveniently be linked with the model and does not require knowledge of TC by other programmers will be presented. It will be shown that this code can be prepared in such a way, that its flexibility would allow to make calculations for systems of different order and with varying elements. Examples of data retrieval using this concept will be given.

2. DATA RETRIEVAL

2.1 Calculations without use of the TQ interface

When modelling solidification of binary Fe-C, one of the methods of calculation of the driving force of solidification is to treat it as value related to degree of undercooling [4]. For that one needs to know liquidus temperature for considered range of carbon concentration. Performing easy TC calculations this can be obtained in just a few moments but if need be, one can even digitalize Fe-C phase diagram, which is so common in the literature. Obtained function can be encoded in the model.

Next example is an abstract one, but it shows very well that it is the way of putting the problem that defines complexity of solution. Suppose that in multicomponent, multiphase system, namely: Fe-C-Si-Mn-S-P, one needs to calculate concentration of Mn, for the carbon content varying from 3 to 5 wt. %, at which graphite and MnS phases, under equilibrium conditions, would begin to solidify simultaneously. This time CALPHAD calculations are essential, but still the solution (**Fig. 1**), as a function of one variable, is easy to be formulated.

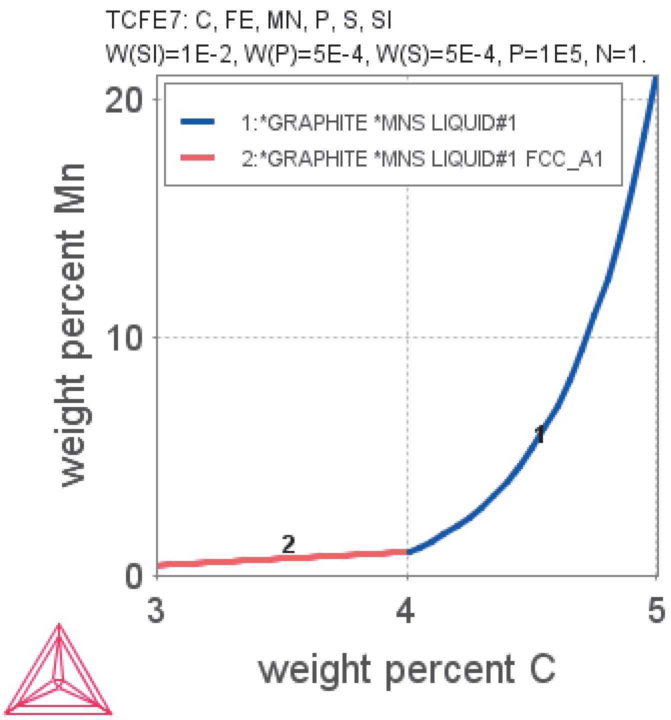

Fig. 1 Plot of relation between content of Mn and C in Fe-C-Si-Mn-S-P system, at which MnS and Graphite would begin to solidify simultaneously, under equilibrium conditions

Another example, calculations of graphite's, austenite's and liquid's density in binary Fe-C system, shows where the limit of such approach lies. Graphite's density, being function only of temperature is straightforward to obtain, but for austenite and liquid phases another variable appears, namely carbon concentration. **Figs. 2** and **3** show surfaces that's values needs to be formulated. For liquid phase equation of the plane can be used to express function of density. Eq. (1) has been derived:

$$\rho_{liq} = -9{,}5 \cdot 10^{-2} \cdot w(C) - 5{,}5 \cdot 10^{-4} \cdot T + 8{,}042 \qquad (1)$$

where T and $w(C)$ stand for temperature given in °C and carbon concentration in wt. %, respectively. Approximating austenite's density using plane would lead to large errors. The properties of **Fig. 3**b, namely parallelism of diagram lines for varying temperature and linear dependence of density on temperature for constant wt. % C, allow to derive surface equation by first formulating polynomial dependence of density vs. weight % C and then appending temperature relation. This resulted with obtaining Eq. (2):

$$\rho_{fcc} = -5\cdot 10^{-2}\cdot [w(C)]^2 - 3{,}17\cdot 10^{-2}\cdot w(C) - 4{,}7\cdot 10^{-4}\cdot T + 8{,}072 \qquad (2)$$

Detailed description of obtaining this relations as well as assessment of deviation from the exact values can be found in [5-11]. Additionally the paper presents results acquired with the modelling program, which incorporated (1) and (2), for simulation of porosity formation.

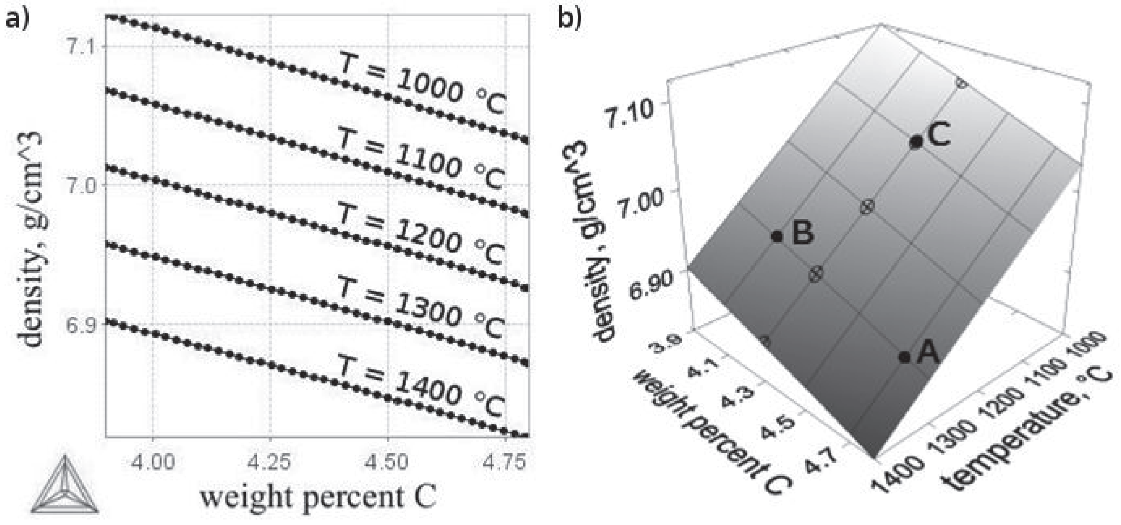

Fig. 2 Dependency of liquid phase's density on temperature and carbon concentration on 2D (a) and 3D (b) plot. Coordinates of points A, B, C were used to determine equation of the plane

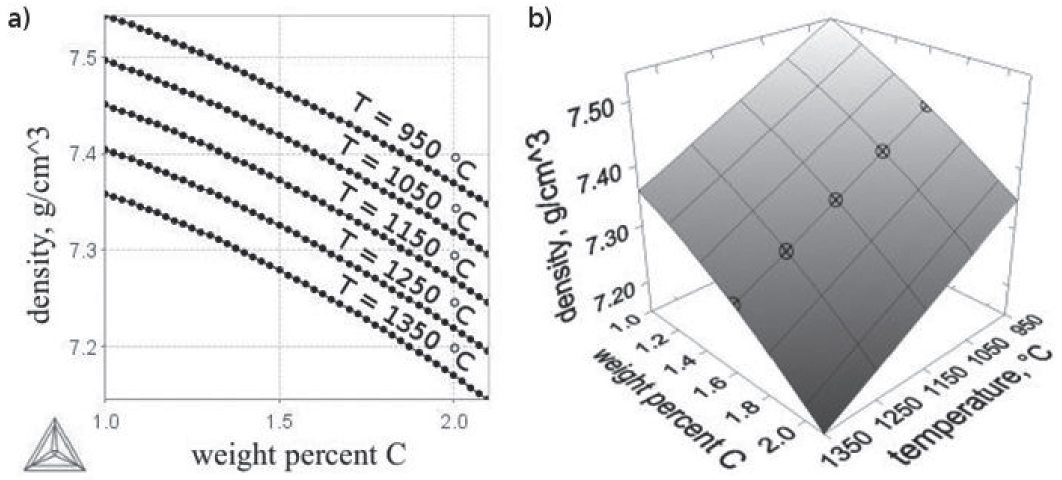

Fig. 3 Dependency of austenite's density on temperature and carbon concentration on 2D (a) and 3D (b) plot

2.2 Calculations involving use of TQ interface

For more complicated shapes of surfaces it becomes troublesome to derive formula to approximate them. **Fig. 4** presents such a surface. One can use mathematical software, e.g. STATISTICA or MATLAB to get the desired function. In order to get better adjustment, data can be split into few ranges but at some point, for even more complex surfaces this procedure will demand much effort to gain at the most fair results. Level of complexity increases further upon introducing new elements to the system, as new element means new variable in the sought formula.

Here with help comes programming interface which is very convenient tool that allows to solve this kind of problems. It was designed to aid in obtaining a great deal of data. It grants the ability of skipping part of deriving formulae to be implemented into the code, but instead the modelling program would get exact result directly from TC via the interface. In order to obtain those values however one needs to know how to get them using TC alone - the algorithm of reaching the point at which equilibrium is calculated and desired value obtained is the same.

To reach the goal of dividing TC-related work from the other part of the program, the need for designing set of procedures that would take over this mission arose. This can, naturally, be done by the means of procedural programming, but the modern approach of Object-Oriented Programming (OOP) is far more dependable, error proof, susceptible to development and more. It incorporates the idea of encapsulation - allowing the user of the class to use and see only few chosen procedures and constituents (e.g. variables). The literature about OOP is very rich, one is referred to [6,7,8,] for description of this technique used in C++ language.

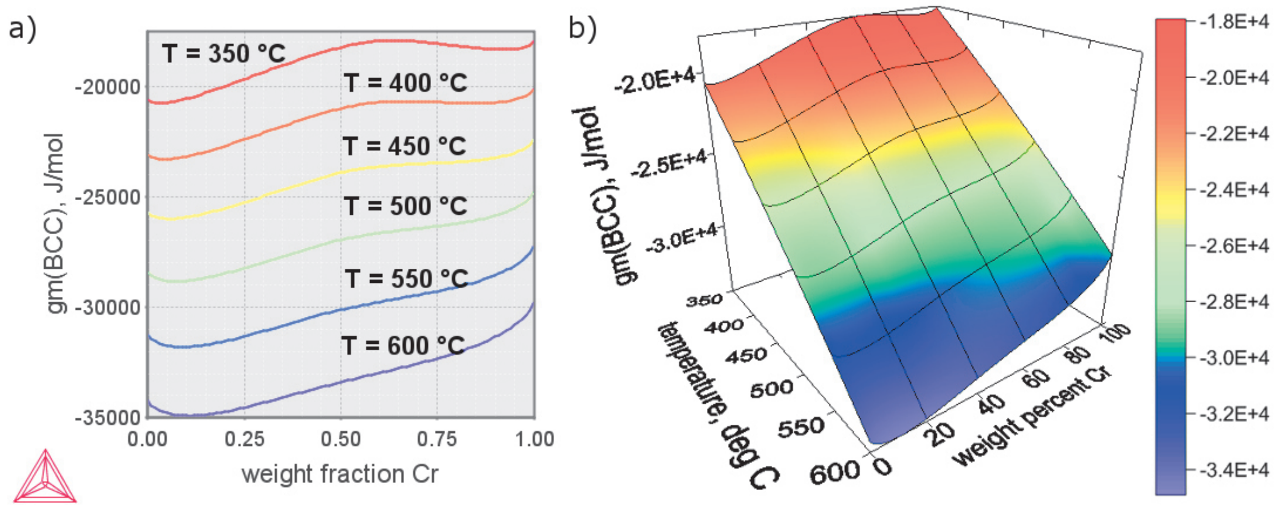

Fig. 4 Dependency of Gibbs Free Energy of BCC phase on wt. % Cr and temperature in Fe-Cr system

3. USE OF OOP CONCEPT IN MODELLING

The main aim of this work is to present and give the advantages of using the approach that makes user of the class independent, even unaware, of how the calculations and retrieval of data are performed. It is achieved by enclosing all of TQ-related subroutines into code of the class and leaving only few procedures useable (public) that would allow to unequivocally identify the system. This external functions are determined by the end user. It is he who decides how the arguments are sent and how wanted value(s) are retrieved. These are very features of OOP itself, the reason for stressing them here is due to, still high, popularity of Fortran language, among the scientists, which lacks the possibility of applying this approach.

Fortran subroutines of TQ interface have been successfully used in MICRESS software package. Until recently they were the only available procedures as C ones have only been introduced in 2013. Coupling of phase-field method with CALPHAD calculations by scientists from Aachen resulted in many publications, e.g. [9,10,11,12]. Thanks to presented concept, programmers with no knowledge of TC are enabled to profit from CALPHAD calculations. Also those who are involved in both: operating CALPHAD-based program and designing models are encouraged to apply this idea in work of their own. The authors of this paper find it very convenient way of developing the modelling software. Once tested and verified it becomes a reliable part of the program with minimal chance of an error to occur, making the whole easier to debug and shortens time needed to reach the final form.

3.1 Examples of classes which use TQ interface subroutines

For calculations of liquidus temperature in Fe-based system, from the interview with end user of the class, one could learn that accessible procedures should allow to declare what is the number of elements in the system, what are they and then carry out computations for varying compositions. The pseudocode of application of prepared class might look as follows:

```
    #include "C_TQ_liquidus.h"                // linking the file containing class code

  C_TQ_liquidus syst("C", "Si", "Mn");      // creating system of given components plus Fe

                     temperature = syst.get_liquidus(3.5, 1.0, 0.5);
                                       // calculation and return of liquidus temperature
                                        for given wt. % of previously declared elements
```

This example illustrates the simplicity of using such a class. Behind this few lines hide all of used TQ subroutines. The programmer, who is engaged with physical and mathematical side of model, in actually putting to use this liquidus temperature in his equations, does not need to know how it was obtained. He gets desired, chosen by him, form of procedures to call. Notice that someone not having any experience at all with CALPHAD-based programs can easily acquire needed.

The flexibility of this class is also worth mentioning. The number of alloying elements is not given directly, but program adjusts as number of arguments changes. The class has been prepared in such a way that the number of elements given in parenthesis can vary from 1 to 9, name of the procedures stays the same.
Similarly other classes that carry out computations in Fe-based systems were prepared, for calculations of chemical potentials in designated phase or retrieving phase's single property (for the list of properties see [12]. Pseudocode for calling procedures has the following form, for calculations of chemical potentials in liquid phase at 1450 K for multicomponent system:

```
                 C_TQ_chem_pot liquid("LIQ", "C", "Si", "S", "P", "Cu");

         pointer = liquid.get_chemical_potentials(1450, 3.2, 1.0, 0.02, 0.02, 0.1);
```

and volume 1 mol of FCC phase at 1500 K

```
                   C_TQ_single_property volume("V", "FCC", "C", "Si");

                     value = volume.get_value(1500.0, 1.0, 0.8);
```

All given examples are brought to visualize how friendly can implementing the CALPHAD calculations into modelling program be. The details, like names of procedures, number and order of arguments, units, method of receiving results are effect of collaboration between author and user of the class.
It can be noticed, that subroutines of presented classes have similar structure of calling. Since there is a set of TQ procedures common for all of them, it becomes extremely helpful to use yet another feature of OOP, namely inheriting. If not for the other aspects that just for this one it is worth to choose this approach when writing the modelling program. Having build tree of succession, changing code in one place will affect all of subsequent classes. Again, the reader is strongly recommended to refer to literature regarding OOP to find out about full list of advantages of using modern programming techniques.

CONCLUSIONS

The examples of data retrieval, using CALPHAD-based program, for modelling of solidification process have been demonstrated. It has been shown, that when acquiring not complicated relations it is sufficient to formulate obtained data and use derived functions in modelling program's algorithm. For more complex relations, mainly for those of more than two variables, the alternative route for getting data, namely employing TQ interface, was presented. Thanks to recent introduction of C subroutines into the interface library files, it has become convenient to use object-oriented programming with C++ rather than Fortran that's development stopped at procedural programming. The idea of designing classes as stand-alone tools that can be implemented in the modelling software and easily used even by scientists without any experience with TC was given. It has been stressed that with this concept programs are very flexible and making changes in the code is much more effortless. It is authors' hope that, after reading this paper, new users would choose and employ similar approach in their work.

This work was supported by project no. 18.18.170.437

REFERENCES

[1] Kaufman L., Bernstein H., Computer Calculation of Phase Diagrams, Academic Press, New York, 1970

[2] Spencer P. J., CALPHAD 32 (2008) pp. 1-8

[3] Lukas H., Fries S. G., Sundman B., Computational Thermodynamics: The CALPHAD Method, Cambridge Univ. Press, 2007

[4] Fraś E. Krystalizacja Metali, WNT, Warszawa 2004

[5] Wróbel M., Burbelko A., Gurgul D. Prace Szkoły Inżynierii Materiałowej, Kraków-Krynica, 2013, pp. X_X

[6] Lafore R., Object-Oriented Programming in C++ (4th Edition), SAMS, Indianapolis, 2002

[7] Balagurusamy E., Object Oriented Programming With C++, Tata McGraw-Hill, New Delhi, 2008

[8] Grębosz J,. Symfonia C++ standard, EDITION 2000, Kraków, 2008

[9] Rudnizki J., Böttger B., Prahl U., Bleck W., Met. Mat. Trans. A, Vol. 42A (2011), pp. 2516-25

[10] Böttger B., Apel M., Santillana B., Eskin D. G., Met. Mat. Trans. A, Vol. 44A (2013), pp. 3765-77

[11] Böttger B., Carré A., Eiken J., Schmitz G. J., Apel M., Trans. IIM Vol. 62, Iss. 4-5 (2009), pp. 299-304

[12] http://www.thermocalc.com/media/6030/tq8_usersguide.pdf, **Table 1**2, p 23