CALPHAD: Computer Coupling of Phase Diagrams and Thermochemistry 48 (2015) 18–26

Contents lists available at ScienceDirect

CALPHAD: Computer Coupling of Phase Diagrams and Thermochemistry

journal homepage: www.elsevier.com/locate/calphad

Set based framework for Gibbs energy minimization

Jeff Snider [a,*], Igor Griva [a], Xiaodi Sun [b], Maria Emelianenko [a,**]

[a] Department of Mathematical Sciences, George Mason University MS 3F2, 4400 University Dr, Fairfax, VA 22030, USA
[b] Department of Energy, Environmental and Chemical Engineering, Washington University, 1 Brookings Drive, St. Louis, MO 63130, USA

ARTICLE INFO

Article history:
Received 29 March 2014
Received in revised form
6 August 2014
Accepted 15 September 2014
Available online 23 October 2014

Keywords:
Mathematical modeling
Phase diagrams
Optimization
Automation
Miscibility gaps
Sublattices

ABSTRACT

A new unified approach to Gibbs energy minimization is introduced. While it has only been tested on binary and ternary systems so far, it has a built in capability of handling arbitrary multicomponent multiphase systems with any number of sublattices, miscibility gaps, order–disorder transitions, and magnetic contributions. This new unified AMPL set-based Gibbs energy description optimizes the data representation and makes it possible to subject the task of phase diagram calculation to numerous existing general purpose optimization strategies as well as custom-made solvers. The approach is tested on a variety of systems, including Co–Mo, Al–Pt and Ca–Li–Na, all known to be computationally challenging for other approaches. In most of the tested systems, the AMPL code reproduces phase diagrams obtained via Thermo-Calc. In other systems, in-depth comparison of results suggests that in prior work a sub-optimal equilibrium might have been identified as a global one, and re-evaluation of previously published diagrams and databases might be necessary.

1. Introduction

The last few decades have seen substantial development of algorithms and software for phase diagram calculations. These specialized software packages provide much needed functionality and capabilities for handling complex material systems. At the same time, the existing software packages for phase diagram calculation often rely on user input of initial conditions for convergence, and require special handling when it comes to formulations involving miscibility gaps and sublattices [1,2]. A number of remedies have been proposed in recent years, some of which have been quite successful at reducing the risks of running into suboptimal solutions, often with a steep cost premium [3,4]. Here a completely new methodology is proposed that allows a user to tackle these issues with the efficient use of the state-of-the-art constrained optimization algorithms that utilize the generalized form of the Gibbs energy minimization problem.

This paper introduces a *novel, set-based energy formulation*, equivalent to the traditional one, yet possessing a number of advantages in automation and flexibility. The new formulation allows the treatment of any multicomponent multiphase system involving an arbitrary number of sublattices, order–disorder transitions, and magnetic terms by means of a single formula. Automated miscibility gap detection is also achieved with no need for additional sampling as implemented by other existing methodologies [3,4].

The set-based approach serves as a bridge between Thermo-Calc-type databases and cutting edge optimization technologies which have become available in recent years. The approach allows the use of AMPL (a Modeling Language for Mathematical Programming) [5], a low-cost software environment which exposes the problem to a wide variety of general purpose optimization solvers as well as custom implementations taking advantage of the structure of the CALPHAD problem. This black box approach has the potential to reduce or eliminate the need for expert knowledge, enabling a simple and straightforward phase diagram evaluation interface for specialists and non-specialists alike.

In order to produce an AMPL compatible system description from a traditional database (TDB file), a system-independent converter was developed to extract all phase data from a TDB and feed it into the main AMPL model. This is not the first adaptation of Thermo-Calc data for further processing within a separate software module, or introduction of a framework capable of computing phase diagrams from Thermo-Calc-type descriptions. Notable existing work inspiring this effort includes the ESPEI infrastructure [6], OpenCalphad initiative [7], and the Gibbs module [8].

ESPEI is a self-optimizing phase equilibrium software package, which integrates databases (crystallographic, phase equilibrium, thermochemical, and modeled Gibbs energy data, etc) and database development (automation of thermodynamic modeling) with GUI (graphical user interface) designed mainly using Microsoft C# and SQL (structured query language). The data is stored in a matrix

* Corresponding author.
** Principal corresponding author.
E-mail addresses: jeff@snider.com (J. Snider), igriva@gmu.edu (I. Griva), xiaodi.daniel.sun@gmail.com (X. Sun), memelian@gmu.edu (M. Emelianenko).

http://dx.doi.org/10.1016/j.calphad.2014.09.005
0364-5916/

format specific to ESPEI and allows for automation of database development [6]. OpenCalphad is an open-source code for performing thermodynamic calculations using the Calphad approach. The code implements several different thermodynamic models which allow the description of thermodynamic state functions, such as the Gibbs energy, as a function of temperature, pressure and composition [7]. Finally, Gibbs module is a tool hosted by nanoHUB [9] that enables rapid prototyping, validation, and comparison of thermodynamic models to describe the equilibrium between multiple phases for binary systems [8].

The approach presented herein complements these efforts and is not meant to introduce yet another solver, or compete with existing packages in terms of speed. It is designed as a research tool allowing scrutiny of phase diagram calculation from a different viewpoint, making it possible to subject the problem to numerous existing general purpose optimization strategies as well as custom-made solvers, revealing similarities and differences between resulting phase diagrams. As seen in Section 6 below, preliminary benchmarking tests suggest that this methodology can lead to some interesting observations when it comes to systematic analysis of phase diagrams obtained by other methods. In particular, new insights into phase stability were obtained by applying the new methodology to several multicomponent test systems containing miscibility gaps, multiple sublattices, and order–disorder transitions. The AMPL based approach precisely mirrored the Thermo-Calc phase diagram for the Co–Mo system, however for the Al–Pt case a lower Gibbs energy associated with a different phase combination was found than that reported by Thermo-Calc using the default settings. Similar findings have been reported earlier, e.g., in [10] for the systems Si–Tb, Al–Nb, Co–Si, and others. While elucidating the source of these discrepancies is outside of the scope of current work, the comparative analysis presented in Section 6 gives evidence to certain inconsistencies in the treatment of Gibbs energies that might warrant further investigation.

The information flow is depicted graphically in Fig. 1. The converter, model, and mapper are "black box" codes which do not depend on the system being investigated. The converter is written in Java, while the model and mapper are implemented in the AMPL modeling language. The model is a description of the problem for any system, completely independent of the number and choice of elements or species. It is described in detail in Sections 3 and 4.

Given a TDB file for a system, the converter generates data and parameter files in the AMPL modeling language. The mapper is an AMPL script which reads the model and data and writes out phase information by temperature and composition. This output constitutes the data for a phase diagram, which is converted (for this paper using MATLAB) into a graphical phase diagram.

The paper is organized as follows. A short description of the AMPL environment is given in Section 2. The standard problem formulation given in Section 3 is followed by the description of the novel set-based formulation in Section 4, with the discussion on the handling of the miscibility gap given in Section 4.6. The details of the use of the novel formulation for the automation of phase diagram calculation are provided in Section 5. Results of numerical tests are shown in Section 6, followed by discussion.

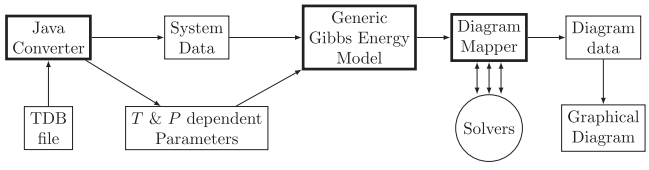

Fig. 1. Data flow.

2. AMPL environment

AMPL is a modeling language and software package providing a unified front-end for an extremely wide variety of solvers, such as LOQO [11], MINOS [12], and SNOPT [13], among others [5,14]. The user can describe the model and separately provide the data, making reuse of the model for alternative systems straightforward. AMPL easily handles any combination of linear and nonlinear objective functions and constraints, and selection of the best solver is left to the user. The "student" version of AMPL with limitations on the number of variables and constraints is freely available online, and a professional version is available for a modest fee. For academic use there are free online services where a job can be submitted with no restriction on size, and the result is returned via email. Many of the most powerful solvers are freely available online. A robust set of documentation is available, and a widespread group of users and AMPL developers actively discuss challenges and advance the state of the art.

In the AMPL modeling language, data is primarily set-based with native commands for set operations on scalars, textual values, and tuples of arbitrary length. Summation, multiplication, and other operations with sets as indices are natural. Set and parameter values may be explicitly defined in the data, or be transparently computed according to their definition in the model when the data is read or updated. It has to be noted that set-based formulation has advantages over other structures traditionally used to store data (e.g. vectors or matrices) due to its flexibility in defining the number and the names of set components. The space is automatically preallocated. The only task left to a user is providing the sets and parameters defined on those sets and then the model will automatically generate the Gibbs energies and the constraints based on this information.

3. Formulation of Gibbs energy minimization problem

First recall the standard Gibbs energy minimization problem formulation [15–17]:

$$\begin{aligned}
&\min_{f,y} G = \sum_p f^{(p)} G^{(p)}(y) \\
&0 \le f^{(p)} \le 1 \quad \text{for each phase } p \\
&0 \le x_e^{(p)} \le 1 \quad \text{for each phase} - \text{species pair } (p,e) \\
&0 \le y_{s,e}^{(p)} \le 1 \quad \text{for each phase} - \text{species} - \text{sublattice triplet } (p,e,s) \\
&\sum_e y_{s,e}^{(p)} = 1, \quad \text{for each phase} - \text{sublattice pair } (p,s) \\
&\sum_p f^{(p)} x_e^{(p)} = f_e^0, \ \text{for each species } e, \quad \text{such that } \sum_e f_e^0 = 1
\end{aligned} \tag{1}$$

Here $x_e^{(p)}$ indicates the mole fraction of species e within the phase, $y_{s,e}^{(p)}$ is the site fraction of species e in sublattice s within the phase, $a_s^{(p)}$ is the site ratio of sublattice s in the phase, and $f^{(p)}$ indicates the mole fraction of the phase in the overall composition. Similar to [15], rather than being the subject of a constraint, x_e is defined as a function of $y_{s,e}$ in the present implementation:

$$x_e^{(p)} := \frac{\sum_s a_s^{(p)} y_{s,e}^{(p)}}{\sum_{c \neq \mathrm{Va}} \sum_s a_s^{(p)} y_{s,c}^{(p)}}. \tag{2}$$

This reduces the number of variables and constraints, and for the solvers tested it results in a modest speed improvement.

The overall number of variables and constraints in this problem depends entirely on the system being examined. The smallest system potentially of interest would have two species and two phases each with a single sublattice, hence six variables and a