Accelerating Complex Chemical Equilibrium Calculations - A Review

Willem A. Roos[a,*], Johannes H. Zietsman[a,b]

[a]*University of Pretoria*
[b]*Ex Mente Technologies*

Abstract

Incorporating multicomponent, multiphase, high-temperature, complex chemical equilibrium calculations into process and multiphysics models can provide significant insights into materials, processes and equipment. We refer to applications where the inclusion of these calculations provided insights that would otherwise be difficult to obtain. From these examples, the advantages and importance of including complex equilibria into models are clear for cases where more accurate descriptions of practically relevant systems are needed.

Equilibrium calculations are, in general, omitted or incorporated in a simplified manner due to their computational expense. The equilibrium state of a complex chemical system is determined by minimising the Gibbs free energy for a given set of system component concentrations, temperature, and pressure. This minimization routine is computationally expensive and makes direct integration of chemical equilibrium calculations into models infeasible.

There have been many attempts to, in one way or another, accelerate these calculations. This includes methods such as creating look-up tables prior to the simulation or in-situ, fitting piecewise polynomial functions to thermochemical properties, phase diagram discretisation, sensitivity derivatives, machine-learning algorithms, and parallelisation. Pre-calculated databases tend to become very large and require much storage space, even when unstructured grids are used or piece-wise polynomials fitted. Neural network results do not adhere to physical laws such as mass conservation and large training sets are required to reduce this error.

In-situ or on-demand methods of creating a database shows great promise because only the thermochemical regions that are of interest to the model are captured in the database, reducing the storage size and the amount of data to search through. No prior knowledge of the system is required to create the database.

The Gibbs phase rule can be used to determine which geometrical features of a phase diagram to discretise and create a sparse database that covers large temperature, pressure and compositional ranges. The lever rule can then be used for fast and accurate interpolation between data points. Established thermochemical theory provides security for the decisions made within the discretisation and interpolation algorithms. Based on this review, an in-situ phase diagram discretisation method strongly based on thermochemical theory such as the Gibbs phase rule and the lever rule holds potential for significant acceleration of complex chemical equilibrium calculations.

Keywords: complex equilibrium calculations; CALPHAD; acceleration; multiphysics models; process models

1. Introduction

Incorporating multicomponent, multiphase, high-temperature, complex chemical equilibrium calculations (equilibrium calculations) into multiphysics and process models (models) can provide valuable insight. Equilibrium calculations provide phase fractions, phase compositions, heat capacity, and enthalpy as functions of temperature, pressure, and composition. Material property models can estimate additional properties such as viscosity, density and electrical conductivity based on equilibrium calculation results. Although this approach can provide more accurate material properties to models, equilibrium calculations are computationally expensive and are usually omitted or incorporated in a simplified manner. As seen in Figure 1, there is a high-order non-linear relationship between a system's number of components and equilibrium calculation time – can become computationally expensive if large number of calculations have to be performed. Although other software than ChemAppPy[1] can be used to perform equilibrium calculations of the systems presented in Figure 1, and show different performances, the trend will remain the same; the more system components considered, the more computationally expensive the equilibrium calculations, and unfortunately, it does not scale linearly.

Multiphysics models can have thousands or even millions of mesh cells and an equilibrium calculation may need to be performed in each cell for every iteration [2] – a significant fraction of the simulation calculation time is spent thereon [3]. When the 10-component system is considered at roughly 17 s per equilibrium calculation, as shown in Figure 1, and a multiphysics model with one million mesh cells have to be solved, the total solving time of a single iteration can take up to 200 days. Multiple iterations are, however, required for convergence and even more when a transient model is solved. When only 10 iterations of this multiphysics model are solved, almost 5.5 years would have passed. In cases with many system components, mesh cells, iterations, and time steps, direct integration of equilibrium calculation software into a model to perform equilibrium calculations at a specified system state and provide thermochemical
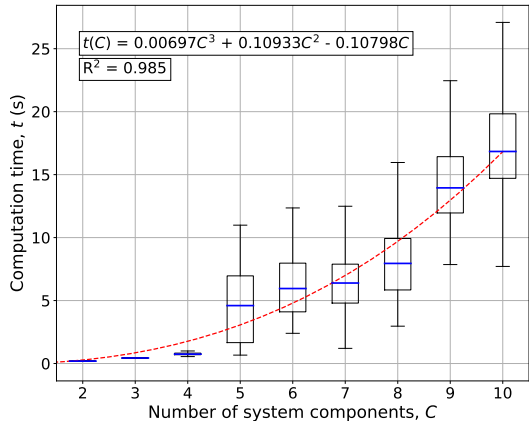

Figure 1: Third-order increase in equilibrium calculation time with number of system components. Initial system was Fe-O, and Si, Al, Mg, Ca, N, Mn, Na, K were added. 250 randomly distributed equilibrium calculations per C-component system. Calculated with ChemAppPy[1], and data from the FToxid database in FactSage 7.2.0. The calculations were done on an Intel Core i7-3770 with a clock speed of 3.4 GHz and had 8 GB of RAM available.

data to the model (simply referred to as direct integration) can result in infeasible solving times.

The same high-order non-linear relationship between a system's number of components and equilibrium calculation time is seen in process models. The difference is that these models do not have mesh cells to consider, but still have many equilibrium calculations that have to be performed each iteration for various chemical reactions that are included in the model. Many iterations may be required to ensure convergence, and even more when a transient models is solved. Again, many system components, iterations, and time steps, direct integration can result in infeasible solving times.

Models that use direct integration with reduced numbers of components may be feasible, but may not show details that more system components can. Several methods have been developed, as discussed in Section 4, to improve the computational efficiency of including equilibrium calculations into models so that more representative and accurate simulations can be

2

performed in a reasonable time. These methods either improve the efficiency of equilibrium calculations or store calculation results for later recall and interpolation, which is less costly than direct integration.

Conceptually, an equilibrium calculation accelerator, together with result data stored in its database, acts as an intermediary between a model and equilibrium calculation software, as illustrated in Figure 2. The result database can be populated prior to solving the model, or while the model is being solved (in-situ). The accelerator recalls stored data and then interpolates to determine thermochemical and physical property values requested by the model at a given system state rather than performing a more computationally expensive equilibrium calculation.
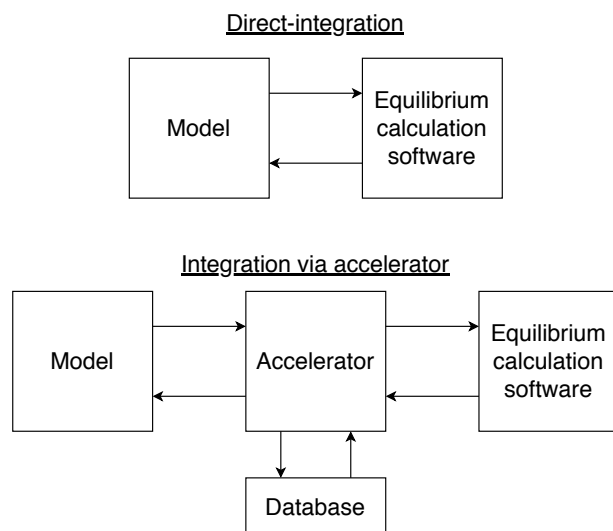

Figure 2: Schematic representation of direct and indirect integration between models and equilibrium calculation software.[4]

In this article, we first discuss equilibrium calculations in short and why they are so computationally expensive, whereafter we look at a few cases where inclusion of equilibrium calculations in models provided valuable insight, showing the importance of including equilibrium calculations into models. We then review methods found in literature developed to improve the efficiency of equilibrium calculations and make it more feasible to include them into mul-

tiphysics and process models.

2. Equilibrium Calculations

Spontaneous physical change is governed by the second law of thermodynamics, which states that such change will only occur when it results in an increase in an isolated system's entropy. For constant-pressure systems, Gibbs free energy (or Gibbs energy) is a useful quantity that allows us to describe entropy change in terms of properties of the system alone; we can conveniently disregard the surroundings. Thermochemical equilibrium is reached when Gibbs energy reaches a minimum; also the state of maximum entropy.

This physical optimisation problem can be cast into mathematical and numerical formulations, and ultimately into computer software that we can use to simulate a system's drive towards equilibrium. This is however an iterative process and, as displayed in Figure 1 become more computationally expensive as more system components are considered and the optimisation problem become more intricate.

Many different methods have been developed to calculate thermochemical equilibrium by determining the minimum Gibbs energy for a system state. Simultaneous and decoupled equation-solving methods have been developed and used. Teh and Rangaiah [5] discussed and evaluated the various equation-solving methods in more detail. Piro et al. [6] employed another method, Gibbs energy partitioning, in the Thermochimica library. Convex hulls have been used as a geometrical approach to determine the state with minimum Gibbs energy [7]. The most popular method, however, has been the Gibbs energy minimization routine, which has been implemented in a variety of open-source[8, 9, 10, 11] and commercial[12, 13, 14, 15, 16] software packages. The OpenCalphad software also provides the capability to perform multiple equilibrium calculations in parallel [9].

3. Applications

In the development of models, it is often adequate to make simplifications regarding thermochemical be-

3

haviour and physical properties. In other cases it is essential to incorporate these details, since they are core to the investigation. Here we focus on the latter.

3.1. Multiphysics Models

Incorporating equilibrium calculations into models have been used extensively in casting design by studying the mechanisms contributing to macrosegregation [17]. The solidification, shrinkage, air-gap development, stresses, and temperature distribution of a casting have also been modelled by incorporating equilibrium calculations into models [18].

Refractory materials are often subjected to high temperature gradients, corrosive environments, and mechanical loads [19] in an attempt to protect equipment or operators nearby. Models have been used to investigate different materials and how they degrade due to oxidation, corrosion, vaporization, sublimation, ablation, and dissolution [19], [20]. There are clear signs in the work done by Blond et al. [19] that the inclusion of equilibrium calculations improves insight into stresses and strains in furnace refractories. Equilibrium calculations have been included into models with adaptive meshes by Tabiei and Sockalingam [20] to investigate the lifespan of dynamic refractory linings of hypersonic vehicles during their re-entry through Earth's atmosphere.

Biomass-based energy technologies have been improved significantly when equilibrium calculations were incorporated into multiphysics models [21]. Multiphase reactors are widely used, but their designs are primarily based on experimental data due to process complexity. The understanding of these processes is limited because dense and erosive flows encumber measurement tools. Even when measuring tools are used, system dynamics are altered, leading to measurement inaccuracy [21]. Models that incorporate equilibrium calculations can bridge the gap between experimental data and the actual behaviour of these reactors [21], which can be used to improve reactor design.

Equilibrium calculations allowed phase compositions, oxygen chemical potential and other thermochemical properties to be included into models that were essential for investigating and estimating performance, safety, and behaviour of nuclear fuel that could not be determined through experimentation during operation [22], [23].

Complex chemical reactions, heat and mass transfer, phase changes, and multiphase flows make it extremely challenging to model pyrometallurgical processes. Molten slags are usually produced, in some cases it is the desired product [24], and can be very corrosive, which can cause refractory lining damage. To combat this, furnace sidewalls are cooled with the intent to solidify slag onto the refractory lining, forming a freeze lining [25]. Including equilibrium calculations into models make it possible to describe chemical reactions, heat and mass transfer, phase changes, and multiphase flow, which can then be used to estimate freeze lining condition.

3.2. Process Models

Process models of ilmenite-smelting DC furnaces[24, 25], have been used to investigate the interaction between slag bath and freeze lining. This was done by determining the influences of thermal and chemical changes and changes in operational parameters on these interactions. Insights from these investigations were used to improve and refine the operational strategies of the process.

There is continual interaction between the flow of molten slag and freeze lining as remelting and solidification occurs. It is important to maintain a freeze lining at all time, but it is near impossible to measure the geometry within molten liquid to ensure its presence. Monitoring of the freeze lining thickness can be done by following the trends of thermocouples in the sidewalls [26]. The thermocouple data is used to calculate heat flux through the freeze lining and estimate its thickness.

The electric arc furnace (EAF) is the main process used to recycle iron and steel and the second-most important process in terms of global steel production [27]. Process models have been used to improve the understanding and control of the EAF process as well as optimizing its energy and resource efficiency [28]. These EAF process models include equilibrium calculations that provide composition and temperature estimations throughout the process that cannot be measured directly due to the harsh conditions within the furnace.

EAF process models can be used to automatically create operational charts in real-time and can be adjusted based on furnace operating conditions [27]. Scenarios such as new control strategies, different injection or charging materials, and installation of new equipment can be evaluated virtually and adjusted with a process model before any changes are implemented. Hay et al. [27] and Hay et al. [28] summarised the different EAF process models that have been developed and their continual improvements.

3.3. Conclusion

Being able to include equilibrium calculations into a model that accurately represented the thermochemical system allowed for virtual prototyping and testing of new processes, control strategies and equipment before any costly physical tests had to be performed, new plants had to be built or changes made to existing plants, or new equipment purchased. Using models were equilibrium calculations have been included to monitor and control processes were conventional monitoring methods are infeasible have been found to be invaluable. Discussed here were only a few examples of where the implementation of equilibrium calculations provide insight into systems and processes that would otherwise be difficult or impossible to obtain.

4. Accelerating Equilibrium Calculations for Modelling

The incorporation of equilibrium calculations in models can provide more accurate results, but at a cost of longer computational time. To reduce equilibrium calculation times in models and make solving times more feasible, several acceleration methods have been developed. The golden thread that runs through the majority of these methods is based on a pre-calculated or in-situ calculated database of the specific thermodynamic system and utilising the stored data in an interpolation routine to accelerate equilibrium calculations.

4.1. Uniform Discretisation of System Space

The feasibility of uniformly discretising the temperature and compositional space of a thermochemical system was investigated by ten Cate et al. [17]. At each discretisation node, an equilibrium calculation had to be performed and thermochemical properties stored for later recall. The composition in a system with C number of components can be described by the concentration of $C-1$ components. An isobaric 4-component system was considered with varying temperature. Therefore, a thermodynamic property of the 4-component system was dependent on four independent variables; temperature and the concentration of three components. The system range of each independent variable was divided into 600 nodes to form a uniform grid. To store two thermodynamic properties as functions of the four independent variables with single precision (32 bit per value) on each of the grid nodes, $2 \times 4 \times 600^4 \times 32\,\mathrm{bit} = 4\,\mathrm{TB}$ of storage would have been needed [17]. The size of a uniform grid pre-calculated database can become too large for the memory of computers [17] to use and searching through that amount of data can be very time-consuming. The space complexity is of order $O(n^d)$ where d is the dimension of the grid [17].

A similar tabulation method was developed by Saad et al. [29]. A known alloy was chosen and the composition variation intervals were identified. These intervals were not known before the model was solved and had to cover the alloy composition extremes – informed estimations had to be made. A temperature interval also had to be identified, but in general was chosen from the initial melting temperature to room temperature. Systematic checks were done within these intervals at predefined step sizes and equilibrium calculations were performed at each temperature and compositional combination. The number of phases, together with the fractions and compositions of each, were determined from each equilibrium calculation. This allowed the temperature and compositional interval of the phase region boundary associated with each phase to be determined. Thereafter, the composition and temperature intervals of each phase was systematically checked to determine thermochemical properties of the phase at different temperatures and compositions. Interpolation can

then be performed between phase composition and temperature nodes to obtain phase properties for an equilibrium calculation.

4.2. Liquidus Surface Mapping

A method was used to store the liquidus and solidus surfaces of solid and eutectic phases in a 3-component system [30], [31]. A number of equilibrium calculations were done prior to the simulation of the model for combinations of various temperatures pressures, and system component concentrations to be used to describe the surfaces.

The concentration of one component was kept constant and the liquidus surface mapping along the isopleth was completed by ranging the other component between its minimum and maximum concentration by a fixed increment. Two partial slopes of the liquidus surface; $m_{l,A} = \partial T/\partial c_{l,A}$ relative to system component A concentration and $m_{l,B} = \partial T/\partial c_{l,B}$ relative to system component B concentration, and two partition coefficients; k_A and k_B were stored for a fixed concentration $c_{l,C}$ of component C. Partition coefficients are used to describe tie-lines between the solidus and liquidus surfaces [31]. The first system component's concentration was then adjusted by a fixed increment and the mapping repeated.

The primary phase's liquidus temperature, T_f, together with the liquidus surface's slopes, $m_{l,A}$ and $m_{l,B}$, are used to describe the liquidus surface for a given composition, as described by Equation (1) and illustrated with Figure 3, [31].

$$T = T_f + m_{l,A} \times c_{l,A} + m_{l,B} \times c_{l,B} \qquad (1)$$

Given the phase composition on the liquidus surface, the corresponding phase's composition on the solidus surface – that is found on the same tie-line – is described by the partition coefficients, as described by Equation (2) and illustrated in Figure 3, [31].

$$c_{s,A}^* = k_A \times c_{l,A}^*$$
$$c_{s,B}^* = k_B \times c_{l,B}^* \qquad (2)$$

With this method a global mapping file was compiled. Bilinear interpolation was used to calculate
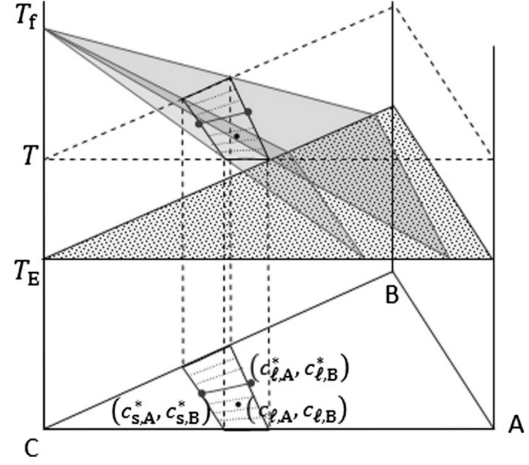

Figure 3: Liquidus surface of two-phase region described by stored partial derivative values of temperature with respect to system component concentrations. Tie lines described by stored partition coefficients used to determine the solidus surface. Taken from [31]

the liquidus temperature, the slopes of the liquidus surface, and the partition coefficients for given component concentrations.

An acceleration factor of 4 was observed in 3-component systems with a mapping filesize of a few MB [30]. When a mapping file was created in a 4-component system, the filesize was found to be in the order of 100 MB. As seen in the feasibility study on uniform discretisation of temperature-compositional space done by ten Cate et al. [17] and the liquidus and solidus surface mapping by [30], as more complex systems with more components are considered, more storage space is required and can become infeasible.

4.3. In-situ Tabulation

Instead of performing a large number of equilibrium calculations and tabulating the results prior to solving a model, an initially empty table can be populated as the model is solving – in-situ – as presented by Pope [32]. As the model requires thermochemical properties at a specified system state, an equilibrium calculation is performed, and results stored to the table. At a later stage, when the model requires

6

thermochemical properties near stored results, interpolation, rather than an equilibrium calculation, is performed and the results returned. The advantage is that the table is populated only in regions of the system space that is accessed by the model, known as the accessed regions, and no unnecessary equilibrium calculations have to be performed beforehand. This reduces the number of computationally expensive equilibrium calculations that need to be performed and reduces the storage requirement of the table.

It was found that the use of in-situ adaptive table completed 10^9 queries in $54\,h$, compared to more than 6 years (theoretically) when direct integration was considered – an acceleration factor of about 10^3. It roughly took $1.5\,h$ of the computational time to complete all the additions to the table and in the remaining computational time only table retrievals were made and interpolation performed [32].

Analysis of the in-situ adaptive tabulation method's performance as well as an improvement to the searching algorithm was done by Chen [33]. An improvement on the algorithm was made by Lu and Pope [34] in terms of the table-searching strategies and the addition of error checking and correction. In comparison to the previous implementation ([32]), the computational time has been halved and the storage requirements have been reduced by a factor of five.

In-situ tabulation was also used in the work of Larsson and Höglund [35] where the system space was divided into a structured reference frame but no equilibrium calculations were performed beforehand – only the nodes' temperature and compositions were determined. When thermochemical properties were required at a specified state, the nodes of the structured reference frame nearest to the specified state were identified, equilibrium calculations performed at their corresponding temperature and compositions, and the results stored to the database. From the results obtained, interpolation is performed to determine thermochemical properties at the specified state. When another specified state is found between reference frame nodes where equilibrium calculations have been performed, they are simply used again to interpolate towards the specified state. In the case where some identified reference frame nodes

do not have any results available, equilibrium calculations are performed, results stored to the database, and interpolation performed. A database storage size limit is also specified and when new results have to be stored, the results that has not been used for the longest period is overwritten.

The in-situ tabulation scheme of Larsson and Höglund [35] was implemented by [3] and its performance evaluated. In one test, an acceleration factor of 2 was observed. In another test, the compositional space accessed by the model was larger and therefore more equilibrium calculations were required, and an acceleration factor of 0.6 was observed.

As noticed in the work by Pope [32], initially only computationally expensive equilibrium calculations are performed and little to no acceleration is observed, or even a deceleration due to the extra overhead computations required to tabulate the data. As the table is populated more densely, the frequency of recalls and interpolations increase up to the point where only recalls and interpolation is performed and large accelerations are observed.

4.4. Parallelisation

Although continuous improvement of the different algorithms reduce the computational time of equilibrium calculations, another method of achieving large acceleration is by performing these calculations in parallel across multiple central processing unit (CPU) cores. A comparison between single-core and parallel computed equilibrium calculations were performed [3] where a number of equilibrium calculations were performed with the Thermo-calc [15] software. These equilibrium calculations were divided equally between a number of CPU cores with the MPI (Message Passing Interface) protocol and a close-to-linear scaling was observed [3]. This scaling continued up to a number of cores where the time required to transfer data between all cores became comparable to equilibrium calculations themselves – no longer advantageous to use more cores.

In one case where the equilibrium calculations were less computationally expensive, a linear scaling was found up to 24 cores. In another case where the equilibrium calculations were more computationally expensive, a linear scaling was found up to 48 cores.

Acceleration of equilibrium calculations are possible by dividing them amongst numerous CPU cores, but there are limits to the scaling and it differs from case to case. This method focuses on distributing the equilibrium calculations between a number of cores rather than storing the data for later recall and potentially avoiding performing each equilibrium calculation.

The in-situ tabulation and interpolation scheme of Larsson and Höglund [35], as discussed in Section 4.3, was implemented and parallelised with the MPI protocol by Pillai et al. [3]. Performance tests were done on 4, 12 and 48 cores where the parallelised in-situ tabulation and interpolation scheme was compared against direct equilibrium calculations on the same number of cores. The one test showed that the parallelised in-situ tabulation and interpolation scheme was about 40 % to 50 % faster than parallelised direct equilibrium calculations. In another test it was 25 % to 50 % slower as the compositional space accessed by the model was larger and therefore more equilibrium calculations were required to populate the database [3]. In either case, acceleration was achieved in comparison to single-core calculations, although the scaling is less than linear.

Some CPUs can have as many as 48 cores or even more, but graphics processing units (GPUs) can have much more; where it is not uncommon for GPUs to have several thousands of cores. Although a GPU has a substantial amount of cores more than a CPU, its cores has much less on-chip storage capabilities, a smaller instruction set, and slower clock speeds. This means that a single GPU core can not perform as computational demanding calculations as a CPU core and is not as fast as a CPU core, but because of the vast amount of cores at its disposal, a GPU can massively parallelise operations and perform them much quicker than a CPU can.

GPU parallelisation of equilibrium calculations was done by Gandham et al. [36]. To avoid latency in the equilibrium calculation being performed on a CPU or GPU, the data required to perform the calculation is stored within the on-chip cache which is present on both CPUs and GPUs. However, the cache size on a CPU is in general much larger than that of a GPU. As the number of system components increase, so does the number of registers required to store the data required to efficiently perform an equilibrium calculation. On a GPU, when this exceeds the maximum number of registers per thread, the GPU's dynamic random access memory (DRAM) is used for storage, leading to a reduction in performance. In contrast, the cache of a CPU is large enough to store all the data required for an equilibrium calculation for the test cases considered. To reduce the demand for the GPU on-chip resources, a mixed-precision approach was taken where equilibrium calculations benefited from the performance of single-precision floating point computations and the accuracy of double-precision computations; majority of computations are performed with single-precision, requiring less on-chip storage, and double-precision only used to refine the solution to the desired accuracy when needed. It was found in the test cases that only 1 % to 2 % of equilibrium calculations that were performed required double-precision calculations for refinement. An acceleration factor of 3 was observed when this mixed-precision approach was used compared to using only double-precision computations for every equilibrium calculation.

To compare the performance between CPU and GPU parallelisation, Gandham et al. [36] implemented the same algorithm for both processing units. The CPU used was a Intel Xeon E5-2630 v3 with eight cores clocked at 2.4 GHz and the GPU was an NVIDIA Tesla K80 board. Across two test cases, the average acceleration factor obtained by the GPU compared to the 8-core CPU (also performing equilibrium calculations in parallel) ranges from 5 for systems with 30 components to 17 for systems with 6 components.

4.5. Polynomial Fit to Thermodynamic properties

Thermodynamic properties can display discontinuities, as seen in Figure 4, with varying temperature as well as variations in system component concentrations. Instead of using a uniform grid to capture thermochemical property data, less nodes can be used if only these discontinuities are captured and stored [17]. However, the locations of these discontinuities need to be known.

A diagram showing the compositions of all stable phases of a thermodynamic system present at equi-

8

librium as a function of system component concentrations, temperature, and pressure is referred to as a phase diagram. A phase diagram is a geometrical representation of a thermodynamic system [37]. Locations of the observed discontinuities in thermodynamic properties correspond to phase region boundaries found on a phase diagram, and when grid nodes are adapted to the phase diagram shape in an unstructured non-uniform manner, the number of nodes used to discretise the temperature and compositional axes are expected to decrease.
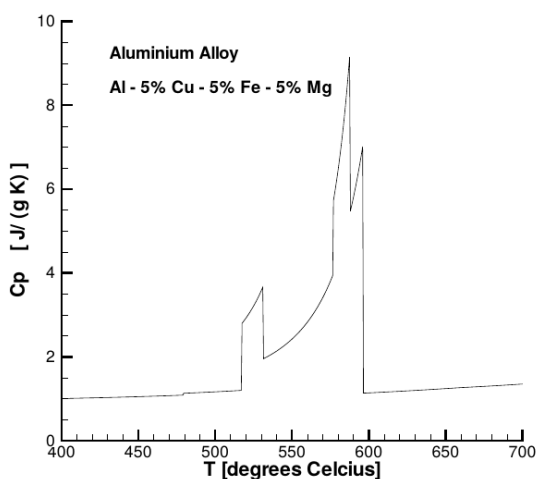

Figure 4: Dependency of heat capacity on temperature for an Aluminium alloy with fixed composition of 5 % Copper, Iron, and Magnesium. Taken from ten Cate et al. [17]

The positions of discontinuities seen in Figure 4 were stored and first-order piecewise polynomials fitted between those nodes. First-order polynomials might not have been the most accurate simplification, but with the addition of only one node a more accurate second-order piecewise polynomials could be used.

To discretise and store the heat capacity, as depicted in Figure 4, for varying temperature and a fixed composition, only 17 nodes were needed if first-order piecewise polynomials were considered and 23 nodes for second-order piecewise polynomials [17]. A reduction factor of around 30 was found for stor-

age requirement when this method was implemented (without a significant loss in the accuracy) in comparison to the uniform grid approach where 600 nodes were used to discretise the temperature axis – discussed in Section 4.1. The mapping was done for a fixed composition, but the composition will seldom be constant throughout the entire domain and model. The method of fitting piecewise polynomials to thermodynamic properties was therefore applied at a number of different system component concentrations. Instead of discretising the concentration of each component into a uniform grid, unstructured meshing was applied by clustering more nodes at important concentrations [17]. It was estimated that a reduction factor for storage requirement of about 3 can be obtained by applying this method to the discretisation of a single component's concentration range [17].

These reductions were applied to the same 4-component system discussed in Section 4.1 where a uniform mesh was used and the database ended up to be in excess of 4 TB in size. If a reduction factor of 30 was found for the temperature axis and a factor of 3 for each composition axis, a total reduction factor of $30 \times 3^3 = 810$ was found. This meant that a 4 TB database would be reduced to about 5 GB [17].

Using a phase diagram to intelligently decide where to store thermodynamic data, rather than using a large uniform mesh, reduces the storage requirement tremendously. This reduction makes the use of such a pre-calculated database more feasible but systems with more components would still require a large amount of storage space.

4.6. Polynomial Regression of Phase Region Boundaries

Regression has been used to fit polynomials to data points on phase region boundaries [38]. Using polynomials and linear interpolation is much less computationally intensive than Gibbs free energy minimization used in equilibrium calculations [38].

Equilibrium calculations were done at different concentrations of system components to obtain the liquidus and solidus temperatures of the phase region. Polynomial functions were fitted to these points

to capture the phase region boundaries. The polynomial was fitted to the data points calculated from within the phase region, but the function could also be used with a set of independent variables outside the range of fitted data. The polynomial used to describe the phase region boundary therefore had to be bounded to capture the phase region boundary limits. Polynomials of a lower order were used to capture these limits.

This regression method was tested in the Al-Si-Mg-Fe system and compared against the thermochemical calculation software Thermo-Calc [15]. It was found that the liquidus and solidus temperatures determined by this method only differed by fractions of a degree Celsius to that calculated by Thermo-Calc [38]. Zhao et al. [39] used a regression method in the Al-Cu-Si system. Compared to Thermo-Calc, the method had a maximum temperature error of $1.37\,^{\circ}\text{C}$ and less than a percentage error on any of the phase compositions. The direct integration with Themro-Calc took $3.66\,\text{h}$ compared to the regression method that only took $147.54\,\text{s}$ – an acceleration factor of almost 90.

4.7. Phase Diagram Discretisation

As discussed previously in Section 4.1, the number of nodes needed to represent the phase diagram become infeasible when uniform grid meshing is done, especially for high-order systems. Meshing of a phase region, shown in Figure 5, was performed with a mesh generator developed by ten Cate et al. [17]. Non-uniform meshing, shown on the right, requires fewer nodes in comparison to a uniform mesh, shown on the left. A distance and size function were used to determine the desired edge length of a mesh cell depending on its distance to the nearest phase region boundary. This allowed for a non-uniform adaptive mesh to be applied to the region instead of a uniform mesh, reducing the number of cells needed to still accurately capture properties within the phase region. Equilibrium calculations were performed at each of the nodes and thermochemical properties stored where interpolation could then be used when properties were requested by the model.

Fewer nodes are needed with non-uniform meshing to describe the phase region and will reduce the stor-age requirement when thermodynamic properties are stored at every node, seeing that the storage requirements are of order $O(n^2)$ where n is the number of discretisation cells [17].
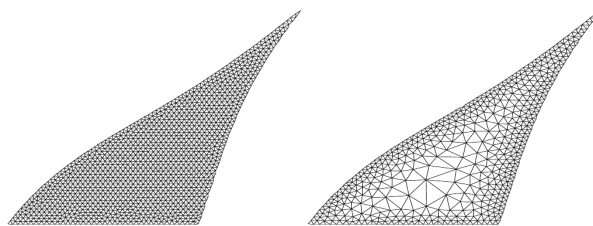

Figure 5: Meshing of the liquid-Pb phase region. Left: Uniform mesh size function. Right: Distance dependent mesh size function. Taken from ten Cate et al. [17].

The meshing of phase regions and the storage of thermochemical data could be used effectively, but the meshing of all phase regions are not necessary. The phase fraction of all stable phases at equilibrium can be calculated with the lever rule [40]. Any extensive property of the system can be determined at a given system state within the phase region with the same compositional and non-compositional intensive properties (chemical potential, temperature and pressure) as the stable phases at equilibrium. This is done by taking the sum of the stable phases' extensive properties (properties at coordinates on the phase region boundaries) weighted by their calculated phase fractions towards the system state (properties at a coordinate within the phase region). For some phase regions, only the meshing of phase region boundaries would therefore suffice, and the lever rule can be used to interpolate within the phase region. This was the basis for a phase region discretisation accelerator proposed by Zietsman [4]. It was shown that the Gibbs phase rule could be used to determine whether a phase region had to be meshed or only its boundaries. This would reduce the number of equilibrium calculations that had to be performed and the storage requirement of the database immensely – an entire phase region can be described by its boundaries alone. For the liquid-Pb (2-phase) phase region in the Pb-Sn (2-component) system, seen in Figure 5, it would only be necessary to mesh the phase region boundaries ac-

cording to the Gibbs phase rule. The lever rule could then been used to calculate thermodynamic properties within the phase region between the linearly interpolated phase region boundary nodes – no meshing would be required inside the phase region. The meshing of some phase regions are unavoidable, such as single-phase regions [4]. This method can be used to include equilibrium calculations into models more efficiently and can even be implemented together with the Scheil-Gulliver solidification method [41] to accelerate solidification models even more.

4.8. Tie-simplices

Phase diagrams are geometrical representations of thermochemical systems. There are many geometrical objects within a phase diagram, of which tie-lines, tie-triangles, etc. are used to represent behaviour within phase regions in isobaric and isothermal sections. A simplex is the generalization of a tetrahedral region of space in n-dimensions, such as a tetrahedron in three-dimensions, a triangle in two-dimensions or a line in one-dimension. Simplices can therefore be used to describe multi-dimensional tie-features in multi-phase regions. The vertices of a tie-simplex are the compositions of the phases present at equilibrium – nodes on the phase region boundaries. A phase region with $\hat{\varphi}$ number of phases can be discretised by a series of tie-simplices of order $\mathbb{R}^{\hat{\varphi}-1}$.

Voskov and Tchelepi [42] showed a method where, for a given phase region, the largest tie-simplex was first identified; for a two-phase region this would be the longest tie-line, for a three-phase region this would be the largest tie-triangle, etc. From the initial tie-simplex, an increment was made orthogonal to it and the next tie-simplex calculated. This was repeated until the entire phase region was discretised by a set of tie-simplices, as seen in Figure 6. Tie-simplices were created for all the phase regions in the system. The phase compositions that are the tie-simplex's vertices, and the thermochemical properties at each, were calculated by an equilibrium calculation for a given temperature and pressure within the phase region, and then stored. Interpolation could be done between vertices of neighbouring tie-simplices, and together with the lever rule, could recall thermochemical properties to be used in models. The

use of tie-simplex tabulation had significant gains in computational efficiency when compared to direct integration methods [42].
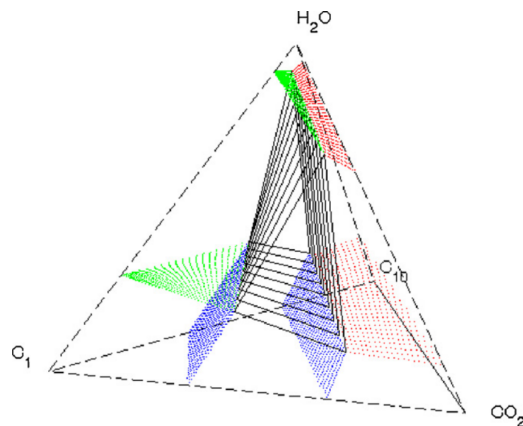

Figure 6: Tetrahedral diagram of 4-component system with simplices capturing phase regions. Taken from [42].

An adaptive strategy was developed where only the necessary tie-simplices were computed when a large number of these calculations were required [43]. If sufficient tie-simplex data was not available to provide the model with information regarding a query, equilibrium calculations would be performed and the database updated with corresponding tie-simplices.

4.9. Support Vector Data Description

A support vector data description (SVDD), similar to a support vector machine, is a non-probabilistic machine-learning classifier that has been used to estimate the location of phase region boundaries of a thermochemical system [44]. Initial temperature and composition coordinates, either obtained from prior knowledge or an optimisation routine, are used to determine phase composition coordinates on phase region boundaries and create a crude support vector data description of the phase region boundaries. Thereafter, an adaptive sampling scheme is employed to determine more phase composition coordinates on the phase region boundaries where it is currently least defined, and the support vector data description is grown until a maximum number of coordinates are

reached. In a case study performed by Kirk et al. [44], a support vector data description, together with the adaptive sampling scheme, required just 1,000 equilibrium calculations to represent phase region boundaries. A total of 505,000 direct equilibrium calculations were performed, as would be done in a traditional method, to verify the accuracy of the support vector data description and only a 5 % misclassification rate was observed – this is considered relatively accurate for its application [44] considering the reduction in the number of equilibrium calculations that need to be performed.

4.10. Sensitivity Derivatives

An on-demand machine learning algorithm was developed that could quickly and accurately estimate thermochemical equilibrium states based on stored results of previous equilibrium calculations [45]. When an equilibrium calculation was performed, sensitivity derivatives with respect to temperature, pressure and system component concentrations where determined. These derivatives could be used, together with changes in input conditions such as temperature differences or change in the concentration of a system component, to estimate the output state.

Stored sensitivity derivatives can only be used to estimate an equilibrium state if the requested input state is close to the reference state – the state at which the stored sensitivity derivatives were calculated. The most performance-critical step of this method is searching for an acceptable stored reference state [45]. An acceptability test is first done on a reference state and only when the error was found to be within a specified tolerance could the reference state and its sensitivity derivatives be used in a first-order Taylor approximation to estimate the equilibrium state.

If no acceptable reference states were found an equilibrium calculation would be performed and the accompanying sensitivity derivatives stored. No prior knowledge of the system or large pre-calculated databases were therefore required.

This method was tested and an acceleration factor of one to two orders of magnitude was achieved. Of the 1,000,000 equilibrium state calculation requests

from the model, only 258 equilibrium calculations had to be performed. The remaining equilibrium states were estimated by the on-demand machine learning algorithm. From these 258 equilibrium calculations, 91 % had been completed within the first 250 of the total 10,000 time steps.

4.11. Artificial Neural Networks

Artificial neural networks (ANNs) were trained on thermochemical properties obtained from equilibrium calculations done for sets of compositions and temperatures. ANNs have been used together with empirical models [46] and integrated with probability density functions [47]. The difficulty with selecting a composition range to be used as a training set is that the range is unknown prior to the simulation. A large training composition range can lead to unnecessary time-consuming equilibrium calculations to create the training set. If the training set composition range is too small then the ANN will not adequately represent the system.

To best capture the used compositional ranges of the model, statistical mapping was done; the system components were selected and small-scale models were done by direct integration of equilibrium calculations [47]. The small-scale models were used to generate a training set for the ANN that was representative of the larger system and its compositional ranges. The neural network was then trained on the automatically generated training set. The statistically mapped training sets were about 10 % of a tabulated database for the same system [47].

ANNs were used to accelerate equilibrium calculations by Guérillot and Bruyelle [48] and an acceleration factor of 10 to 20 was achieved for most of the time steps. However, it was stated that ANNs do not conserve mass and the error increases with every time step and is highly dependent on how well the ANN is trained. To reduce the error a larger training set would be required. A suggestion was made to use the ANN's result as the initial guess for the equilibrium calculation solver, potentially reducing the number of iterations required before convergence is achieved, reducing the computational cost.

ANNs were used to accelerate equilibrium calculations by Strandlund [49] as well where an acceleration

12

factor of 70 was observed with a 5 % maximum relative error allowed. Strandlund [49] mentions that the number of hidden units should be reduced as much as possible in order to train ANNs as fast as possible, but the more hidden units are used, the more complicated functions the ANN can represent – thermochemical properties with more complex behaviour can be represented more accurately. Knowledge of the system is therefore important; knowledge of the behaviour of the thermochemical properties as temperature, pressure, and composition of the system varies can be used to intelligently select the number of hidden units required – reducing the amount of training data required and the time spent on training the ANN to accurately estimate thermochemical properties.

4.12. k-Nearest Neighbours

A k-NN (k-nearest neighbour) algorithm [50] was used to construct a thermochemical property model. Many equilibrium calculations were performed to create a training set of thermochemical properties as a function of the specified states. After training was complete the k-NN algorithm could estimate thermochemical properties based on a specified system state by determining the nearest neighbours to the specified state and performing a weighted average between them. The error made by the k-NN model was within 1 % from the direct calculation results and showed an acceleration from 15.28 days to 4 minutes – a factor of about 5500. Other machine-learning methods, as referred to in [50], were considered when this method was developed.

5. Conclusion

Insights were obtained into systems and processes that, without the inclusion of thermochemical equilibrium calculations into models, would otherwise be difficult or impossible.

The computationally expensive Gibbs free energy minimization routine to obtain the equilibrium state of these complex systems is the main culprit to why these calculations are not incorporated into more models. As more system components are included in the equilibrium calculations to model complex systems, the more computationally expensive the Gibbs free energy minimization routine – the relationship between the number of components and calculation time is unfortunately a high-order non-linear relationship. Several methods have been developed in an attempt to accelerate equilibrium calculations and make it feasible to include into models.

Pre-calculated databases have the drawback of possibly becoming too large for computers to use. A pre-calculated uniform grid database of two thermochemical properties in a 4-component system was estimated to be in excess of 4 TB in size. By clustering nodes to important concentrations that correspond to discontinuities in thermodynamic properties this database could be reduced to 5 GB, which is significantly smaller but could still present recalling problems. For this reason, it would be better to have a database that is populated in-situ as the model is being solved – the model creates its own database of the accessed region for later recall. Initially this method is slower than direct integration, but as the database is populated the frequency of recalls and linear interpolation increase and the time spent on equilibrium calculations decrease, especially if many iterations and time steps are taken by the model. The database of an in-situ method only contains data of the system that the model has accessed before and no unnecessary data is generated and stored.

Using a system's phase diagram as a guide to discretise the temperature-compositional space allowed non-uniform discretisation of phase regions. Thermochemical data could be stored at the nodes of the non-uniform mesh for later recall. The non-uniform discretisation allowed for less nodes to be used, compared to uniform discretisation, as the nodes could be concentrated at the phase region boundaries.

The discretisation of phase regions are however not always necessary. By utilising thermochemical tools such as the Gibbs phase rule it could be determined whether an entire phase region had to be discretised or if it was only necessary to discretise its boundaries and store the resulting tie-simplices. The lever rule could be used with the stored tie-simplices to determine the thermochemical properties within the entire phase region without there being a single data point

13

stored inside the phase region. This lead to a remarkable decrease in the amount of data that had to be stored. Because the lever rule is only another form of linear interpolation the acceleration of this method shows great promise.

There are great advantages in using concepts from thermochemistry theory such as phase diagrams, the Gibbs phase rule, and the lever rule in creating an accelerator. The thermochemical theory is a strong base that provides security for the decisions taken when the system is discretised and interpolation is done with the stored data. Combining this with an in-situ method of discretisation would produce a sparse database that covers large temperature, pressure and compositional ranges. An accelerator that utilises these thermochemical tools to build an in-situ database consisting of tie-simplices created by equilibrium calculations could show great promise and should be investigated.

The acceleration of equilibrium calculations is only one hurdle that has to be overcome to include thermochemistry into models. Additional PDEs are required in the models to ensure the mass balance of system components and phases throughout the model. Other hurdles may present themselves as we investigate larger systems, but there are very good methods developed to help address the current hurdle we are facing – the acceleration of equilibrium calculations for more feasible model solution times.

Acknowledgements

References

[1] Ex Mente Technologies, Chemapppy, 2019. URL: https://www.ex-mente.co.za/chemapppy.

[2] A. Ishmurzin, M. Gruber-Pretzler, F. Mayer, M. Wu, A. Ludwig, Multiphase/multicomponent modelling of solidification processes: Coupling solidification kinetics with thermodynamics, International Journal of Materials Research 99 (2008) 618–625.

[3] R. Pillai, T. Galiullin, A. Chyrkin, W. Quadakkers, Methods to increase computational efficiency of calphad-based thermodynamic and kinetic models employed in describing high temperature material degradation, CALPHAD: Computer Coupling of Phase Diagrams and Thermochemistry 53 (2016).

[4] J. Zietsman, Efficient storage and recall of slag thermochemical properties for use in multiphysics model, in: R. Reddy, P. Chaubal, P. Pistorius, U. Pal (Eds.), The 10th International Conference on Molten Slags, Fluxes and Salts, The Minerals, Metals and Materials Society, Seattle, USA, 2016, pp. 635–644. doi:doi:10.1002/9781119333197.ch68.

[5] Y. Teh, G. Rangaiah, A study of equation-solving and gibbs free energy minimization methods for phase equilibrium calculations, Chemical Enginnering Research and Design 80 (2002) 745–759.

[6] M. Piro, S. Simunovic, T. Besmann, B. Lewis, W. Thompson, The thermochemistry library thermochimica, Computational Materials Science 67 (2013) 266 – 272.

[7] A. Voskov, A. Dzuban, A. Maksimov, Ternapi program for the calculation of ternary phase diagrams with isolated miscibility gaps by the convex hull method, Fluid Phase Equilibria 388 (2015) 50–58.

[8] B. Sundman, U. Kattner, M. Palumbo, S. Fries, Opencalphad - a free thermodynamic software, Ingegrating Materials and Manufacturing Innovation 4 (2015) 1 – 15.

[9] B. Sundman, X.-G. Lu, H. Ohtani, The implementation of an algorithm to calculate thermodynamic equilibria for multi-component systems with non-ideal phases in a free software, Computational Material Science 101 (2015) 127–137.

[10] R. Otis, Z.-K. Liu., pycalphad: Calphad-based computational thermodynamics in python, Journal of Open Research Software 5 (2017) 1 – 11.

[11] D. V. Nichita, S. Gomez, E. Luna, Multiphase equilibria calculation by direct minimization of gibbs free energy with a global optimization method, Computers & Chemical Engineering 26 (2002) 1703 – 1724.

[12] C. Bale, E. Bélisle, P. Chartrand, S. Decterov, G. Eriksson, A. Gheribi, K. Hack, I. Jung, Y. Kang, J. Melançon, A. Pelton, S. Petersen, C. Robelin, J. Sangster, P. Spencer, M. van Ende, Factsage thermochemical software and databases - 2010 – 2016, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 54 (2016) 35–53.

[13] K. Shobu, Catcalc: New thermodynamic equilibrium calculation software, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 33 (2009) 279–287.

[14] R. Davies, A. Dinsdale, J. Gisby, J. Robinson, S. Martin, Mtdata - thermodynamic and phase equilibrium software from the national physical laboratory, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 26 (2002) 229–241.

[15] J. Andersson, H. T., L. Höglund, P. Shi, B. Sundman, Thermo-calc and dictra, computational tools for materials science, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 26 (2002) 273–312.

[16] S.-L. Chen, S. Daniel, F. Zhang, Y. Chang, X.-Y. Yan, R. Xie, F.-Y. Schmid-Fetzer, W. Oates, The pandat software package and its applications, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 26 (2002) 175–188.

[17] A. ten Cate, B. Geurts, M. Muskulus, D. Kooster, A. Muntean, J. van Opheusden, N. Köster, A. Muntean, A. Peschansky, B. Vreman, P. Zegeling, Modeling and simulation of phase-transitions in multicomponent aluminum alloy casting, in: Proceedings of the sixty-third European Study Group Mathematics with Industry 2008, Enschede, The Netherlances, 2008, pp. 117–139.

[18] C. Agelet de Saracibar, M. Chiumenti, M. Cervera, Current developments on the coupled thermomechanical computational modeling of metal casting processes, Enhancement and Promotion of Computational Methods in Engineering and Science (2006) 247–260.

[19] E. Blond, T. Merouki, N. Schmitt, E. de Bilbao, A. Gasser, Multiphysics modelling applied to refractory behaviour in severe environments, Advances in Science and Technology 92 (2014) 301–309.

[20] A. Tabiei, S. Sockalingam, Multiphysics coupled fluid/thermal/structural simulation for hypersonic reentry vehicles, Journal of Aerospace Engineering 25 (2012) 273–281.

[21] S. Pannala, S. Simunovic, G. Frantziskonis, Computational Modeling in Lignocellulosic Biofuel Production, 2010. doi:doi:10.1021/bk-2010-1052.ch011.

[22] E. Corcoran, M. Kaye, M. Piro, An overview of thermochemical modelling of candu fuel and applications to the nuclear industry, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 55 (2016) 52–62.

[23] K. Samuelsson, J.-C. Dumas, B. Sundman, J. Lamontagne, C. Guéneau, Simulation of the chemical state of high burnup (u, pu)o2 fuel in fast reactors based on thermodynamic calculations, Journal of Nuclear Materials 532 (2020) 151969–1–151969–12.

[24] J. Zietsman, Interactions between freeze lining and slag bath in ilmenite smelting, Ph.D. thesis, University of Pretoria, Pretoria, South Africa, 2004.

[25] J. Zietsman, P. Pistorius, Modelling of an ilmenite-smelting dc arc furnace process, Minerals Engineering 19 (2006) 262–279.

[26] J. Zietsman, P. Pistorius, Ilmenite smelter freeze lining monitoring by thermocouple measurements: Model results, in: Heavy Minerals 2005, Society for Mining, Metallurgy, and Exploration, 2005, pp. 221–228.

[27] T. Hay, T. Echterhof, V.-V. Visuri, Development of an electric arc furnace simulator based on a comprehensive dynamic process model, Processes 7 (2019) 852.

[28] T. Hay, A. Reimann, T. Echterhof, Improving the modeling of slag and steel bath chemistry in an electric arc furnace process model, Metallurgical and Materials Transactions B 50 (2019) 2377–2388.

[29] A. Saad, C.-A. Gandin, M. Bellet, Temperature-based energy solver coupled with tabulated thermodynamic properties – application to the prediction of macrosegregation in multicomponent alloys, Computational Materials Science 99 (2015) 221–231.

[30] X. Doré, H. Combeau, M. Rappaz, Modelling of microsegregation in ternary alloys: Application to the solidification of al-mg-si, Acta Materialia 48 (2000) 3951–3962.

[31] M. Wu, J. Li, A. Ludwig, A. Kharicha, Modeling diffusion-governed solidification of ternary alloys – part 1 : Coupling solidification kinetics with thermodynamics, Computational Materials Science 79 (2013) 830–840.

[32] S. Pope, Computationally efficient implementation of combustion chemistry using in-situ adaptive tabulation, Combustion Theory and Modelling 1 (1997) 41–63.

[33] J.-Y. Chen, Analysis of in situ adaptive tabulation performance for combustion chemistry and improvement with a modified search algorithm, Combustion Science and Technology 176 (2004) 1153–1169.

[34] L. Lu, S. Pope, An improved algorithm for in-situ adaptive tabulation, Journal of Computational Physics 228 (2009) 361 – 386.

[35] H. Larsson, L. Höglund, A scheme for more efficient usage of calphad data in simulations, CALPHAD: Computer Coupling of Phase Diagrams and Thermochemistry 50 (2015) 1–5.

[36] R. Gandham, K. Esler, K. Mukundakrishnan, Y. Zhang, C. Fang, V. Natoli, Gpu acceleration of equation of state calculations in compositional reservoir simulation, in: 15th European Conference on the Mathematics of Oil Recovery, Amsterdam, Netherlands, 2016.

[37] J. Gibbs, A method of geometrical representation of the thermodynamic properties of substances by means of surfaces, Transactions of the Connecticut Academy of Arts and Sciences (1873) 382–404.

[38] K. Qiu, R. Wang, C. Peng, X. Lu, N. Wang, Polynomial regression and interpolation of thermodynamic data in al-si-mg-fe system, Calphad: Computer Coupling of Phase Diagrams and Thermochemistry 48 (2015) 175–183.

[39] G. Zhao, X. Li, D. Xu, J. Guo, H. Fu, Y. Du, Y. He, Numerical computations for temperature, fraction of solid phase and composition couplings in ternary alloy solidification with three different dhermodynamic data-acquisition methods, CALPHAD: Computer Coupling of Phase Diagrams and Thermochemistry 36 (2012) 155–162.

[40] W. Smith, J. Hashemi, Foundations of Materials Science and Engineering, 4th ed., McGraw-Hill, 2006.

[41] B. Sundman, I. Ansaram, III.2 - The Gulliver–Scheil method for the calculation of solidification paths, Woodhead Publishing Series in Metals and Surface Engineering, second edition ed., Woodhead Publishing, 2008, pp. 343–346. doi:doi:https://doi.org/10.1533/9781845693954.3.343.

[42] D. Voskov, H. Tchelepi, Tie-simplex based mathematical framework for thermodynamical equilibrium computation of mixtures with an arbitrary number of phases, Fluid Phase Equilibria 283 (2009) 1–11.

[43] D. Voskov, H. Tchelepi, Compositional space parametrization for miscible displacement simulation, Transport in Porous Media 75 (2008) 111–128.

[44] T. Kirk, E. Galvan, R. Malak, R. Arroyave, Computational design of gradient paths in additively manufactured functional graded materials, Journal of Mechanical Design 140 (2018) 1111410–1–1111410–9.

[45] A. Leal, S. Kyas, D. Kulik, M. Saar, Accelerating reactive transport modeling: On-demand machine learning algorithm for chemical equilibrium calculations, Transport in Porous Media 133 (2020) 161–204.

[46] F. Christo, A. Masri, E. Nebot, T. Turányi, Utilising artificial neural network and repromodelling in turbulent combustion, Proceedings of ICNN'95 - International Conference on Neural Networks (1995) 911–916.

[47] F. Christo, A. Masri, E. Nebot, S. Pope, An integrated pdf/neural network approach for simulating turbulent reacting systems, Symposium (International) on Combustion/The Combustion Institute 26 (1996) 43–48.

[48] D. Guérillot, J. Bruyelle, Geochemical equilibrium determination using an artificial neural network in compositional reservoir flow simulation, Computational Geosciences 24 (2020) 697–707.

[49] H. Strandlund, High-speed thermodynamic calculations for kinetic simulations, Computational Materials Science 29 (2004) 187–194.

[50] R. Jha, N. Chakraborti, D. Diercks, A. Stebner, C. Ciobanu, Combined machine learning and calphad approach for discovering process-structure relationships in soft magnetic alloys, Computational Materials Science 150 (2018) 202–211.