ThermoCodegen: a python/C++ package for the generation of custom thermodynamic models

Marc Spiegelman ¹,², Owen Evans ¹,³, Mark Ghiorso ⁴, Lucy Tweed ², and Cian Wilson ⁵

¹ Dept. of Applied Physics and Applied Math, Columbia University ² Dept. of Earth and Environmental Science, Columbia University ³ Now at Google ⁴ OFM Research ⁵ Earth and Planets Laboratory, Carnegie Institution for Science ¶ Corresponding author

DOI: 10.21105/joss.04874

Summary

Many important problems in solid-Earth science require a consistent coupling between thermodynamics and geodynamics. Examples include global mantle convection with thermodynamically consistent phase changes, the generation and flow of magma in the mantle, reactive fluid flow and geological carbon sequestration in the crust and even the coupled flow of water and ice in glaciers and ice sheets. Given the complexity of these systems, it is important that the individual user/scientist have considerable control over the formulation and computational choices involved for both the thermodynamic and geodynamic components of models while maintaining transparency and reproducibility. Access to flexible open-source software for the creation and exploration of thermodynamic models and their incorporation into large scale reactive flow simulations is essential for these endeavors.

Statement of need

ThermoCodegen is a software package for the reproducible generation of custom thermodynamic and kinetic reaction models for minerals, melts and some aqueous fluids commonly encountered in the solid-Earth sciences. It is part of the larger ENKI project for thermodynamic modeling whose flagship code ThermoEngine (Ghiorso & Wolf, 2019; Wolf & Ghiorso, 2019) provides a JupyterHub, cloud-based environment for the interactive generation and exploration of thermodynamic models in Earth sciences. In particular, ThermoEngine provides a critical code-generation module, coder.py for generating high-performance C code from SymPy expressions for the free energies of thermodynamic endmembers and phases.

ThermoCodegen shares the same underlying code-generation infrastructure as ThermoEngine however its design is driven more by the need for ready incorporation of thermodynamics into large scale geodynamics simulation codes for both equilibrium and disequilibrium transport.

Key additional features of ThermoCodegen include:

- **Portable model storage**: Thermodynamic model generation for endmembers and phases are described in detailed Jupyter notebooks using coder.py and SymPy, however, the final models are stored in structured xml files that contain all the information required to describe and auto-generate code for thermodynamic databases of endmembers and phases.
- **Kinetic modeling**: ThermoCodegen also provides code generation and storage of custom kinetic reaction models built on the phase databases.

*co-first author
• **C++ support**: The code also generates C++ libraries for inclusion in other modeling systems such as TerraFERMA (Wilson et al., 2017; Wilson & Spiegelman, 2015) or ASPECT (Bangerth et al., 2021; Heister et al., 2017; Kronbichler et al., 2012).

• **Python bindings**: Python bindings of the C++ classes are generated using pybind11 (Wenkel et al., 2017) for direct import into python applications.

• **ThermoEngine support**: Python bindings of ThermoCodegen objects can also be imported back into ThermoEngine as a custom database for use in equilibration or calibration calculations (requires installing the tcg_model_integration branch of ThermoEngine).

**Relationship to other Software packages for Solid-Earth Thermodynamic modeling**

Computational thermodynamic software has been a central tool for solid-Earth science for over 40 years and has focused primarily on calculating phase equilibrium for several distinct self-consistent “Thermodynamic Datasets” such as those for solid mineral phases (Berman, 1988; Holland & Powell, 1998, 2011; Stixrude & Lithgow-Bertelloni, 2005, 2011), silicate liquids (Ghiorso et al., 2002; Ghiors & Sack, 1995; Holland et al., 2018), and ionic species in aqueous solutions (Shock et al., 1992; Tanger & Helgeson, 1988).

Many of these datasets are associated with specific software for calculating equilibrium assemblages using a range of Gibbs free energy minimization algorithms. Some of the most commonly used software includes the MELTs family of models (Asimow & Ghiorso, 1998; Ghiorso et al., 2002; Ghiors & Gualda, 2015; Ghiors & Sack, 1995; Gualda et al., 2012) for equilibration of silicate minerals and melts, which has also been repackaged as alphaMELTS (Antoshechkina & Ghiors, 2018; Smith & Asimow, 2005) as well as in its current incarnation within the ENKI project. Primarily used for metamorphic petrology, THERMOCALC\(^1\) (Powell et al., 1998) is closely associated with the thermodynamic models of Holland and Powell (Holland & Powell, 1998, 2011), while HeFESTo is used for the calculation of phase equilibria of the high pressure/temperature solid-state models of Stixrude and Lithgow-Bertelloni (Stixrude & Lithgow-Bertelloni, 2005, 2011). Note that not all of these packages have open-source licenses. In addition, there are several more general Gibbs free energy minimization packages that can be used with multiple data sets. Perple\_X (Connolly, 1990, 2005, 2009) is one of the most widely used. More recent additions include THERIAK-DOMINO (Capitani & Petrakakis, 2010) which also supports the models of Holland and Powell, Thermolab (Vrijmoed & Podladchikov, 2022) a MATLAB based toolkit, and MAGEMin (Riel et al., 2022) an open-source Gibbs energy minimization package written in C (with julia and MATLAB bindings) for adaptive phase-diagram generation with a range of modern optimization techniques. In particular, MAGEMin includes the igneous thermodynamic database of Holland et al. (2018).

These packages provide critical thermodynamic modeling infrastructure for solid-Earth science but are complementary and distinct in design and applicability to ThermoCodegen. In particular, the primary features added by ThermoCodegen are:

- the ability to more easily create and explore custom, consistent models of all thermodynamic parameters through the use of symbolic differentiation and automatic code generation
- the ability to generate code for disequilibrium reaction kinetics
- the ability to create custom C++ and python libraries for direct inclusion in geodynamics models.

Given the complexity of these high-dimensional open-system problems, having effective open-source software for reproducible exploration of both equilibrium phase diagrams and disequilibrium transport remains a central challenge and goal in geodynamics and ThermoCodegen is designed to address several of these critical challenges.

---

1 Not to be confused with the commercial package Thermo-Calc
Software design/overview

The design of ThermoCodegen follows the hierarchical structure of thermodynamic modeling commonly used in Earth Sciences and illustrated in Figure 1.

**Figure 1:** The hierarchical structure of thermodynamic modeling. \( \mu^k \) is an endmember’s chemical potential model and \( M^k \) its mass. \( G_i \) is a phase’s Gibbs free energy model and \( \rho_i \), \( C_{Pi} \), \( s_i \), and \( \alpha_i \) are (for example) its density, heat capacity, specific entropy, and thermal expansivity, derived consistently by symbolic differentiation of \( G_i \). \( \nu_j \) describes the stoichiometry of reactions between the endmembers of the phases, \( A_j \) the resulting reaction affinities, \( R_i \) their rates, and \( \Gamma^k_i \) and \( \Gamma_i \) the resulting endmember and phase source terms. The Thermodynamic Database and Reactive System are functions of the temperature, \( T \), pressure \( P \), the number of moles, \( n \), or mass fractions, \( C \), of the endmembers, and presence (e.g. volume fraction) of the phases, \( \phi \). These can be supplied by coupling to a dynamic model that may additionally calculate phase velocities, e.g., \( v_s \) and \( v_f \). Image adapted from Tweed (2021).

At the lowest level are thermodynamic **Endmembers** which provide chemical potentials, \( \mu^k(T, P) \), for stoichiometric chemical components. The actual components can be arbitrary (e.g., oxides, minerals, ionic species) but have fixed chemical composition and thermodynamic properties that only depend on temperature and pressure (or temperature and volume, for models of the Helmholtz free energy).

Given a space of thermodynamic endmembers, we can construct homogeneous **Phases** as combinations of endmembers and develop thermodynamic models for their Gibbs free energy, \( G_i(T, P, n_i) \), that depend on temperature, pressure, and the number of moles of each endmember \( n_i = n^k_i \).

Given \( G_i(T, P, n_i) \) for each phase, all of the other thermodynamic properties (and their derivatives) can be derived by partial differentiation of \( G_i \). For example, entropy \( S_i \), volume \( V_i \), chemical potentials \( \bar{\mu}_i \), heat capacity \( C_{Pi} \), and thermal expansivity \( \alpha_i \) are all related to \( G_i \) by:

\[
S_i = -\frac{\partial G_i}{\partial T} \quad V_i = \frac{\partial G_i}{\partial P} \quad \bar{\mu}_i = \frac{\partial G_i}{\partial n^k_i} \quad C_{Pi} = T \frac{\partial S_i}{\partial T} = -T \frac{\partial^2 G_i}{\partial T^2} \quad \alpha_i = \frac{1}{V_i} \frac{\partial V_i}{\partial T} = \frac{1}{V_i} \frac{\partial^2 G_i}{\partial P^2}
\]

Thus if we can define an analytic model for \( G_i(T, P, n_i) \) for any phase, we can use symbolic differentiation and code-generation to provide custom routines to generate all thermodynamic properties consistently.

We define a set of phases, together with their constituent thermodynamic endmembers as a **Thermodynamic Database** corresponding to the blue box in Figure 1. Standard databases available in the literature include the mineral database of Berman (1988) and its extension to silicate liquids through the MELTs family of models (Ghiorso et al., 2002; Ghiorso & Sack, 1995), as well as the high-pressure mineral phases of Stixrude and Lithgow-Bertelloni (Stixrude & Lithgow-Bertelloni, 2005, 2011) or metamorphic phases of, for example, Holland & Powell (2011). All of these are currently available through python bindings to ObjectiveC code in

---

ThermoEngine. However, the ObjectiveC objects are not readily incorporated within other codes.

ThermoEngine also provides support for generating custom thermodynamic models of endmembers and phases through the thermoengine.coder module written by Mark Ghiorso. coder uses SymPy to describe the free energies of endmembers and phases and then uses automatic differentiation and code-generation to write efficient C source code (and wrap it in Cython) for all thermodynamic properties of phases. Our extensions in thermocodegen.coder generate the same underlying C code and interfaces as thermoengine.coder but extend its functionality in several critical ways for better integration with large-scale geodynamic modeling.

**ThermoCodegen**

ThermoCodegen is designed to produce code for just the colored boxes in Figure 1.

**Custom Thermodynamic Databases**

For **Thermodynamic Databases** of endmembers and phases, ThermoCodegen extends thermoengine.coder to record the internal state of each thermodynamic model as a python dictionary in order to store and generate code from xml files that completely describe all variables, parameters and SymPy models for endmembers and phases. These xml files are parsed using the science neutral options package Spud (Ham et al., 2009). Endmembers are stored with the extension .emml and phases are stored as .phml files, which can be viewed with the Spud GUI diamond. Given a set of consistent endmember and phase description files, the ThermoCodegen script tcg_builddb can be used to inspect, validate and autogenerate code for a custom Thermodynamic Database as well as provide C++ wrappers and python bindings using pybind11.

At this point, all source code, together with .emml and .phml files describing the full thermodynamic models, and optionally compiled C++ and python libraries are made available as a compressed tar-ball. This object is considered a Thermodynamic Database and can be used locally or made available as a web-accessible URL, for example with a zenodo DOI.

Given a custom thermodynamic database, these can then be used in optimization codes such as thermoengine.equilibrate to find equilibrium assemblages that minimize the global Gibbs free energy.

**Reaction Kinetic Objects**

For better integration with open-system geodynamics codes that do not assume equilibrium, ThermoCodegen also provides code for generating C++ libraries and python bindings for custom kinetics models that describe a set of reactions between endmembers and phases within a given thermodynamic database. The software provides interfaces for calculating affinities, reaction rates, and thermodynamic properties of phases as well as their exact derivatives for use in non-linear solvers that take Jacobians. These reaction kinetic models are stored as xml files with the extension .rxml from which custom C++ libraries and python bindings are generated using the build script tcg_buildrx.

These reaction libraries, along with the underlying database objects can then be incorporated in external modeling software such as ThermoEngine, TerraFERMA, ASPECT, Jupyter notebooks and general python or C++ codes. Inclusion in dynamic models is considered external to ThermoCodegen, however, the ThermoCodegen distribution also provides containers that include a complete installation of TerraFERMA along with some example test cases.

Detailed [documentation](https://doi.org/10.21105/joss.04874) for ThermoCodegen is available that includes a more detailed description of the software design as well as instructions for using the package through the provided docker and singularity containers (local installation guidelines are also available). Most importantly, the documentation also includes fully worked out examples illustrating the
work flow for using ThermoCodegen including a set of Jupyter notebooks for constructing and exploring a range of thermodynamic models.

Acknowledgements

This work was supported by NSF SI² award NSF-ACI1550337, NSF-EAR1520732, NSF-EAR1049905 and the Sloan Foundation project on Net-Zero and Negative Emissions Technologies.

References


Tanger, J. C., & Helgeson, H. C. (1988). Calculation of the thermodynamic and transport properties of aqueous species at high pressures and temperatures; revised equations of...


