Computational thermodynamics using the DWPM mixture model

with applications to phase equilibria and molecular dynamics

Stefan Endres

March 11, 2019

Institute of Applied Materials
Department of Chemical Engineering
University of Pretoria

Link to project repository: https://github.com/Stefan-Endres/DWPM-Mixture-Model ▸ Link

Introduction and perspective

Perspective i

In Chemical Engineering, thermodynamic computations are often an important and limiting step in simulation, design and control of processes. Equilibrium calculations involve computing thermodynamic properties of chemical mixtures in finite volume elements such as:

- Physical properties of mixtures
- Phase equilibria
- Reaction equilibria

Review of IAM developed software libraries

Overview i

The software library abstracts the thermodynamic calculations for any system by assuming that a thermodynamic model is capable of producing an expression for the Gibbs free energy and/or the potential energy of the system. The central equation for modelling the properties of mixtures is the DWPM mixture rule which attempts to capture the properties of a diverse set of models. The software provides universal methods for solving the following problems:

- Phase equilibria computation
    - Phase equilibrium at a P, T, $\mathbf{x}$ point
    - Phase envelopes
- Parameter optimisation of thermodyanmic models over a data set
- There are several methods for phase equilibria calculations of thermodynamic models

Overview ii

- The only formulation that is guaranteed to exclude various issues such as spinodal phase splits found in and other false solutions locally optimal formulations is global minimisation of the Gibbs energy surface.
- Global minimisation and stability analysis both involve fitting tangent solution planes (called the TPD in literature)
- The Dual problem formulation by [Mitsos and Barton,] capture these constraints in a semi-infinite programme.

Overview iii

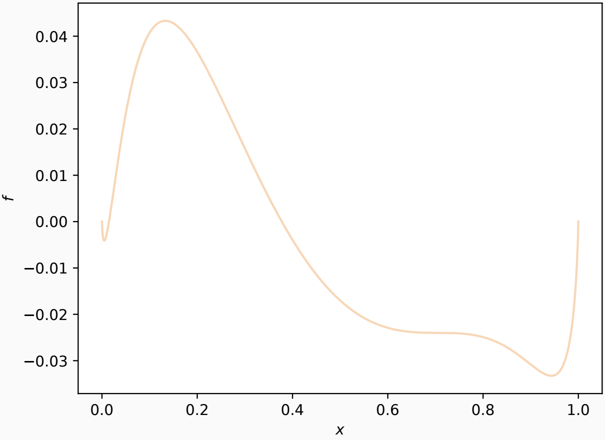


Figure 1: A 1-dimensional (binary) nrtl Gibbs energy surface of butyl-acetate

Overview iv

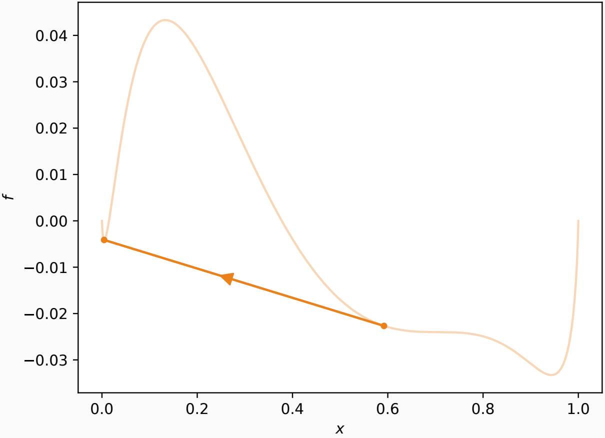


Figure 2: Tangent plane solution on corresponding to an equilibrium phase split

Overview v

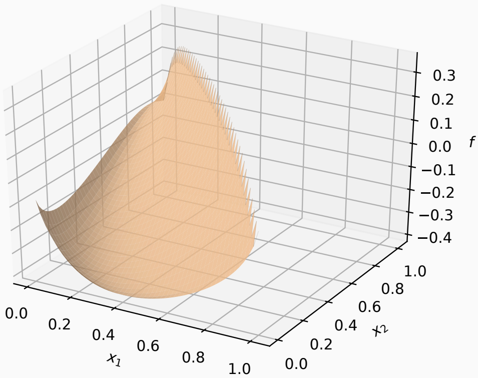


Figure 3: A 2-dimensional (ternary) Gibbs energy surface

Overview vi

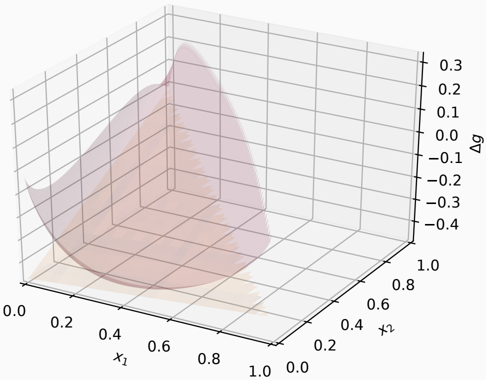


Figure 4: Tangent plane solution on corresponding to an equilibrium phase split

Current features i

- Can be applied to any n-component system
- The system can have an arbitrary number of phases:
  - For parameter optimisation the phases known a-priori is automatically determined by the data set
  - New liquid phases predicted by the model (outside the data range) are detected automatically and added to the phases
  - By default the VdW EOS with DWPM mixing rule can have one vapour phase and an arbitrary number of liquid phases
  - Solid phases are possible, but it is recommended to use another EOS
- Parameter optimisation of an equation of state model to a dataset
  - Currently the VdW EOS with DWPM mixing rules and the NRTL EOS are built in, but it is very easy to add more models
  - Any model supplying a Gibbs free energy works, therefore all EOS, CPA, G-EoS, PC-SAFT, DFT etc. will all work with this software
- Optimisation is done with a novel algorithm and is very reliable

Current features ii

- The VL(L)E phase envelopes can be computed from any EOS with a given parameter set
- The VL(L)E phase envelopes for can graphed for either a fixed pressure or temperature in ternary systems or varying P,T for binary. The real world data can be added based on the options supplied to the software
- These graph outputs are publication quality
- In addition you can also plot the Gibbs free energy surface of the system for binary and ternary systems easily (which is a nice way to understand the behaviour of a system)

Comparison to other software (parameter optim.) i

- Many commercial software libraries and literature articles involve simpler equation based approaches which solves the parameter optimisation problem to a local minima. This approach:
  - Can scale to many components in a system
  - Has many issues finding a feasible representation of the underlying Gibbs free energy, often defects are present in the energy surface or spinodal solutions are used in parameter fitting
  - This limits the applications and often causes problems in simulations or with finding the true parameters for a data set

Comparison to other software (parameter optim.) ii

- A rigorous statement of the problem was first formulated as a weighted least squares error between measured and predicted mole fractions by Bolas et. al. [Mitsos et al., 2009, Bollas et al., 2009]
- The objective function for n_e number of experiments each with a n_p^i number of observed phases is given by

$$\min_{\mathbf{p},\mathbf{x}} f(\mathbf{p},\mathbf{x}) = \sum_{i=1}^{n_e} \sum_{k=I}^{n_p^i} w^{i,k} \left(\mathbf{x}^{m,i,k} - \mathbf{x}^{p,i,k}(\mathbf{p})\right)^2 \qquad (1)$$

- where $\mathbf{x}^{m,i,k}$ are the measured phase compositions and $\mathbf{x}^{p,i,k}$ are the phase compositions predicted by the model in mole fractions
- Additional constraints are imposed on the lower level problem

Comparison to other software (parameter optim.) iii

- This can be solved as bilevel problem wherein we optimise the model parameters $\mathbf{p}$ in the upper level problem and solve the equilibrium problem $\mathbf{x}^{p,i,k}$ in the lower level problem
- Well behaved resultant model
  - Well behaved resultant model guaranteed by its constraints
  - Enforces global stability using the tangent plane distance function (TPD)
  - Enforces the correct number of phase splits
  - Enforces the correct number of phases in every split
  - Excludes spurious phases
- **Limited scaling!** (roughly ≤ 10 component systems)

Comparison to other software (parameter optim.) iv

- The reformulation in the current software allows for the bilevel problem to be restated as a single level problem that is significantly easier to solve

$$\min_{\mathbf{p}} f(\mathbf{p}) \tag{2}$$

- The reformulation has the following features:
  - The solution to the single level problem is proven to be the globally optimal solution of the bilevel problem
  - Convergence proven for any black box thermodynamic model
  - The model's derivatives are not estimated and convergence is guaranteed for broadest class of non-smooth, non-continuous models
  - Complete topological understanding of the Gibbs energy surface
  - Far more efficient scaling (100s of components)

Comparison to other software (phase equilibrium) i

- The bilevel parameter estimation and dual equilibrium calculation problems are both complex since we need to solve it by iterating between two problems
- In bilevel parameter estimation
  - The an upper level parameter optimisation problem minimising the objective function
  - The lower level problem solving either the an equilibrium calculation or equivalently satisfying the TPD
- In equilibrium calculation
  - The an upper level problem which solves the primal problem of globally optimising the Lagrangian of the Gibbs energy surface
  - The lower level linear programming problem solving the Lagrange multipliers (proven to be equal to the chemical potential by [Mitsos and Barton,] (2007))

Comparison to other software (phase equilibrium) ii

- Current phase equilibrium solver uses the dual formulation by Mitsos and Barton (2007)
- Volume root calculations using latest algorithms from literature
- The calculations are done by using an expression for the change in reduced Gibbs free energy of mixing of a model at a fixed pressure, temperature and feed composition $\Delta g_{mix}\left(P, T, \mathbf{x}\right)$

Comparison to other software (phase equilibrium) iii

- Math-free overview of a single equilibrium computation of the VdW-DWPM EoS with k phases[1] and i components:
  1. Supply a P, T and initial mixture composition $\mathbf{z}$
  2. Feed P, T to expressions of the residual Gibbs free energy expressions of the pure components $g_i^{R,k}(P,T)$ for every k, i
     - 2.1 Pure component parameters a_i and b_i are used
     - 2.2 There is an internal computation to find the pure volume roots of the EoS (an implicit equation), here we compute V_i^k for every k, i. For example every V_1^l, V_2^l, V_1^v, V_2^v (pure vapour and pure liquid volume roots for every component in a binary system)
  3. Feed P, T to expressions of the residual Gibbs free energy expressions of the mixture $g_{mix}^{R,k}(P,T,\mathbf{z})$
     - 3.1 Internally we compute the mixture parameters $a_{mix}(\mathbf{z})$ and $b_{mix}(\mathbf{z})$ at the mixture composition $\mathbf{z}$
     - 3.2 There is an internal computation to find the mixture volume roots of the EoS (an implicit equation), here we compute V_{mix}^k for every k, i. For example V_{mix}^l, V_{mix}^v (mixed vapour and mixture liquid volume roots

Comparison to other software (phase equilibrium) iv

4. We can now compute the change in the Gibbs free energy of mixing (using the partial molal property relationship) for every phase k

$$\Delta g_{mix}^{k}\left(P,T,\mathbf{z}\right) = g_{mix}^{R,k}\left(P,T,\mathbf{z}\right) - \sum_{i}^{n} z_i g_i^{R,k}\left(P,T\right) + \Delta g_{mix}^{IG,k}\left(\mathbf{z}\right)$$

5. Finally we can now use the values of Δg_{mix}^{k} to find the true change in Gibbs free energy of mixing:

$$\Delta g_{mix} = \min\{\Delta g_{mix}^{k}\}\forall k$$

- Note that we store that phase of the smallest Δg_{mix}^{k} since this determines whether the final equilibrium solution is "liquid like" or "vapour like", usually there is only one vapour phase, but several possible liquid phases.

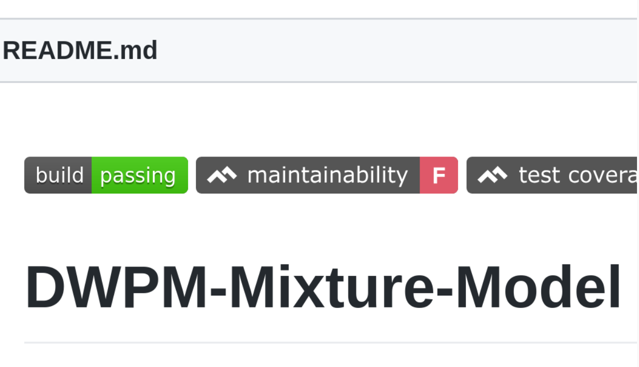
[1] Here "k" phases refers to the number of volume roots used, usually one liquid root l and one vapour root v is used, but more are possible. Note that more than 2 phases are possible since l can have several phases (VL(L)E equilibrium

Limitations of the current software i

- Poor data handling: data needs to be supplied as an excel spreadsheet in .csv format (rather than downloading data directly like ASPEN or factsage does)
- Documentation for developers is extensive, but documentation for users is limited

Future and maintainability of the software i

- Biggest issue:



README.md

build passing maintainability F test covera

DWPM-Mixture-Model

Figure 5: Codeclimate's opinion

Future and maintainability of the software ii

- The new parameter optimisation methods (including rigorous constraints by Mitsos et. al. 2009 for VL(L)E systems) is finished, but has not been merged with the public software yet. This method is extremely fast compared to the current method used and is also more reliable
- Much work is currently being done on making the software user friendly and easing the implementation of arbitrary energy models

Tutorial

Installation and setup i

1. Install the latest version of Python by following the instructions on the departmental website: http://chemeng.up.ac.za/wiki/index.php/Python ▸ Link
2. Go to the online repository and click on the green "Clone or download" button which provides further instructions for both Windows and Unix based systems: https://github.com/Stefan-Endres/DWPM-Mixture-Model ▸ Link
3. Modify the config.cfg file in the base directory to point towards your own data directory (usually the dropbox containing the data). For example my entry contains the data directory "datadir: /home/stefan_endres/projects/dwpm data/Data/"
4. Navigate towards cloned file directory using either Powershell (Windows) or a CLI (UNIX) to begin parameter optimisations or simulations.

Data handling and adding new chemical systems i

1. Use the example spreadsheets in the data directory as a template to add new data
2. For every system you want to simulate with the DWPM-VdW EoS you will need:
   2.1 A pure component data .csv file containing vapour phase data for every component in the system (stored in the "/Data/Pure_Component" directory)
   2.2 A data set containing the equilibrium phase data of the mixed system in a .csv file (stored in the "/Data/nComp_E" directory)
   2.3 The names of the pure components and the components in the mixed file names must match (**case sensitive!**)
3. Note that you do not need to optimise the pure component parameters individually, when you simulate a new system the software will automatically optimise and save if no pure component parameters are found

Data handling and adding new chemical systems ii

4. If you cannot find the data files in the shared Dropbox it is also available on my Google drive here: `https://drive.google.com/drive/folders/11xhfI2kJ5Qphz3HqGOjem08dS3uhJBqq?usp=sharing` ▸ Link

Parameter optimisation and simulation of these systems i

1. The base command to run the software from the command line[2] or Windows powershell calls python and the main python file:

   ```
   $ python main.py
   ```

2. Add the components you wish to simulate using the -c argument

   ```
   $ python main.py −c carbon_dioxide ethane
   ```

3. The base command to run the software from the command line[3] or Windows powershell calls python and the main python file:

   ```
   $ python main.py
   ```

4. Specify you wish to simulate using the -p argument, the default is x and y (VL(L)E)

   ```
   $ python main.py −c carbon_dioxide ethane −p x y
   ```

Parameter optimisation and simulation of these systems ii

5. Finally to do phase separation calculations where the software detects all phase separations and equilibrium points specify the pressure and temperature only using the arguments -P and -T

```
$ python main.py -c carbon_dioxide ethane -p x y \
    -P 101e3 -T 298.15
```

6. To do a single phase equilibrium computation at a specific feedpoint, add the using the -z argument

```
$ python main.py -c carbon_dioxide ethane -p x y \
   -P 101e3 -T 298.15 -z 0.5
```

Parameter optimisation and simulation of these systems iii

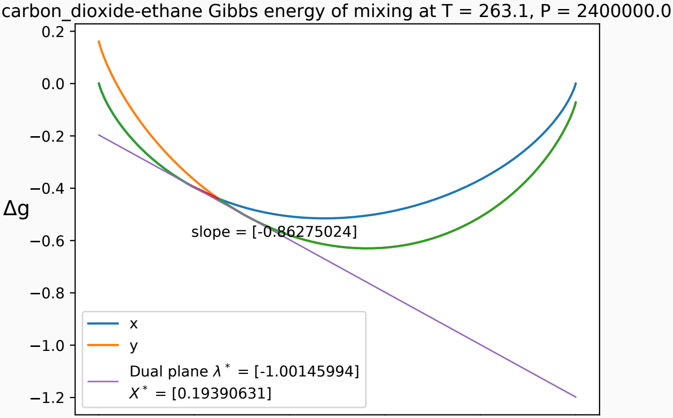


Figure 6: A phase equilibrium simulation of CO2 and ethane

Parameter optimisation and simulation of these systems iv

To optimise new parameters to a data set and save the results add the -optim and -save arguments:

```
$ python main.py -c acetone water -p x y -optim -save
```

The -pltiso argument will plot either isotherms or isobars with the current optimised parameters at the data P, T, $\bar{x}$ data points to compare the fit:

```
$ python main.py -c acetone water -p x y -pltiso
```

At a specific isotherm or isobar add for example a temperature argument:

```
$ python main.py -c acetone water -p x y -T 298.15 \
    -pltiso
```

[2] change directory to the installed directory first as discussed in the the installation and setup

[3] change directory to the installed directory first as discussed in the the installation and setup

Conclusions

Conclusions i

- The software is useful for rapid testing of thermodynamic models as well as for professional use in simulations requiring many thermodynamic computations
- However, the limitations in data handling and maintainability should be overcome before improvements to user facing end can begin.

Thank you for your time.

References

References i

Bollas, G., Barton, P., and Mitsos, A. (2009).
Bilevel optimization formulation for parameter estimation in vapor-liquid(-liquid) phase equilibrium problems.
Chemical Engineering Science, 64:1768–1783.

Mitsos, A. and Barton, P. I.
A dual extremum principle in thermodynamics.
AIChE Journal, 53(8):2131–2147.

Mitsos, A., Bollas, G. M., and Barton, P. I. (2009).
Bilevel optimization formulation for parameter estimation in liquid–liquid phase equilibrium problems.
Chemical Engineering Science, 64(3):548 – 559.

Questions?