The Journal of Open Source Software

# DESPASITO: A Python Package for SAFT EOS Parametrization and Thermodynamic Calculations

**Jennifer A. Clark** [1¶], **Nathan Duff**[1], **Andrew Abi-Mansour**[2], and **Erik E. Santiso** [1]

**1** Department of Chemical and Biomolecular Engineering, North Carolina State University, Raleigh, NC, 27695, USA **2** The Molecular Sciences Software Institute, Blacksburg, VA, 24060, USA ¶ Corresponding author

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Open Journals

**Reviewers:**

- @openjournals

**Submitted:** 01 January 1970
**Published:** unpublished

## Summary

DESAPSITO is an open-source Python package that provides access to complex thermodynamic calculations, with a special interest in the statistical associating fluid theory (SAFT) equation of state. The modular design of DESPASITO provides an extensible architecture that could be easily customized or extended by other researchers. The currently supported equations of state (SAFT-$\gamma$-Mie, SAFT-$\gamma$-SW, and Peng-Robinson) thoroughly represent the dynamic use of this platform and its intentional design for expansion. This growth mindset extends into the other modules, as additional thermodynamic calculations and parameter fitting routines may be included. Nonetheless, the current state of DESPASITO can be employed by the simulation community for coarse-grained forcefield development with SAFT-$\gamma$-Mie. The DESPASITO package is available from PyPi and its source code is hosted on GitHub, which provides a platform for community-driven contributions and feedback.

## 1. Statement of Need

To improve on the 20+ years needed between material development and market use,(Plunkett, 2017) theoretical studies have accelerated material design endeavors.(Fermeglia et al., 2020; J. M. Lee, 2019) Of the theoretical methods available, coarse-grained molecular dynamics (CG-MD) simulations have gained attention for their ability to access transport and mechanical properties, as well as phase separations of materials.(Joshi & Deshmukh, 2020; H. Lee & Larson, 2009; Miwatani et al., 2020; Nikkhah & Thompson, 2021) CG methods represent an atomistic ensemble as a single "bead" that must sufficiently capture the degrees of freedom of the underlying structure to produce realistic results.(Fermeglia et al., 2020) Although CG-MD yields computational efficiency in comparison to all-atom molecular dynamics (AA-MD), generating parameters that are representative of the system's physics is a continuing struggle. An appealing method of obtaining parameters entails a top-down approach to reproduce some macroscopic property in a simulation. Comparison to macroscopic properties entails an iterative procedure which is time consuming, as many lengthy simulations are needed.

The SAFT- -Mie CG methodology has emerged as an appealing CG formalism as the parameters are linked to the SAFT equation of state (EOS).(Müller & Jackson, 2014; Pervaje et al., 2019) The Statistical Associating Fluid Theory (SAFT) equation of state (EOS)(Walter G. Chapman et al., 1988, 1990; W. G. Chapman et al., 1989) inherently connects macroscopic properties with the nonbonded interactions of segments through the Helmholtz free energy. Details of theory, applications, and limitations of SAFT variants up to this point have been thoroughly described elsewhere.(Economou, 2002; Müller & Gubbins, 2001; Pervaje et al., 2019; Tan et al., 2008) SAFT has been historically applied to the complex fluids encountered by the oil and gas industry as an EOS, but recently it has received attention from computational scientists and engineers for molecular simulations. SAFT has many advantages due to its deep-rooted basis in statistical mechanics. SAFT-$\gamma$ (Avendaño et al., 2011; Ervik et al., 2016; Mejía, Herdes, et al., 2014; Müller & Jackson, 2014) offers transferable heteronuclear segments that have been developed into a CG formalism. (Dufal et al., 2014; Papaioannou et al., 2011, 2014) Other homonuclear versions of SAFT have also been applied to molecular simulations.(Cárdenas & Mejía, 2016; Galliero et al., 2007; José Matías Garrido et al., 2016; Sauer & Chapman, 2003) Use of SAFT improves CG parametrization, as the parameters fit to an EOS using experimental data can be applied in simulations, avoiding the much more computationally intensive alternative of iterative parametrization simulations with various system conditions. The result is a less computationally intensive methodology yielding chemical specificity. Both SAFT-$\gamma$-Mie (Avendaño et al., 2013; J. M. Garrido et al., 2016; Morgado et al., 2016; Pervaje et al., 2018, 2020; Walker et al., 2019, 2020) and SAFT-$\gamma$-square-well (SW) (Galindo et al., 1998; Gil-Villegas et al., 1997) have been successfully applied to molecular simulations.

Due, to its complexity, specialized software is needed to use SAFT either as an EOS or to predict simulation parameters. Bottled SAFT (Ervik et al., 2016) is a web application meant to remedy issues of parametrization by applying corresponding states for homonuclear molecules. This method relies on possessing the critical properties, acentric factor, and liquid density at 0.7 reduced temperature. Once bead parameters are acquired, whether from the literature or by other means, MD simulations can be carried out. Nevertheless, properties derived from simulations using SAFT are usually accompanied by the EOS prediction, reintroducing the need for specialized software. There are three fundamental requirements for a software that uses SAFT. The first is programming the EOS to generate the Helmholtz free energy. Second, thermodynamic functions must use the EOS to generate system properties. Third, a parametrization package should support both the EOS and thermodynamic functions to facilitate parameter fitting to macroscopic data. Writing this software is not a trivial task and serves as a barrier in development and application of related research. For a specific variant, PC-SAFT, the commercial package, Aspen, handles thermodynamic calculations and parameter estimation. Similarly, the commercial package gSAFT from Process Systems Enterprise, PSE, supports SAFT-$\gamma$-Mie. Another open-source thermodynamic package, CoolProp has been working toward implementing PC-SAFT.(Bell et al., 2014) Additionally, a Python package(Baird, 2021) that handles thermodynamic calculations with PC-SAFT was developed in the same time period as this work. This Python package is based on a set of open-source MATLAB scripts(Martín et al., 2011) that support the original SAFT implementation and PC-SAFT. However, these open-source options do not support parametrization. To produce an open-source software that fulfills the three requirements, this work describes the Python package: Determining Equilibrium State and Parametrization Application for SAFT, Intended for Thermodynamic Output (DESPASITO).(Clark et al., 2021)

Herein, we describe DESPASITO, a tool that (1) performs thermodynamic calculations of complex mixtures and (2) fits parameters to experimental data with the potential of accelerating and improving CG simulation research. DESPASITO is capable of reproducing studies published by groups with similar codes.(Avendaño et al., 2011, 2013; J. M. Garrido et al., 2016; Herdes et al., 2015, 2016; Lobanova et al., 2016; Mejía, Cartes, et al., 2014; Morgado et al., 2016; Müller & Jackson, 2014) Although DESPASITO focuses on the SAFT-$\gamma$ variant, it is equipped with general thermodynamic and parameterization modules that any supported EOS can use. In creating this platform to fit parameters for the SAFT EOS, we consequently equipped DESPASITO to handle other equations of state that cannot be solved explicitly. This platform is designed to allow easy and intuitive extension of each module and supported features. It is our purpose to build a base that will supply community accessibility to complex thermodynamic calculations.

# 2. Implementation and Architecture

## 2.1 Overview

The central purpose of DESPASITO is to calculate thermodynamic properties with equations of state (EOS). This package is implemented in the Python 3 programming language with four modules that use a combination of procedural and object-oriented approaches. DESPASITO incorporates the features supplied by the MolSSI Computational Molecular Science Python Cookiecutter (e.g., git, continuous integration, pytest, and readthedocs). The simplest use of DESPASITO involves passing a JSON input file to the command-line interface (CLI), which reads an input file and distributes relevant input data to appropriate modules. An input file may contain instructions for either a thermodynamic or parameter fitting calculation. Alternatively, DESPASITO can be used as a library via its application programming interface (API) for designing complex workflows. In the following sections we will first discuss the CLI, and in the subsequent sections discuss the API as it pertains to the core modules: *Equations of State*, *Thermodynamics*, and *Parameter Fitting*.

## 2.2 Command-Line Interface

DESPASITO provides a CLI that is useful when it is executed as a program. This simplifies the user experience as calculations are initiated in an input file. Here we describe some differences between the two interfaces. Once the input file is assembled, a calculation is submitted with: `python -m despasito -i input.json` The input file uses the JSON format to import the relevant information as a dictionary and then distributes relevant information to the other modules. Upon completing the provided instructions, the results are written to an output file. At a minimum, the input file contains a definition of the molecular components (i.e., bead definitions and quantities) and the bead self- and cross-interaction parameter files in the JSON format, as well the required instructions for either a thermodynamic or parameter fitting calculation. Whether the input file contains instructions for a thermodynamic or parameter fitting calculation is indicated by the presence of the keyword `optimization_parameters`. This section must then be accompanied by dictionaries containing experimental data and thermodynamic calculation requirements for a specific `data_class_type` defined within the dictionary structure. When defining experimental data, a user may list the relevant data in arrays within the input file or include the path to a comma separated file. Rather than further elaborating on the input keyword options or definitions here, the interested user will find the package documentation and tutorials useful. The ability to initiate either a thermodynamic or parameter fitting calculation from a single input file fosters a simple user experience for two complex processes. The logic followed by the CLI is summarized in Figure 1. This diagram shows two workflows that cover typical calculations. However, a user may choose to use DESPASITO as a library to construct a customized workflow or use the module components individually.

Flowchart of `despasito.main.run`, the function called by the command line interface. This process can distinguish between a thermodynamic and parameter fitting calculation. The colors orange, red, green, and purple, respectively represent the input_output, equations_of_state, thermodynamics, and parameter_fitting modules. Gray color indicates actions taken within `despasito.main.run`.

**Figure 1:** Flowchart of `despasito.main.run`, the function called by the command line interface. This process can distinguish between a thermodynamic and parameter fitting calculation. The colors orange, red, green, and purple, respectively represent the input_output, equations_of_state, thermodynamics, and parameter_fitting modules. Gray color indicates actions taken within `despasito.main.run`.

## 2.3 Computations Available in Current Version

The currently available capabilities in DESPASITO (v0.2.0) can be separated into three categories: *Equations of State*, *Thermodynamics*, and *Parameter Fitting*. Each of these categories is distinguished in this package as a separate module. Here we will summarize the options available to each category (i.e., module), supplemented with knowledge of program architecture where necessary.

Currently three EOSs are supported in DESPASITO: SAFT- $\gamma$-Mie (Avendaño et al., 2013; J. M. Garrido et al., 2016; Morgado et al., 2016; Pervaje et al., 2018, 2020; Walker et al., 2019, 2020), SAFT- $\gamma$-SW (Galindo et al., 1998; Gil-Villegas et al., 1997), and Peng-Robinson (Peng & Robinson, 1976). The latter was included as an example of an implementation of a non-SAFT EOS to demonstrate the extensible design in DESPASITO. An EOS object can encapsulate a variety of equations of state, and any compatible EOS object can interact with the *Thermodynamics* and *Parameter Fitting* modules. Details of EOS classes are elaborated on in Section 2.4. Even though DESPASITO is sufficient for CG-MD development with SAFT- -Mie or SAFT- -SW, the current EOS options are admittedly limited. Adding another EOS, or family is straightforward in DESPASITO. Further details can be found in the Reuse Policy section or in the documentation.

DESPASITO provides accessibility to many common thermodynamic calculation types, such as: multicomponent bubble point and dew point, binary flash, Hildebrand solubility, as well as vapor, liquid, and saturation properties. We also included a `verify_eos` function to carry out thermodynamic consistency tests for contributed EOSs. Our EOS classes are equipped to handle any number of components, however, some of the thermodynamic calculations are only robust for binary fluids (i.e., bubble point and dew point). Modeling binary systems with this set of thermodynamic calculations are sufficient for our initial purpose in fitting interaction parameters from experimental data. Second derivative properties are a priority in the next release.

To fit parameters to experimental data, many options are available to refine the process. A user can customize the parameter fitting routine by defining the objective functional form, choosing a global optimization method and its associated options, refining minimization method and its associated options, and defining nonlinear constraints. The default global optimization method is differential evolution, although many others are available as shown in Table 1. The custom method, `grid_minimization`, is based off the `scipy` optimization function, `brute`. Like `brute`, a grid of parameter values is created, but in this custom function the parameter sets are then changed with a minimization route to fit the experimental data. Some parameters can also be frozen. For example, a grid of two parameters may be defined, while those remaining are minimized. Global optimization of EOS parameters is agnostic to the EOS or thermodynamic data being used. Like the EOS is encapsulated in an object, the thermodynamic calculations are also abstracted from the parameter fitting process with Data classes. The Data class passes updates EOS parameters and returns the objective value used in the fitting procedure. The available data classes supported by DESPASITO, are found in Table 1. Further details on Data classes can be found in the Reuse Policy section or documentation.

## 2.4 Application Programming Interface

DESPASITO provides an application programming interface (API) where the capabilities of each of the core modules (i.e., *Equations of State*, *Thermodynamics*, and *Parameter Fitting*) may be accessed through their respective entry-points shown in Table 2. The *Equations of State* module is meant to supply an EOS object with a consistent set of methods exposed for the convenience of the *Thermodynamics* and *Parameter Fitting* modules. Other methods may be included for the internal function of the EOS object or independent use as an imported library. For the *Thermodynamics* module, any supported thermodynamic calculation can be computed through its entry point given the system information and an EOS object. The result of a thermodynamic calculation is a dictionary of both the system inputs and calculated outputs.

The entry-point for the *Parameter Fitting* module handles both the interaction between EOS objects and thermodynamic calculations and the interaction between thermodynamic calculations and an objective function used in the parameter fitting method. These interactions are handled with the initiation of a `Data` class for each dataset. However, because each dataset may represent a different molecular composition containing a common target bead, they must have an EOS object initiated beforehand. With these three core functions the major features of DESPASITO can be imported and used as a library.

Aside from the entry-points for the core modules previously described, the *Input / Output* module provides the functionality of importing and exporting the JSON files handled in by the CLI. Additionally, logging can be enabled to view intermediate values with the function, `despasito.initiate_logger`. Although all functions in DESPASITO are accessible to the user, we expect the functions described in this section to have streamlined the majority of use cases.

Although the outcome from using module entry-points for the *Thermodynamic* and *Parameter Fitting* modules are comparable to the CLI use of DESPASITO, exposing an EOS object and its available methods can have broader use cases. Figure 2 shows the inheritance structure of EOS classes in DESPASITO using an abstract template class to standardize EOS object methods and attributes. This standardization ensures compatibility with the *Thermodynamics* and *Parameter Fitting* modules. The SAFT EOS object uses another level of organization. As described earlier, SAFT relates nonbonded interactions to macroscopic properties through the Helmholtz free energy. Contributions to the Helmholtz free energy are segmented into terms, often these include the ideal, monomer, chain, and association site contributions. However, different SAFT variants may or may not include other terms (e.g. electrostatic and solvation terms (Shahriari & Dehghani, 2018) or polar terms (Dominik et al., 2005)). Often the ideal and association terms are consistent throughout the variants of SAFT and so are grouped in a common SAFT class, while the others are expressed in a `SaftType` class. An EOS object can be initiated, by using DESPASITO as a library, to allow each of these contributions to be handled individually. The *Parameter Fitting* module uses an abstract template class to standardize experimental `Data` classes. The role of Data objects is to handle EOS and thermodynamic calculations and output an objective value representing the ability of a given parameter set to reproduce experimental data.

*UML diagram of* Equation of State* classes used in DESPASITO. Required attributes and methods are shown in abstract class templates. Other methods and attributes are removed for clarity, except in the special case of SAFT. Here we illustrate how the common Helmholtz energy contributions are expressed in a shared class, while the specialized contributions of a SAFT variant are contained in an imported object.*

## 2.5 Compiled Modules

Operational speed can be increased with the optional use of compiled code within EOS classes. Apart from the association site contribution of the SAFT EOS, DESPASITO defaults to a pure Python implementation for the supported EOSs. It is heavily recommended that the association term calculation is then computed using `cython` or `numba`. The `method_stat` class is passed to an EOS class to indicate which method should be used. First, the keyword `numba` initiates just-in-time compilation for EOS functions with that supported option. Second, the `cython` flag initiates use of supported EOS functions that were optionally compiled upon installation. Third, the `python` flag signals that only Python should be used, *including* association site calculations, although its use is not recommended except for an exercise in patience. All three of these options have been implemented for the SAFT- -Mie EOS.

## 2.6 Parallelization

Thermodynamic calculations of the same type can be run in parallel. Python is equipped with a shared memory multiprocessing module that is handled in the utility directory, `despasito.utils`.

Here, the `MultiprocessingJob` class handles initiation of a processing pool, calculation dispatch, individual thread logging and consolidation into a job log. The addition of this shared memory utility allows a large set of calculations to run in parallel using high-performance computing (HPC) resources. Both the *Thermodynamics* and *Parameter Fitting* modules are equipped to use this class. If given numerous state conditions at which to perform a thermodynamic calculation (e.g., a list of liquid compositions to perform bubble point calculations), the calculations would be split and reassembled by the multiprocessing pool. A parameter fitting procedure may use the multiprocessing class in one of two ways. First, a single parameter set could be assessed at a time, where the thermodynamic calculations needed to compare to experimental data are split among the processing threads. Although, if a population of parameter sets are to be assessed for every iteration (as in differential evolution), the multiprocessing object may split the evaluation of the parameter sets among the processing threads. If the global algorithm supports workers, this is the default behavior.

## 3. Additional Features: Prediction of Cross-Interaction Parameters for SAFT-$\gamma$-Mie

DESPASITO is equipped to predict cross-interaction parameters (between segments of different types) using the open-source package, Multipole Approach of Predicting and Scaling Cross Interactions (MAPSCI) (*MAPSCI*, n.d.) as a plug-in. As the name implies, MAPSCI uses the multipole moments of molecular fragments in conjunction with their SAFT self-interaction parameters to estimate their cross-interactions. Currently, this package supports SAFT- -Mie. Use of this plugin reduces the degrees of freedom during parametrization, as SAFT self-interaction parameters are often fit to experimental data, and cross-interaction parameters usually require mixture data. Relying purely on experimental data for cross-interaction parameters is difficult due to the combinatorial explosion in the number of interaction parameters needed for an increasing number of components. MAPSCI is based on a derivation of the combining rules (in this case applied to SAFT- -Mie) that were extended to utilize multipole moments of molecular fragments.(Clark & Santiso, 2021) Although the multipole moments from small molecules can be found experimentally, the multipole moments of fragments can be derived from density functional theory (DFT).(Clark & Santiso, 2021) The difficulty in coarse-graining systems while retaining chemical specificity is a constant challenge that we seek to remedy.

## 4. Quality Control

Initial EOS development and module verification has been achieved through replicating the thermodynamic and parameter fitting results for existing publications. Unit testing is available through `pytest`, where continuous integration of MacOS, Windows, and Linux are handled by GitHub Actions. For each operating system, Python 3.7 and 3.8 are tested. These tests cover over fifty percent of the package according to `codecov`. Much of what is unaccounted for includes exception handling and redundant functions for alternative compilation through Numba and Cython.

## Acknowledgements

We thank Jacob Kausler for his aid in logging multiple processes.

## Funding statement

Jennifer Clark was supported by a fellowship from The Molecular Sciences Software Institute under NSF grant OAC-1547580.

# References

Avendaño, C., Lafitte, T., Adjiman, C. S., Galindo, A., Müller, E. A., & Jackson, G. (2013). SAFT-$\gamma$ force field for the simulation of molecular fluids: 2. Coarse-grained models of greenhouse gases, refrigerants, and long alkanes. *The Journal of Physical Chemistry B*, *117*(9), 2717. https://doi.org/10.1021/jp306442b

Avendaño, C., Lafitte, T., Galindo, A., Adjiman, C. S., Jackson, G., & Müller, E. A. (2011). SAFT-$\gamma$ force field for the simulation of molecular fluids. 1. A single-site coarse grained model of carbon dioxide. *The Journal of Physical Chemistry B*, *115*(38), 11154. https://doi.org/10.1021/jp204908d

Baird, Z. (2021). *Pcsaft*. https://pypi.org/project/pcsaft/1.2.0/

Bell, I. H., Wronski, J., Quoilin, S., & Lemort, V. (2014). Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. *Ind. Eng. Chem. Res.*, *53*(6), 2498. https://doi.org/10.1021/ie4033999

Cárdenas, H., & Mejía, A. (2016). Phase behaviour and interfacial properties of ternary system $CO_2$ + n-butane + n-decane: Coarse-grained theoretical modelling and molecular simulations. *Molecular Physics*, *114*(18), 2627. https://doi.org/10.1080/00268976.2016.1170221

Chapman, W. G., Gubbins, K. E., Jackson, G., & Radosz, M. (1989). SAFT: Equation-of-state solution model for associating fluids. *Fluid Phase Equilibria*, *52*, 31. https://doi.org/10.1016/0378-3812(89)80308-5

Chapman, Walter G., Gubbins, K. E., Jackson, G., & Radosz, M. (1990). New reference equation of state for associating liquids. *Industrial & Engineering Chemistry Research*, *29*(8), 1709. https://doi.org/10.1021/ie00104a021

Chapman, Walter G., Jackson, G., & Gubbins, K. E. (1988). Phase equilibria of associating fluids: Chain molecules with multiple bonding sites. *Molecular Physics*, *65*(5), 1057. https://doi.org/10.1080/00268978800101601

Clark, J. A., Duff, N., Abi-Mansour, A., & Santiso, E. E. (2021). *DESPASITO: Determining equilibrium state and parametrization application for SAFT, intended for thermodynamic output*. https://doi.org/10.5281/zenodo.4663126

Clark, J. A., & Santiso, E. E. (2021). SAFT-$\gamma$-mie cross-interaction parameters from density functional theory-predicted multipoles of molecular fragments for carbon dioxide, benzene, alkanes, and water. *Journal of Physical Chemistry B*, *125*(15), 3867. https://doi.org/10.1021/acs.jpcb.1c00851

Dominik, A., Chapman, W. G., Kleiner, M., & Sadowski, G. (2005). Modeling of polar systems with the perturbed-chain SAFT equation of state. Investigation of the performance of two polar terms. *Industrial & Engineering Chemistry Research*, *44*(17), 6928. https://doi.org/10.1021/ie050071c

Dufal, S., Papaioannou, V., Sadeqzadeh, M., Pogiatzis, T., Chremos, A., Adjiman, C. S., Jackson, G., & Galindo, A. (2014). Prediction of thermodynamic properties and phase behavior of fluids and mixtures with the SAFT-$\gamma$-mie group-contribution equation of state. *Journal of Chemical & Engineering Data*, *59*(10), 3272. https://doi.org/10.1021/je500248h

Economou, I. G. (2002). Statistical associating fluid theory: A successful model for the calculation of thermodynamic and phase equilibrium properties of complex fluid mixtures. *Industrial & Engineering Chemistry Research*, *41*(5), 953. https://doi.org/10.1021/ie0102201

Ervik, Å., Mejía, A., & Müller, E. A. (2016). Bottled SAFT: A web app providing SAFT-$\gamma$-mie force field parameters for thousands of molecular fluids. *Journal of Chemical Information and Modeling*, *56*(9), 1609. https://doi.org/10.1021/acs.jcim.6b00149

Fermeglia, M., Mio, A., Aulic, S., Marson, D., Laurini, E., & Pricl, S. (2020). Multiscale molecular modelling for the design of nanostructured polymer systems: Industrial applications. *Molecular Systems Design & Engineering*, *5*(9), 1447. https://doi.org/10.1039/d0me00109k

Galindo, A., Davies, L. A., Gil-Villegas, A., & Jackson, G. (1998). The thermodynamics of mixtures and the corresponding mixing rules in the SAFT-VR approach for potentials of variable range. *Molecular Physics*, *93*(2), 241. https://doi.org/10.1080/002689798169249

Galliero, G., Lafitte, T., Bessieres, D., & Boned, C. (2007). Thermodynamic properties of the mie n-6 fluid: A comparison between statistical associating fluid theory of variable range approach and molecular dynamics results. *The Journal of Chemical Physics*, *127*(18), 184506. https://doi.org/10.1063/1.2801997

Garrido, J. M., Algaba, J., Míguez, J. M., Mendiboure, B., Moreno-Ventas Bravo, A. I., Piñeiro, M. M., & Blas, F. J. (2016). On interfacial properties of tetrahydrofuran: Atomistic and coarse-grained models from molecular dynamics simulation. *The Journal of Chemical Physics*, *144*(14), 144702. https://doi.org/10.1063/1.4945385

Garrido, José Matías, Mejía, A., Piñeiro, M. M., Blas, F. J., & Müller, E. A. (2016). Interfacial tensions of industrial fluids from a molecular-based square gradient theory. *AIChE Journal*, *62*(5), 1781. https://doi.org/10.1002/aic.15190

Gil-Villegas, A., Galindo, A., Whitehead, P. J., Mills, S. J., Jackson, G., & Burgess, A. N. (1997). Statistical associating fluid theory for chain molecules with attractive potentials of variable range. *The Journal of Chemical Physics*, *106*, 4168. https://doi.org/10.1063/1.473101

Herdes, C., Forte, E., Jackson, G., & Müller, E. A. (2016). Predicting the adsorption of n-perfluorohexane in BAM-P109 standard activated carbon by molecular simulation using SAFT-$\gamma$ mie coarse-grained force fields. *Adsorption Science & Technology*, *34*, 64. https://doi.org/10.1177/0263617415619528

Herdes, C., Totton, T. S., & Müller, E. A. (2015). Coarse grained force field for the molecular simulation of natural gases and condensates. *Fluid Phase Equilibria*, *406*, 91. https://doi.org/10.1016/j.fluid.2015.07.014

Joshi, S. Y., & Deshmukh, S. A. (2020). A review of advancements in coarse-grained molecular dynamics simulations. *Molecular Simulation*, *47*(10-11), 786. https://doi.org/10.1080/08927022.2020.1828583

Lee, H., & Larson, R. (2009). Multiscale modeling of dendrimers and their interactions with bilayers and polyelectrolytes. *Molecules*, *14*, 423. https://doi.org/10.3390/molecules14010423

Lee, J. M. (2019). *A review on recent progress in the aspect of stability of oxygen reduction electrocatalysts for proton-exchange membrane fuel cell: Quantum mechanics and experimental approaches*. *7*(9), 22. https://doi.org/10.1002/ente.201900312

Lobanova, O., Mejía, A., Jackson, G., & Müller, E. A. (2016). SAFT-$\gamma$ force field for the simulation of molecular fluids 6: Binary and ternary mixtures comprising water, carbon dioxide, and n-alkanes. *The Journal of Chemical Thermodynamics*, *93*, 320. https://doi.org/10.1016/j.jct.2015.10.011

*MAPSCI: Multipole approach of predicting and scaling cross interactions*. (n.d.).

Martín, Á., Bermejo, M. D., Mato, F. A., & Cocero, M. J. (2011). Teaching advanced equations of state in applied thermodynamics courses using open source programs. *Education for Chemical Engineers*, *6*(4), e114. https://doi.org/10.1016/j.ece.2011.08.003

Mejía, A., Cartes, M., Segura, H., & Müller, E. A. (2014). Use of equations of state and coarse grained simulations to complement experiments: Describing the interfacial properties of carbon dioxide + decane and carbon dioxide + eicosane mixtures. *Journal of Chemical & Engineering Data*, *59*(10), 2928. https://doi.org/10.1021/je5000764

Mejía, A., Herdes, C., & Müller, E. A. (2014). Force fields for coarse-grained molecular simulations from a corresponding states correlation. *Industrial & Engineering Chemistry Research*, *53*(10), 4131. https://doi.org/10.1021/ie404247e

Miwatani, R., Takahashi, K. Z., & Arai, N. (2020). Performance of coarse graining in estimating polymer properties: Comparison with the atomistic model. *Polymers*, *12*(2), 382. https://doi.org/10.3390/polym12020382

Morgado, P., Lobanova, O., Müller, E. A., Jackson, G., Almeida, M., & Filipe, E. J. M. (2016). SAFT-$\gamma$ force field for the simulation of molecular fluids: 8. Hetero-segmented coarse-grained models of perfluoroalkylalkanes assessed with new vapour-liquid interfacial tension data. *Molecular Physics*, *114*(18), 2597. https://doi.org/10.1080/00268976.2016.1218077

Müller, E. A., & Gubbins, K. E. (2001). Molecular-based equations of state for associating fluids: A review of SAFT and related approaches. *Industrial & Engineering Chemistry Research*, *40*(10), 2193. https://doi.org/10.1021/ie000773w

Müller, E. A., & Jackson, G. (2014). Force-field parameters from the SAFT-$\gamma$ equation of state for use in coarse-grained molecular simulations. *Annual Review of Chemical and Biomolecular Engineering*, *5*, 405. https://doi.org/10.1146/annurev-chembioeng-061312-103314

Nikkhah, S. J., & Thompson, D. (2021). Molecular modelling guided modulation of molecular shape and charge for design of smart self-assembled polymeric drug transporters. *Pharmaceutics*, *13*(2), 141. https://doi.org/10.3390/pharmaceutics13020141

Papaioannou, V., Lafitte, T., Adjiman, C. S., Galindo, A., & Jackson, G. (2011). Simultaneous prediction of phase behaviour and second derivative properties with a group contribution approach (SAFT-$\gamma$-mie). In E. N. Pistikopoulos, M. C. Georgiadis, & A. C. Kokossis (Eds.), *21st european symposium on computer aided process engineering* (Vol. 29, p. 1593). Elsevier. https://doi.org/10.1016/B978-0-444-54298-4.50097-0

Papaioannou, V., Lafitte, T., Avendaño, C., Adjiman, C. S., Jackson, G., Müller, E. A., & Galindo, A. (2014). Group contribution methodology based on the statistical associating fluid theory for heteronuclear molecules formed from mie segments. *The Journal of Chemical Physics*, *140*(5), 054107. https://doi.org/10.1063/1.4851455

Peng, D.-Y., & Robinson, D. B. (1976). A new two-constant equation of state. *Industrial & Engineering Chemistry Fundamentals*, *15*, 59. https://doi.org/10.1021/i160057a011

Pervaje, A. K., Tilly, J. C., Detwiler, A. T., Spontak, R. J., Khan, S. A., & Santiso, E. E. (2020). Molecular simulations of thermoset polymers implementing theoretical kinetics with top-down coarse-grained models. *Macromolecules*, *53*(7), 2310. https://doi.org/10.1021/acs.macromol.9b02255

Pervaje, A. K., Tilly, J. C., Inglefield, D. L., Spontak, R. J., Khan, S. A., & Santiso, E. E. (2018). Modeling polymer glass transition properties from empirical monomer data with the SAFT-$\gamma$ mie force field. *Macromolecules*, *51*(23), 9526. https://doi.org/10.1021/acs.macromol.8b01734

Pervaje, A. K., Walker, C. C., & Santiso, E. E. (2019). Molecular simulation of polymers with a SAFT-$\gamma$-mie approach. *Molecular Simulation*, *45*(14-15), 1223. https://doi.org/10.1080/08927022.2019.1645331

Plunkett, J. W. (2017). *Plunkett's nanotechnology & MEMS industry almanac 2017*. PLUNKETT RESEARCH. ISBN: 978-1-62831-439-7

Sauer, S. G., & Chapman, W. G. (2003). A parametric study of dipolar chain theory with applications to ketone mixtures. *Industrial & Engineering Chemistry Research*, *42*(22), 5687. https://doi.org/10.1021/ie034035u

Shahriari, R., & Dehghani, M. R. (2018). New electrolyte SAFT-VR morse EOS for prediction of solid-liquid equilibrium in aqueous electrolyte solutions. *Fluid Phase Equilibria*, *463*, 128.

https://doi.org/10.1016/j.fluid.2018.02.006

Tan, S. P., Adidharma, H., & Radosz, M. (2008). Recent advances and applications of statistical associating fluid theory. *Industrial & Engineering Chemistry Research*, *47*(21), 8063. https://doi.org/10.1021/ie8008764

Walker, C. C., Genzer, J., & Santiso, E. E. (2019). Development of a fused-sphere SAFT-$\gamma$ mie force field for poly(vinyl alcohol) and poly(ethylene). *The Journal of Chemical Physics*, *150*(3), 034901. https://doi.org/10.1063/1.5078742

Walker, C. C., Genzer, J., & Santiso, E. E. (2020). Extending the fused-sphere SAFT-$\gamma$ mie force field parameterization approach to poly(vinyl butyral) copolymers. *The Journal of Chemical Physics*, *152*(4), 044903. https://doi.org/10.1063/1.5126213