Introduction to Use Association NRTL Python Code

This document explains how to use the attached Association NRTL activity coefficient model python code. An example file is also included for illustration. The reference for the Association NRTL model is “*Hao, Y., & Chen, C. C. (2020). Nonrandom two‐liquid activity coefficient model with association theory. AIChE Journal, 67(1), e17061.”.*

# Setup external python libraries

Several open-source third-party python libraries, e.g. numpy, scipy, matplotlib, have been used for numerical computation, optimization, and plotting. A “requirements.txt” file is attached for installing all dependent libraries by using the command “pip install -r requirements.txt”. Python virtual environment is recommended.

# Example Association NRTL model application

Activity coefficient calculation for a binary system, chloroform-water, is illustrated here to show how to use the Association NRTL model python code. The example code is in the attached “example\_AssociationNRTL.py” file.

## Create “AssociationNRTL” object from model input parameters

# Use chloroform-water binary case as an example

# 1-chloroform 2-water

ri **=** np**.**array**([**2.87**,** 0.76**])** # r\_I parameters as in Eqn 3, 10, 11

tauij **=** np**.**zeros**((**2**,** 2**))** # tau\_IJ as in Eqn 4, 5

tauij**[**0**][**1**]** **=** **-**1.21

tauij**[**1**][**0**]** **=** 2.85

alphaij **=** np**.**zeros**((**2**,** 2**))** # alpha\_IJ as in Eqn 5, 6

alphaij**[**0**][**1**]** **=** alphaij**[**1**][**0**]** **=** 0.2

nu **=** **[[**1**],** **[**2**,** **-**2**]]** # nu\_A,I as in Eqn 7, 10, 11; positive for HB donor sites and negative for acceptor sites

delta **=** **[[**0.145**],** **[**1.0**,** 1.0**]]** # delta^A or delta^D as in Eqn 12

# Association NRTL class is constructed by model parameters

ChloroformWaterBinary **=** AssociationNRTL**(**r\_i**=**ri**,**

tau\_ij**=**tauij**,**

alpha\_ij**=**alphaij**,**

nu\_i**=**nu**,**

delta\_i**=**delta**)**

r\_i: Normalized volume parameters for each component. They are passed as a numpy 1-dimentional array with size equal to the number of components (NOC) in the system.

tau\_ij and alpha\_ij: NRTL binary interaction parameter and nonrandomness factor, specified as numpy 2-dimensional arrays with sizes equal to NOC.

nu\_i: number of association sites on each component molecule. It is specified as a list of lists. The size of the outer layer list is NOC. The inner list holds integers representing number of association sites on the molecule, with positive numbers representing hydrogen bond (HB) donor sites and negative numbers representing HB acceptor sites. For example, water has 2 HB donor sites and 2 HB acceptor sites on each molecule and is specified as [2, -2]; chloroform, [1], has 1 HB donor site; n-hexane, [], has no HB donor or acceptor site; ethylene glycol butyl ether (EGBE), [1, -2, -2], has 1 HB donor site and 4 HB acceptor sites, albeit with different association strengths (see section 3.4 of the reference).

delta\_i: molecule-specific association strength parameters, specified as list of lists with the same structure as nu\_i representing the association strength of each corresponding site.

AssociationNRTL: the Association NRTL model is designed as an object, which is created by all the input parameters as described above.

## Calculate activity coefficients from “AssociationNRTL” object

# activity coefficient is calculated by compute function by pass in temperature and mole fraction

x **=** np**.**array**([**0.2**,** 0.8**])**

T **=** 300.0 # K

lnGamma **=** ChloroformWaterBinary**.**compute**(**x**,** T**)**

**print(**f'Calculated ln(gamma):\n{lnGamma}'**)**

After the “AssociationNRTL” object is created from the input parameters, activity coefficients can be calculated repeatedly at any temperature and composition using the “compute” function.

x: mole fraction of each component, which is specified as numpy 1-dimensional array of size NOC.

T: temperature in Kelvin.

Output: the “compute” function returns as numpy 1-dimentional array with the size of NOC in the system.

# Additional calculated details (optional)

# additional information of calculated values can be retrieved by passing in an empty dictionary

info **=** **{}**

lnGamma **=** ChloroformWaterBinary**.**compute**(**x**,** T**,** info**)**

**print(**f'Additional calculated values:\n{info}'**)**

If an empty dictionary (info) is passed to the “compute” function (optional), some calculated details will be stored in that dictionary object. An example “info” object holding information is shown below:

**{**'xA'**:** **[**0.13804127110109904**,** 0.6125265667831137**,** 0.18647545025320697**],**

'lnGammaC'**:** **[-**0.25208731207917334**,** **-**0.028968923750010422**],**

'lnGammaR'**:** **[**0.4838221528364055**,** 0.0808723270678825**],**

'lnGammaA'**:** **[**2.8584956519828877**,** 0.29794976343586255**]}**

xA: unbonded site fraction in the mixture in the order of all HB acceptor sites, and all HB donor sites.

lnGammaC, lnGammaR, lnGammaA: combinatorial, residual, and association contributions to the activity coefficients.